Artificial Intelligence Chemistry 3 (2025) 100087

Contents lists available at ScienceDirect

Artificial Intelligence Chemistry

journal homepage: www.journals.elsevier.com/artificial-intelligence-chemistry

Enhanced prediction of ionic liquid toxicity using a meta-ensemble learning framework with data augmentation

Safa Sadaghiyanfam [a], Hiqmet Kamberaj [b,c], Yalcin Isler [d,*]

[a] Izmir Katip Celebi University, Department of Biomedical Technologies, Balatcik Campus, Cigli, 35620 Izmir, Turkey
[b] International Balkan University, Department of Computer Engineering, Skopje, North Macedonia
[c] National Institute of Physics, Fan Noli Square, Academy of Sciences of Albania, Tirana, Albania
[d] Alanya Alaaddin Keykubat University, Department of Electrical and Electronics Engineering, Kestel Campus, Alanya, 07425 Antalya, Turkey

ARTICLE INFO

Keywords:
Ionic liquids (ILs)
Machine learning
Meta-ensemble learning
Data augmentation
Recursive Feature Elimination (RFE)
Quantitative Structure-Activity/Property Relationship (QSAR/QSPR)
Chemception
Feature reduction
GridSearchCV

ABSTRACT

Ionic liquids are unique in their properties and potential to be green solvents. Still, the toxicity concern remains, compelling the need for excellent predictive models for safe design and application. This work reports the introduction of a general, robust meta-ensemble learning framework for predicting the toxicity of ionic liquids using molecular descriptors and fingerprints. The proposed model incorporates the Random Forest, Support Vector Regression, Categorical Boosting, Chemical Convolutional Neural Network as a base classifier and an Extreme Gradient Boosting meta-classifier. The framework uses Recursive Feature Elimination for feature selection and GridSearchCV for tuning the best hyperparameters. Without augmentation of the data, the RMSE equals 0.38, MAE equals 0.29, coefficient of determination (R^2) equals 0.87, and Pearson correlation equals 0.94. Data augmentation further improved model performance: RMSE = 0.06, MAE = 0.024, R^2 = 0.99, and a Pearson correlation of 0.99. In addition, this indicates that the data-augmented model outperforms all existing models with prominence in its strength and prediction capacity. Thus, the present framework provides a superior tool for computer-aided molecular design of safer and more effective ionic liquids.

1. Introduction

Ionic liquids (ILs) are typically described as salts in a liquid state, consisting of ions, with melting points below 100 degrees Celsius [1,2]. In recent decades, the use and study of ILs have grown substantially, drawing considerable interest. Their versatility in structure and properties makes ILs highly suitable for a wide range of extraction [3], adsorption [4], electrochemistry [5], biocatalysis [6] applications. ILs demonstrate negligible vapor pressure at room temperature [7], are non-flammable, and maintain strong physical and chemical stability. As a result, they are seen as alternatives to traditional solvents, though this perception can be misleading, as they are often incorrectly assumed to be non-toxic and environmentally friendly solvents [8]. Despite being termed green solvents, the widespread use of ILs requires careful consideration of their toxicity when applying them on an industrial scale. According to the principles of green chemistry [9], chemical products should be designed and developed in a way that reduces or eliminates the production and use of harmful substances to both the environment and human health. Thus, green chemistry aims to use non-toxic substances and prioritizes compounds derived from renewable resources [9,10]. The toxicity of ILs has become an unavoidable subject of discussion. Research has shown that ILs exhibit varying levels of toxic effects on fish [11], plants [12], cells [13], and microorganisms [14]. However, the vast number of ILs created by combining different anions and cations makes it impractical to experimentally test the toxicity of each one [15]. For comprehensive testing and active design of a wide range of IL properties and structures, computational models are preferred over experimental methods. These models are faster, safer, and more cost-effective [16–18]. Research in computational chemistry and modeling has immense potential for the future development of new and environmentally friendly ILs.

1.1. Critical descriptors

The descriptors, Electrostatic Potential Surface Area (S_{EP}) and Screening Charge Density Distribution Area (Sigma-Profile, S_σ), played a critical role in predicting the toxicity of ionic liquids (ILs) [19]. The S_{EP} descriptor, which quantifies the surface area of molecules within specific electrostatic potential ranges, provided insights into the electron-level interactions crucial for understanding the impact of cations and anions on acetylcholinesterase enzyme activity.

* Corresponding author.
E-mail addresses: safa.sd1993@gmail.com (S. Sadaghiyanfam), h.kamberaj@gmail.com (H. Kamberaj), islerya@yahoo.com (Y. Isler).

https://doi.org/10.1016/j.aichem.2025.100087
Received 16 October 2024; Received in revised form 21 January 2025; Accepted 24 February 2025
Available online 5 March 2025

Meanwhile, the Sigma-Profile (S_σ) descriptor, derived from the COSMO-RS methodology, characterized the distribution of screening charge density across molecular surfaces, highlighting regions such as hydrogen-bond donors, hydrogen-bond acceptors, and non-polar zones. These descriptors collectively revealed that cations exhibited a more significant influence on toxicity than anions, underscoring their critical role in molecular interactions and toxicity mechanisms.

1.2. State of the art

The study of ionic liquids (ILs) has progressed significantly over the past few decades, with considerable advancements in their applications, particularly due to their unique properties. This section provides a comprehensive review of recent research, focusing on the computational modeling approaches used to predict the toxicity and physical properties of ILs. Cao et al. [20] employed Multiple Linear Regression (MLR), Support Vector Machine (SVM), and Extreme Learning Machine (ELM) as modeling approaches. ELM showed the best performance, with an R^2 of 0.974 for the training set and 0.937 for the test set. The study found that cations had a greater impact on IL toxicity than anions, and the toxicity of ILs increases with the elongation of the alkyl side chain length. Longer alkyl chains can integrate into the polar headgroups of phospholipid bilayers, disrupting cell membrane structure and increasing permeability. ILs with longer side chains exhibit surfactant-like behavior, interacting with membrane proteins and damaging cell membranes. This increase in lipophilicity enhances ILs' ability to integrate into cell membranes, thereby contributing to their toxic effects. According to [19], MLR and ELM were employed to develop QSAR models, with ELM significantly outperforming MLR. ELM, a three-layer artificial neural network, leverages a streamlined training process by fixing randomly initialized weights and biases in the hidden layer and optimizing the output weights analytically. This efficient approach enabled ELM to effectively handle complex nonlinear relationships between molecular descriptors such as S_{EP} and S_σ. The toxicity outcomes achieved an R^2 of 0.969 for the training set and 0.950 for the test set. These results highlight ELM's computational efficiency, high accuracy, and robust generalization performance. Yuan et al. [21] utilized the Tox21 dataset, which includes 12 properties divided into Nuclear Receptor and Stress Response panels, with over 12,000 molecules to apply a Convolutional Neural Network (CNN) with four hidden layers for toxicity prediction. The CNN model, leveraging a binary cross-entropy loss function and techniques like SMOTE for data augmentation, achieved higher Area Under the Curve (AUC) values across most of these properties compared to traditional machine learning methods. The multi-channel grid-based CNN method outperformed other deep learning approaches, such as Chemception, Autoencoder, and DNN. Wang, Song, and Zhou [22] employed a Feedforward Neural Network (FNN) and Support Vector Machine (SVM) for modeling. The SVM model slightly outperformed the FNN model, achieving an R^2 of 0.9202 for the test set. Both models demonstrated good predictive performance, with the SVM model showing better accuracy. Daili and Francesco [23] used a dataset of 127 ILs, with σ-profile descriptors calculated using the GC-COSMO method [24]. The models included Multiple Linear Regression (MLR) and Multilayer Perceptron (MLP). The MLP-2 model, which was noted as the best-performing QSAR model in the study, demonstrated the highest predictive accuracy, achieving an R^2 of 0.938 for the test set. The study highlighted the superior performance of MLP-2 in predicting IL toxicity toward IPC-81 leukemia rat cell lines. A study by Baskin, Epshtein, and Ein-Eli [25] benchmarked machine learning methods for modeling the physical properties of ionic liquids. This study benchmarked various machine learning methods, including Partial Least Squares Regression (PLS), Random Forest Regression (RFR), Extreme Gradient Boosting (XGBoost), Associative Neural Network (ASNN), Deep Neural Network (DNN), and others, on datasets containing 407 to 1204 ILs. Nonlinear ML methods significantly outperformed linear ones, with TransCNN and TransCNF showing superior performance due to their advanced ability to analyze chemical structures encoded in SMILES strings. Fan et al. [26] applied Random Forest (RF) and XGBoost models. The XGBoost model, optimized via Bayesian methods, outperformed RF with an R^2 of 0.957 for the test set. SHAP analysis revealed that cationic side chain length and specific anions significantly influenced IL toxicity. Tabaaza et al. [27] applied various machine learning models, including Decision Tree, Random Forest, Extra Trees Regression, Gradient Boosting Regression, and XGBoost. The XGBoost model performed best, achieving an R^2 of 0.79 for the test set. Feature importance analysis indicated that cationic hydrophilicity and side chain length significantly impact toxicity. Smith et al. [28] develop MLR-Elastic Net (MLR-EN) and Artificial Neural Network (ANN) models. The MLR-EN model consistently achieved higher R^2 values on unseen datasets, indicating more reliable and robust performance. A study by Mousavi et al. [29] compared white-box machine learning, deep learning, and ensemble learning approaches for modeling H2S solubility in ionic liquids. This study compared various models, including GMDH, GP, DBN, and XGBoost. The XGBoost model achieved the best performance with an AAPRE of 1.14%, demonstrating superior accuracy in predicting H2S solubility. Fan et al. [30] applied a Deep Convolutional Neural Network (DCNN) model, achieving an R^2 of 0.965 for the test set. The 10-fold cross-validation confirmed consistent performance, with the DCNN model outperforming traditional QSAR/QSPR models. Semenyuta et al. [31] developed QSTR models using Associative Neural Network (ASNN), Transformer Convolutional Neural Network (Trans-CNN), and Random Forest (RF) on datasets of 75 compounds for Daphnia magna toxicity and 99 compounds for Danio rerio toxicity. The models achieved high q^2 values (a statistical parameter used to measure the predictive ability of models, specifically in the context of cross-validation), successfully predicting IL toxicity with high accuracy. Abdullah et al. [32] applied various models, including Ridge Regression, LASSO, Decision Tree, Random Forest, Extra Trees, Gradient Boost, and Support Vector Regression. Random Forest showed the best performance, with MSDC identified as the most significant descriptor, contributing 67% to the prediction. Building on these findings, our study presents an improved model for predicting IL toxicity through a meta-ensemble learning framework combined with data augmentation. This method overcomes the limitations of prior models by combining predictions from various machine learning algorithms, thereby increasing both accuracy and robustness. The meta-ensemble framework not only boosts predictive performance but also enhances interpretability by utilizing multiple molecular descriptors, making it a valuable tool for future assessments of IL toxicity.

1.3. Contributions

The main contributions of this research study are summarized as follows:

- To develop a robust **meta-ensemble learning** framework for predicting the toxicity of ionic liquids (ILs) using molecular descriptors and fingerprints.
- To compute and integrate molecular fingerprints (using Morgan algorithm) and **RDKit descriptors from SMILES strings**, forming a comprehensive feature matrix.
- To apply **Recursive Feature Elimination (RFE)** with a RandomForestRegressor for effective dimensionality reduction, selecting the most informative features for toxicity prediction.
- To explore and optimize the performance of various base models, including **Random Forest, Support Vector Regression (SVR), XGBoost, and CatBoost, through GridSearchCV and RandomizedSearchCV**.
- To construct and train **a neural network model** using TensorFlow, incorporating Conv1D, MaxPooling1D, and Dense layers for capturing intricate patterns within the data.

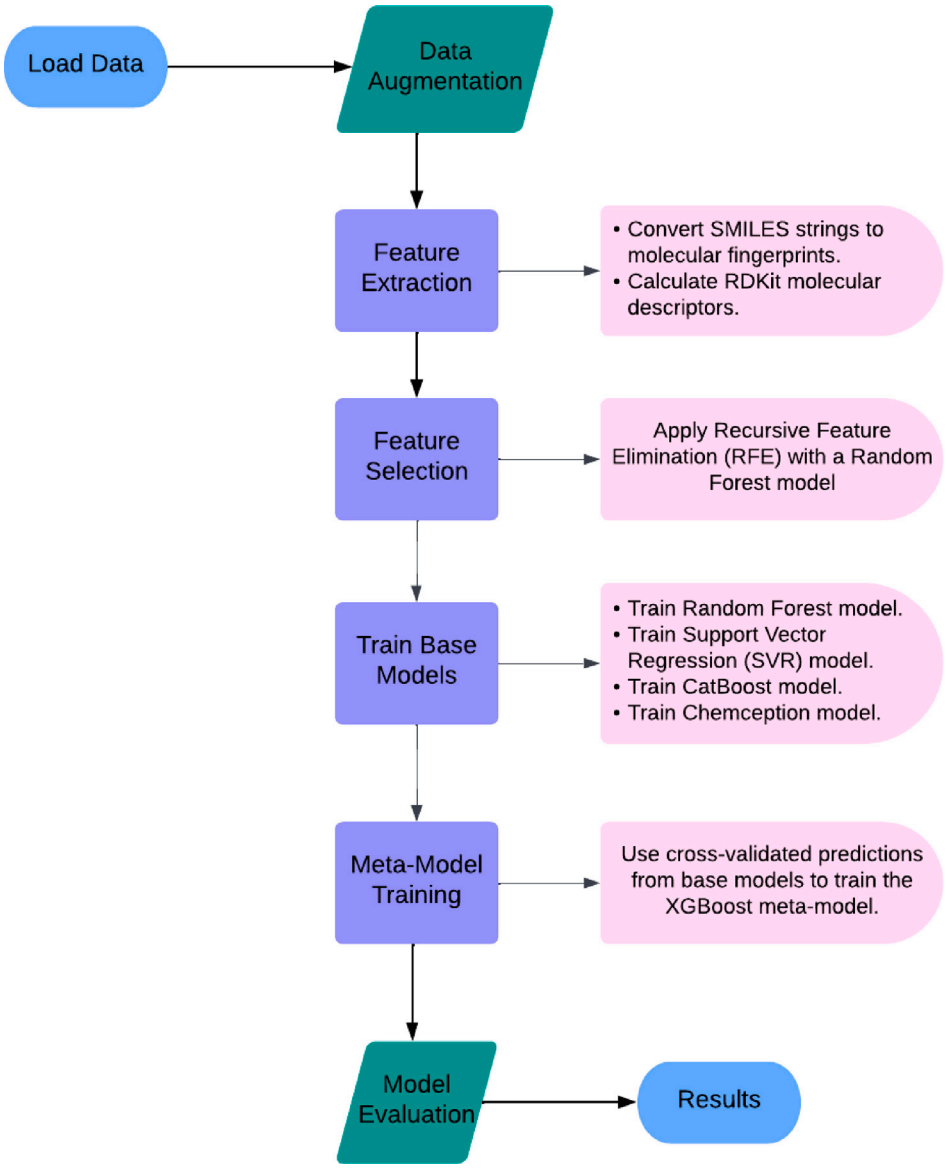

Fig. 1. Proposed meta-ensemble framework for predicting ionic liquid toxicity.

- To integrate predictions from **multiple base models into a meta-learner**, enhancing overall prediction accuracy and robustness for IL toxicity.
- To present a **comprehensive analysis of the meta-ensemble model's effectiveness**, highlighting its potential for aiding in the design of safer and more effective ionic liquids.

In this study, we propose an innovative meta-ensemble framework for predicting the toxicity of ionic liquids (ILs) by integrating molecular descriptors [33] and fingerprints [1]. You can see the proposed approach in Fig. 1. Our approach is designed to handle both extensive and limited training samples effectively. We address the challenge of predicting IL toxicity by combining multiple machine-learning models to enhance prediction accuracy and robustness, a significant advancement over traditional single-model approaches commonly found in the literature. Theoretical foundations for ensemble learning suggest that by minimizing generalization error through error mitigation and leveraging the diversity of different models, our ensemble approach capitalizes on the strengths of each component model. This leads to more accurate and stable predictions, as each model can capture unique aspects of the data. Moreover, ensemble methods enhance robustness against overfitting by averaging predictions from multiple models, which helps reduce noise and prevents fitting to specific data patterns. Supported by theoretical frameworks like the Condorcet Jury Theorem, ensemble learning shows that a group of models can achieve higher accuracy collectively than any single model [34].

We utilized several performance metrics to assess our model, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), and Pearson correlation. In addition, we tracked the loss function during training to ensure convergence. The loss function offers a numerical evaluation of how closely the model's predictions match the actual data. By monitoring the reduction in loss across iterations, we confirm that the model converges toward an optimal solution, thereby enhancing its predictive accuracy and stability.

Our findings demonstrate the superiority of the meta-ensemble model in capturing complex relationships within the data, offering a promising tool for the design of safer and more effective ionic liquids.

1.4. Organization

The rest of the paper is organized as follows: Section 2 describes the materials and methods used in this study, as well as the assessment criteria adopted for the experiments. Section 3 discusses the observations

Table 1
Data for ionic liquids and their experimental $logEC_{50}$.

IL No.	SMILES	Experimental $logEC_{50}$
1	[N+](C)(C)(CC)COCC.[Cl-]	3.59
2	O1c4c(O[B-]12Oc3c(O2)cccc3)cccc4.CC[N+](CC)(CC)CC	1.17
3	[N+](C)(C)(Cc1ccccc1)CCCCCCCCCC.[Cl-]	0.64

and presents the results of the proposed approaches. Finally, Section 4 provides a summary and conclusion.

2. Materials and methods

This study is comprehensively structured into five steps: Data Acquisition, Feature Extraction, Feature Reduction, Model Development, and Model Application. Each step and the operations performed within them are explained in detail under their respective sub-headings.

2.1. Dataset description

In this study, the data for the toxicity prediction of ionic liquids was obtained from the article by [35]. Toxicity is expressed as $logEC_{50}$, the base-10 logarithm of the half-maximal effective concentration (EC_{50}) measured in micromolar (μM). Higher values of $logEC_{50}$ indicate lower toxicity. The dataset includes information on the toxicity of various ionic liquids (ILs) measured by their $logEC_{50}$ values. The dataset comprises 355 entries of different ionic liquids, each characterized by SMILES (Simplified Molecular Input Entry System) and corresponding Experimental $logEC_{50}$ which experimentally represent the toxicity of the Ionic Liquid. Below is a sample of the data showing a few rows to illustrate the structure and type of information included: (see Table 1).

2.1.1. Distribution of cation families and anions in the dataset

That dataset includes a diverse range of cations and anions. Below, we provide a detailed breakdown of the dataset:

Cation families.

- **1-Butyl-3-methylimidazolium:** Present in 9 instances
- **1-Ethyl-3-methylimidazolium:** Present in 5 instances
- **3-Methyl-1-octylimidazolium:** Present in 4 instances
- **3-Methyl-1-nonylimidazolium:** Present in 3 instances

Other cations, including pyridinium, ammonium, and morpholinium-based structures, appear less frequently, contributing to the dataset's diversity.

Anions families. The most frequent anions in the dataset are:

- **Amide:** 52 instances
- **Chloride:** 43 instances (combining capitalization variations)
- **Tetrafluoroborate:** 38 instances (including alternative spellings)
- **Bromide:** 14 instances (combining capitalization variations)

Less common anions include sulfate, acetate, and phosphate, among others. These anions further expand the chemical diversity of the dataset.

We conducted several analyses to ensure the dataset's quality and reliability. The Tanimoto similarity index was used to evaluate the structural diversity of the molecules, confirming that the dataset includes a broad range of compounds and is not biased toward particular chemical structures.

Additionally, a violin plot was utilized to examine the distribution of toxicity values, helping to identify any skewness or anomalies. These quality assessments verify the dataset's suitability for developing robust predictive models. Detailed explanations of the Tanimoto similarity and Violin plot analyses are provided in the subsequent sections.

The data used in this study can be found in the Supplementary Information.

2.1.2. Tanimoto similarity

The Tanimoto Similarity, often referred to as the Jaccard index, is a measure used to assess the similarity between two sets. In chemistry and molecular informatics, this metric is commonly applied to compare chemical structures. It plays a crucial role in activities like virtual screening, molecular fingerprinting, and compound clustering. This definition is detailed in the article by Willett et al. [36]. The overall similarity of the molecules was calculated and expressed as a percentage, resulting in approximately 19.21%. This indicates that, on average, the molecules in the dataset share about 19.21% similarity with each other based on the Tanimoto Similarity index. A similarity percentage of 19.21% (or 0.1921) falls within the "Low Similarity" range. This suggests that, on average, the molecules in the dataset do not share many common features and are relatively diverse.

This study aims to explore a diverse set of molecules, where a low Tanimoto similarity percentage is desirable, as it reflects a wide variety of chemical structures (Fig. 2). The observed overall similarity of 19.21% indicates significant structural diversity, which is advantageous for investigating a broad spectrum of chemical properties. The detailed Tanimoto similarity matrix is provided in the Supporting Information as an Excel file.

2.1.3. Violin plot

A violin plot combines the features of a box plot – summarizing statistics such as the median, interquartile range, and outliers – with a kernel density plot, which represents the data's probability density. To estimate this density, a kernel smoothing function, typically Gaussian, is applied, allowing for a more detailed and continuous visualization of the data distribution. The violin plot is an effective tool for visualizing the distribution [37] of toxicity values in the dataset, offering both summary statistics and insights into the data's density and variability For a comprehensive guide, refer to Atlassian [38]. In this study, toxicity is quantified as $logEC_{50}$, indicating the concentration at which an ionic liquid exhibits a 50% toxic effect. These values are expected to be continuously distributed, reflecting a spectrum of toxic effects across varying concentrations. The violin plot integrates the features of a box plot with a kernel density plot, enabling us to observe the median, interquartile range, potential outliers, and the overall shape of the data's distribution. This visualization is particularly useful for detecting patterns such as skewness, bimodal distributions, or clusters that might not be apparent from summary statistics alone.

Fig. 3 presents the violin plot, depicting the distribution of $logEC_{50}$ values (where $logEC_{50}$ is the logarithmic concentration (in mol/L) at which a compound causes a 50% toxic effect) for the ionic liquids in our dataset. The figure illustrates the following key points:

- **Consistent Distribution:** The $logEC_{50}$ distributions across the Training, Test, and Validation datasets exhibit consistency, confirming that the dataset splitting process preserved the original characteristics of the data.
- **Balanced Representation:** The visualization underscores a well-balanced and representative distribution across all subsets, facilitating robust model training, validation, and testing.
- **Enhanced Comparability:** The integration of all data splits into a single plot provides a clear and comprehensive perspective, enabling the identification of potential biases and ensuring the interpretability of the dataset.

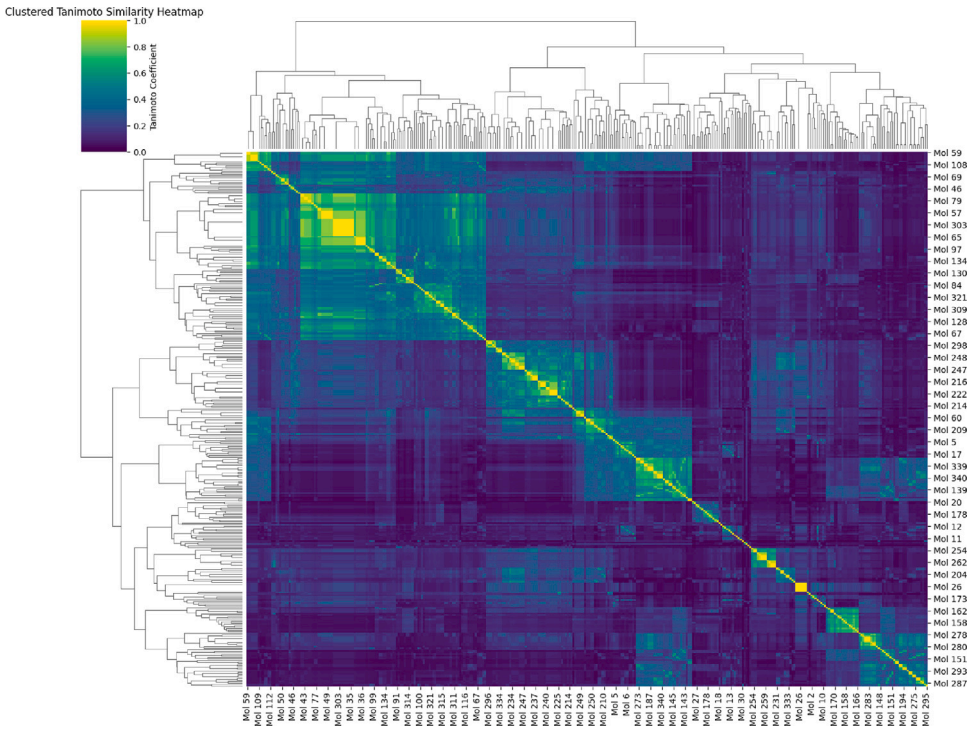

Fig. 2. Clustered heatmap of the Tanimoto similarity matrix for the dataset. Each cell represents the Tanimoto coefficient between two molecules, ranging from 0 (no similarity) to 1 (identical structures). The hierarchical clustering highlights groups of structurally similar molecules, providing insight into the diversity of the dataset.

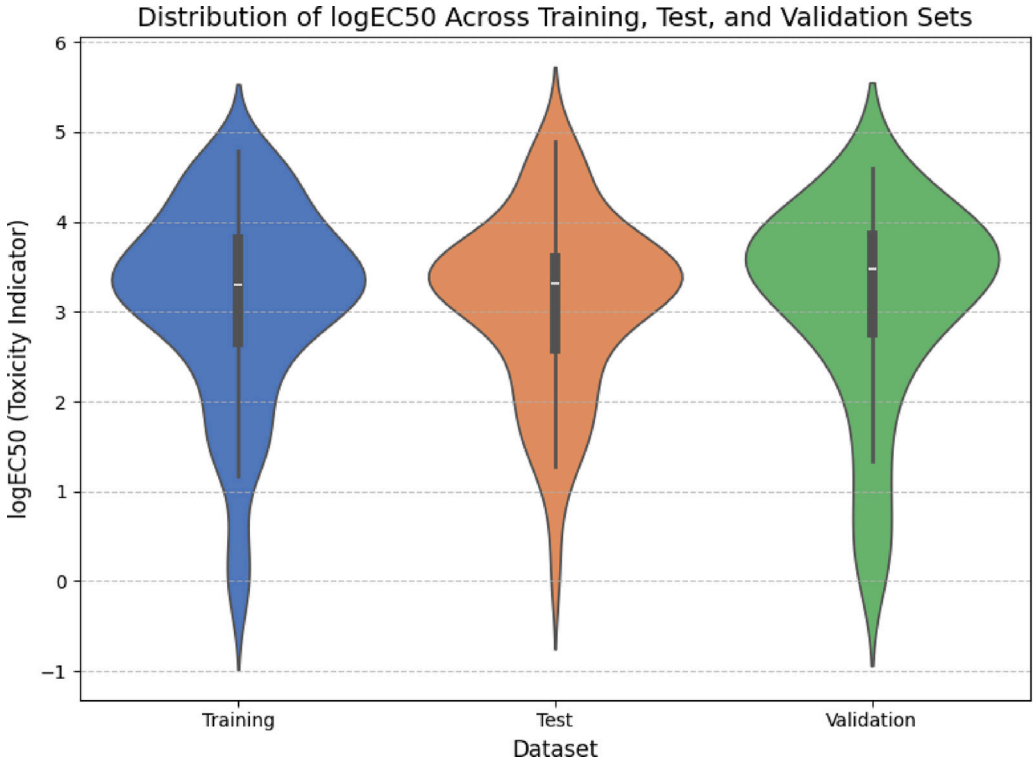

Fig. 3. Distribution of $logEC_{50}$ values across the Training, Test, and Validation datasets.

Using $logEC_{50}$ aligns directly with standard practices in computational toxicology, offering a clear and consistent metric for toxicity measurement. This approach eliminates ambiguity and ensures alignment with widely recognized methodologies in the field. Displaying the distributions for Training, Test, and Validation sets keeps the focus on the data critical to modeling, avoiding the need for subjective classifications and fostering transparency. More details can be found in [39].

2.1.4. Data augmentation

Data augmentation is a technique that enriches the diversity and size of the training data without the collection of more data. For

molecular data, molecular structures are usually represented in the form of SMILES (Simplified Molecular Input Line Entry System) strings. The following augmentation techniques have been used:

1. **Canonical SMILES Generation:**
   - The canonical SMILES is a unique representation of a molecule in a standardized format [40].
   - This standardization ensures a consistent and unique representation of molecules, preventing redundant entries and providing a reliable input for machine learning models.
2. **Random SMILES Generation:**
   - Multiple random SMILES strings were generated for each molecule by randomizing the ordering of atoms and bonds while maintaining the molecular structure [41].
   - This introduces variability in the representation, enabling the model to generalize better across different inputs.
3. **Tautomer Enumeration:**
   - Tautomers are alternate forms of a molecule that differ in the placement of hydrogen atoms and double bonds [42].
   - Using RDKit's `TautomerEnumerator`, all possible tautomers for each molecule were generated, ensuring chemically relevant variations are included in the dataset.

The implementation workflow for preparing the augmented dataset is outlined below:

1. **Read the Original Dataset:**
   - The dataset containing 355 ionic liquids with their SMILES strings and experimental $\log EC_{50}$ values was loaded.
2. **Apply Augmentation for Each Molecule:**
   - Generate the canonical SMILES.
   - Create multiple random SMILES (default: 5 variations per molecule).
   - Enumerate all possible tautomers of the molecule.
   - Combine all augmented SMILES into a unique set to avoid duplicates.
3. **Retain the Original Toxicity Values:**
   - Each augmented SMILES representation was assigned the same experimental $\log EC_{50}$ value as the original molecule to maintain consistency in the toxicity data.
4. **Compile the Augmented Dataset:**
   - The augmented dataset was saved as a new CSV file, ready for downstream machine learning tasks.

Tautomerization and its limited influence on ionic liquid toxicity. Tautomerization, a process involving hydrogen atom migration accompanied by changes in bonding arrangements (e.g., keto-enol shifts), is well-documented to affect properties such as solubility, pKa, and binding affinities in drug-like molecules [43]. However, its impact on ionic liquids is negligible due to their distinct physicochemical characteristics [44].

The primary factors influencing ionic liquid toxicity are as follows:

- **Cation-Anion Interactions:** The pairing of cations and anions plays a critical role in determining ionic liquid properties, including stability, hydrophobicity, and lipophilicity, which directly influence toxicity .
- **Structural Rigidity:** Ionic liquids generally possess rigid frameworks and well-defined charge distributions, reducing the relevance of tautomerization in their property modulation.

Each augmented SMILES string is assigned the same experimental $\log EC_{50}$ value as the original molecule from which it was derived. This approach is predicated on the assumption that the structural modifications introduced during augmentation, such as canonicalization, randomization, and tautomerization, do not fundamentally alter the core chemical properties that influence toxicity. The augmented data used in this study can be found in the Supplementary material. To enhance model robustness and generalization, the dataset was augmented with Random SMILES strings and tautomers. Random SMILES were generated to address potential biases from canonical SMILES encoding, ensuring the model focuses on core molecular properties rather than overfitting to specific patterns. We implemented a rigorous standardization process using canonical SMILES generated by RDKit's "Chem.MolToSmiles" function with the "canonical=True" argument. This process ensures that each unique molecule is represented by a single, deterministic SMILES string, thereby removing redundancy caused by different atom and bond orderings. Similarly, tautomers, as alternative structural forms of molecules, were included to capture biologically relevant variability. Assuming similar toxicity across tautomers due to their comparable physicochemical properties, consistent $\log EC_{50}$ values were assigned to all forms. The initial dataset comprised 355 instances. After data augmentation, the dataset size expanded to 2119 instances, with a total of 1744 missing values identified, significantly increasing its diversity and enhancing the robustness of the predictive models.

2.2. Featurization stage

The featurization phase of our framework involves two types of featurizers. Each featurizer processes SMILES strings and generates fixed-length base features as output. These two steps, conversion of SMILES strings to molecular fingerprints and calculation of molecular descriptors, are where the transformation of chemical information encoded in SMILES (simplified molecular input line entry system) strings into numerical representation suitable for machine learning models is critical in our framework.

2.2.1. Logarithm of the half maximal effective concentration ($logEC_{50}$)

$\log EC_{50}$ is the base-10 logarithm of the EC_{50} value, where EC_{50} represents the concentration of a substance needed to produce 50% of its maximal effect [45]. In the context of our study, it reflects the logarithmic transformation of the half-maximal effective concentration of ionic liquids (ILs) that inhibit acetylcholinesterase (AChE) enzyme activity. This transformation simplifies comparison across substances with varying potencies and is a standard metric in toxicological research [46]. Toxicity is expressed as $\log EC_{50}$ in the literature where higher values of $\log EC_{50}$ indicate lower toxicity [35,47].

2.2.2. Conversion to molecular fingerprints

The initial step in the featurization process involves converting SMILES strings into molecular fingerprints. For this study, we employed the Morgan algorithm to generate Extended-Connectivity Fingerprints (ECFP) as fixed-length bit vectors, encoding topological molecular features. Specifically, ECFP4 (with a diameter of 4) was generated using a radius parameter of 2. This choice effectively captures key molecular substructures, making it a widely accepted input for cheminformatics and machine learning applications. The workflow translated input SMILES strings into RDKit [48] molecule objects, with the Morgan algorithm subsequently used to compute the fingerprints. For invalid SMILES strings, a zero-vector of the designated length was generated.

Parameter selection for ECFP. The Extended-Connectivity Fingerprints (ECFP) were generated using a bit vector length of 2048 and a radius of 2, corresponding to the widely used ECFP4 fingerprint. These parameters were selected based on their ability to effectively encode the topological features of molecular structures while balancing computational efficiency and representational quality.

Radius (2): The two-bond radius captures local structural environments around each atom. This strikes a balance between detail and efficiency, avoiding the sparsity of smaller radii (e.g., ECFP2) and the redundancy of larger ones (e.g., ECFP6). ECFP4 has been demonstrated as effective in QSAR modeling due to its detailed representation of molecular substructures.

Length (2048): A 2048-bit vector minimizes hash collisions, ensuring sufficient resolution to distinguish diverse molecular structures. This bit length is a standard in cheminformatics, balancing representation quality and computational cost.

2.2.3. Calculation of RDKit descriptors

In addition to molecular fingerprints, a full set of RDKit molecular descriptors is computed for each SMILES string. Descriptors are used to numerically represent various chemical characteristics of the molecule. This involves converting a SMILES string to an RDKit molecule object, applying a set of pre-defined RDKit descriptor functions to that molecule, and then generating an array of values for those descriptors. The molecular descriptors were calculated using RDKit's Descriptors.descList, which provides a comprehensive set of physicochemical, topological, and electronic properties. These descriptors include molecular weight, LogP, TPSA, and molecular connectivity indices, calculated based on established methodologies [49–51]. Detailed definitions can be found in the RDKit documentation [48].

Number of RDKit descriptors. A total of 210 RDKit molecular descriptors were generated for each compound. These descriptors provide a numerical representation of various molecular properties, including topological, geometrical, and physicochemical features.

Handling of Cations and Anions: Each ionic liquid in the dataset consists of a cation and an anion, represented together as a single SMILES string. The RDKit descriptor calculation function processes the entire SMILES string as a unified entity. This approach ensures that the descriptors capture the combined structural and chemical features of both the cation and the anion, effectively reflecting their interactions and their contributions to the ionic liquid's overall properties. For cases where a SMILES string is invalid or cannot be processed, a zero vector of length 210 is assigned. This maintains consistency across the dataset and ensures compatibility with the machine learning pipeline.

If the SMILES string is invalid, a corresponding zero vector of the appropriate length is then generated. These molecular fingerprints and descriptors provide a combined, strong, and extensive representation of molecular data that improves the performance of the following machine learning tasks through the incorporation of both topological features and chemical properties, which make the models more predictive. Descriptor names are described in Supporting Information.

2.2.4. Combining molecular features for enhanced predictive modeling

The molecular fingerprints and descriptors are combined into a complete feature matrix by stacking them in one large numpy array. Finally, these two arrays are concatenated to form one single feature matrix, X, while the target variable, representing experimental $\log EC_{50}$ values, is contained in y. This enables the models to have higher predictive power through integration with rich representations of chemical data.

2.3. Feature selection using recursive feature elimination with random forest

In this study, the selection of features was a critical step to ensure that the predictive model was both effective and efficient. Given the high dimensionality of the dataset, with 2,258 potential features, it was essential to reduce this number to prevent overfitting, enhance model interpretability, and improve computational efficiency.

To identify the optimal number of features, we employed the Elbow Method in conjunction with Recursive Feature Elimination (RFE) [52] using a Random Forest Regressor [53]. The Elbow Method is a widely recognized technique for determining the point at which adding more features yields diminishing returns in terms of model performance. We evaluated the model's performance across a range of feature subsets by systematically reducing the number of features and monitoring the cross-validation R^2 score. The feature evaluation process involved testing subsets of features ranging from 50 to 2,258, with specific increments to capture the most informative features while balancing the computational cost. The R^2 scores obtained from 5-fold cross-validation were plotted against the number of features to visually identify the "elbow point"—the point where the performance gain begins to plateau.

Our analysis revealed that the model's performance peaked at around 650 features, achieving the highest R^2 score within this range (see Fig. 4). This observation indicated that 650 features provided the best balance between model complexity and predictive accuracy. Features beyond this point did not contribute significantly to model performance and potentially introduced noise, thereby justifying the decision to retain only the top 650 features.

By selecting 650 features, we ensured that the model maintained high predictive power while minimizing the risk of overfitting and reducing computational overhead. This selection process was informed by both empirical evidence from the Elbow Method and the theoretical understanding that including too many features can degrade model performance. Thus, the chosen feature set represents the most efficient and effective subset of features for predicting the toxicity of ionic liquids in our study.

2.3.1. Addressing limitations of random forest feature importance

While the Random Forest algorithm provides useful feature importance scores, it has known limitations when handling correlated features. Specifically, when two features are highly correlated, their importance can be split, leading to shared and potentially undervalued rankings. To mitigate this issue:

- **Recursive Feature Elimination (RFE):** The use of RFE ensures that features are iteratively evaluated and eliminated based on their contribution to model performance. This systematic process reduces redundancy and isolates the most informative features.
- **Cross-Validation with R^2 Scores:** To further validate feature importance, a range of feature subsets was evaluated using 5-fold cross-validation, ensuring that the selected features contribute meaningfully to toxicity prediction across different data splits.
- **Elbow Method for Optimal Feature Selection:** By employing the Elbow Method (Fig. 4), we identified the optimal subset of 650 features, balancing model performance and complexity.

Our approach minimizes the limitations associated with correlated features in Random Forest while ensuring the robustness and interpretability of the selected features. For further theoretical context on the limitations of feature importance methods, readers may refer to [54].

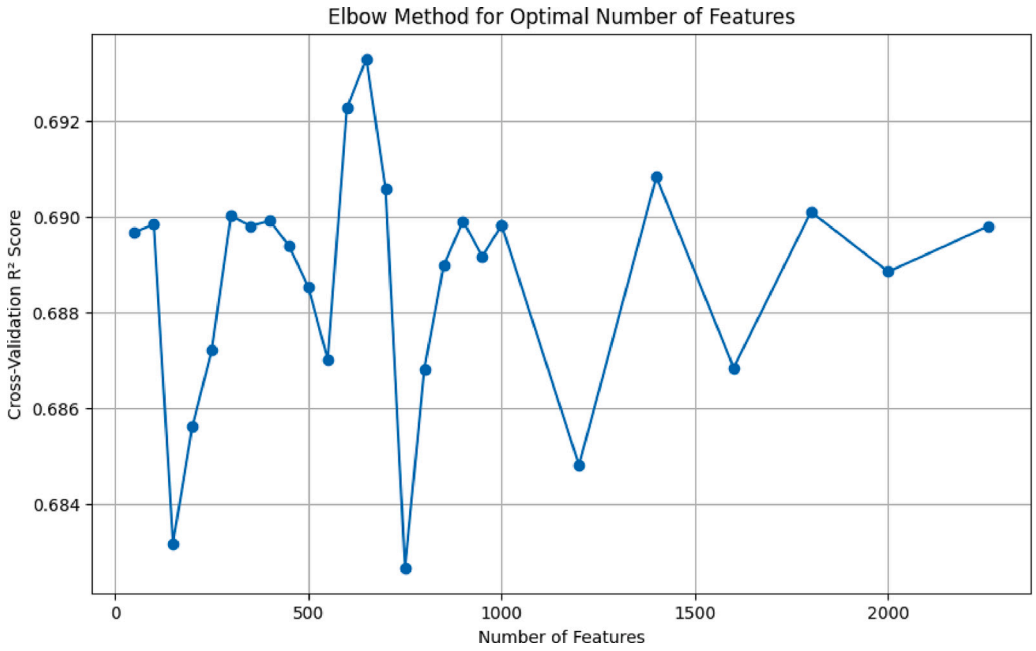

Fig. 4. Elbow Method for determining the optimal number of features. The peak R^2 score is observed around 650 features.

2.3.2. Feature selection process and reproducibility

Recursive Feature Elimination (RFE) with a Random Forest Regressor was employed to determine the optimal number of features for predicting toxicity. The process involved evaluating subsets of features ranging from 50 to 2258, with features removed in increments of 50 initially and larger steps (200) at higher ranges. A 5-fold cross-validation was conducted using the R^2 score as the performance metric.

The subset with 650 features was selected as optimal based on the following considerations:

- **Highest Mean R^2 Score:** The subset with 650 features produced the highest mean R^2 value across cross-validation folds, demonstrating slightly better generalization compared to smaller subsets.
- **Minimization of Redundancy:** Larger subsets (e.g., 1000+ features) introduced redundancy, which did not improve model performance but increased computational complexity.
- **Retention of Key Features:** Smaller subsets (e.g., 50 or 100 features), while achieving competitive R^2 scores, excluded important molecular descriptors such as `SMR_VSA5` and `fr_unbrch_alkane`. The 650-feature subset ensured that key molecular properties were retained without introducing noise.

Narrow Range of R^2 Values: Although the R^2 values for subsets above 50 features fell within a narrow range, the selection of 650 features was justified by its superior balance between performance and feature diversity. Fig. 4 shows the R^2 scores across evaluated subsets, highlighting the plateau at 650 features.

Reproducibility Measures: To ensure reproducibility:

- A fixed random seed (`random_state=42`) was used during RFE and cross-validation.

This robust process ensured that the selected features captured meaningful molecular properties and contributed effectively to model generalization.

2.4. Base model stage

This stage outlines the methodology employed for training the predictive models, including the selection and training of base models. The base learning phase comprises four distinct base models, each trained on features derived from the featurization stage.

- **Random Forest (RF)**: It is a robust base learner that builds multiple trees on bootstrapped samples with random feature subsets, enhancing predictive accuracy and reducing overfitting. It is suitable for large, high-dimensional datasets, capturing complex patterns and interactions, and provides feature importance measures for model interpretation.
- **Support Vector Regression (SVR)**: It is utilized as a base model due to its effectiveness in high-dimensional spaces and its ability to handle non-linear relationships through kernel functions. It aims to find a function with deviations within a specified margin while maximizing the margin of tolerance. Fine-tuning hyperparameters like the penalty parameter and kernel type enhances its performance, making SVR valuable for capturing complex data patterns [55].
- **Categorical Boosting (CatBoost)**: CatBoost, derived from Categorical Boosting and developed by Yandex, is used as a base model for its superior handling of categorical data and protection against overfitting. It processes categorical features natively with minimal preprocessing, improving model performance. The ordered boosting technique prevents target leakage and overfitting, especially in small datasets. Optimized for both CPU and GPU, CatBoost is efficient for large-scale datasets and excels in managing complex data patterns with automated hyperparameter tuning, making it a valuable component of the ensemble [56].
- **Chemception**: ChemCeption leverages convolutional neural networks (CNNs) to process and analyze chemical data, specifically using SMILES strings and molecular graphs. It automatically extracts features from raw chemical data, eliminating the need for extensive manual feature engineering. This enhances the predictive power of models in cheminformatics by capturing complex relationships in molecular structures. With end-to-end learning, ChemCeption allows direct learning from chemical representations, making it a powerful tool for tasks such as molecule property prediction, drug discovery, and material science [57].

2.5. Meta-model learning

A meta-model learning approach was implemented to combine predictions from multiple base models, including Random Forest, Support Vector Regressor (SVR), CatBoost, and a convolutional neural network (Chemception). Model-ensemble learning enhances predictive capabilities by combining outputs from multiple base models [58,59]. Key benefits include:

- **Improved Predictive Accuracy:** Ensembles combine the strengths of individual models, often outperforming any single model by reducing the impact of individual weaknesses.
- **Reduction in Overfitting:** Averaging or voting across models stabilizes predictions and reduces variance, particularly in small or noisy datasets.
- **Diversity in Predictions:** By leveraging the strengths of both linear models (e.g., Ridge Regression for linear trends) and non-linear models (e.g., Decision Trees for complex patterns), ensembles deliver more robust results.
- **Improved Generalization:** Ensembles generalize better to unseen data, minimizing biases from individual models.
- **Flexibility:** Combining diverse models with different architectures or training methods optimizes performance by utilizing a wide range of data characteristics.

The final meta-model was built using XGBoost, introduced by Chen and Guestrin [60], which aggregated the outputs of these base models to improve predictive accuracy. XGBoost's efficacy has been widely acknowledged in molecular property and toxicity prediction tasks. Its ability to capture non-linear relationships, coupled with robust regularization and scalability, makes it an indispensable tool for cheminformatics applications.

The dataset consisted of 355 samples and 650 features. The Chemception model, a key component of the meta-model, comprised a Conv1D layer, MaxPooling1D layer, and two Dense layers, with a total of 1,034,085 trainable parameters. This architecture allowed the model to capture complex patterns necessary for accurate predictions. To prevent overfitting, 5-fold cross-validation was used during training, and hyperparameters were optimized using `GridSearchCV` for XGBoost and `RandomizedSearchCV` for Chemception. Regularization terms (L1 and L2) were applied in XGBoost to penalize complexity and promote generalization.

Overall, the meta-model effectively integrated the strengths of multiple models, resulting in a robust and accurate predictive framework that was well-tuned and generalizable.

2.6. Hyperparameter tuning

In this research, GridSearchCV was employed for hyperparameter tuning to optimize model performance. GridSearchCV, a cross-validated exhaustive search method from Sklearn, systematically explores the hyperparameter space to identify optimal parameters. This approach enhances predictive accuracy and robustness by using cross-validation to evaluate each combination, providing reliable performance estimates and avoiding overfitting. For a comprehensive understanding, please refer to the following paper [61].

2.7. Model evaluation

To evaluate the performance of the meta-model on the test set, several key statistical metrics were employed. These metrics included the Coefficient of Determination (R^2), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Pearson Correlation Coefficient (r). The following formulas were used to calculate these metrics:

R-squared (R^2)

See Eq. (1)

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

Mean absolute error (MAE)

See Eq. (2)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

Root mean square error (RMSE)

See Eq. (3)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

Pearson correlation coefficient (r)

See Eq. (4)

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4}$$

In these formulas, y_i refers to the observed toxicity values from the test set, while $\hat{y}_i$ represents the predicted values generated by the meta-model. The term n is the total number of data points used in the evaluation. To account for different magnitudes in y_i, the metrics are calculated using the raw observed values without scaling, as this study focuses on directly comparing predictions to actual values. The RMSE and MAE provide insight into the model's error magnitude, with RMSE being more sensitive to larger errors due to the squaring of residuals. A small scale in these metrics indicates that the errors are generally low, whereas a large scale suggests more significant discrepancies between predicted and observed values.

Additionally, the standard deviation of the errors was calculated to assess the consistency of the model's predictions. A low standard deviation indicates that the errors are tightly clustered around the mean error, suggesting reliable performance across different data points. The Pearson Correlation Coefficient, which measures the linear relationship between predicted and observed values, further supports the robustness of the model's predictions. The scikit-learn package [62] in Python was utilized to implement these evaluation metrics, ensuring standardized and reliable calculations across all test scenarios.

3. Implementation details

Python libraries 'RDkit', 'NumPy' [63], and 'scikit-learn' were used in this study.

3.1. Process of converting SMILES strings to molecular fingerprints

- **Molecular Fingerprint Conversion**: The "smiles_to_fingerprint" function utilized RDKit to transform SMILES strings into molecular fingerprints through the Morgan algorithm. It returned the fingerprint as a NumPy array or a zero bit vector if the molecule conversion fails. Consequently, 2048 molecular fingerprints were computed.
- **Descriptor Calculation**: The "calculate_descriptors" function computed molecular descriptors from a SMILES string using 'RDKit'. It returned the descriptors as a 'NumPy' array or a zero array if the molecule conversion fails. Consequently, 210 molecular descriptors were computed.

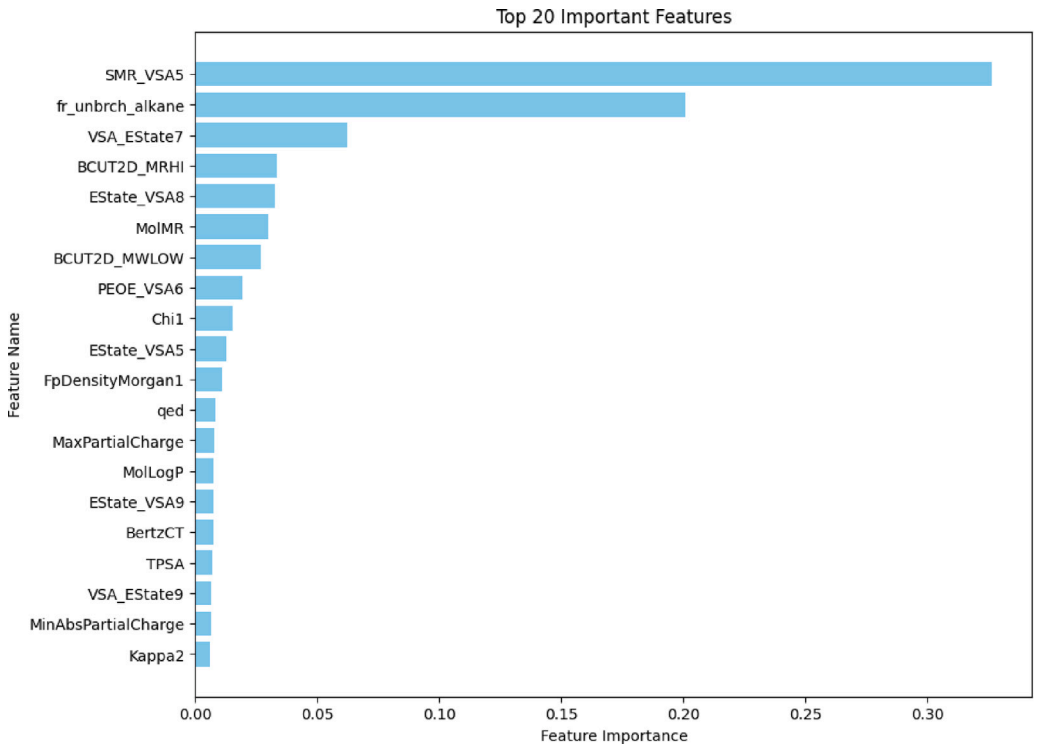

Fig. 5. The top 20 features identified as the most important for predicting toxicity include descriptors such as `SMR_VSA5` (van der Waals surface area weighted by refractivity), `MolMR` (molecular refractivity), and `fr_unbrch_alkane` (unbranched alkane fragments). These descriptors provide insights into the physicochemical properties of ionic liquids, such as dispersion forces, molecular flexibility, and electronic activity, which are critical for understanding their interaction with acetylcholinesterase enzymes and resulting toxicity. For example, high values of `SMR_VSA5` suggest strong dispersion interactions with biological targets, while features like `EState_VSA8` and `PEOE_VSA6` indicate electronic activity in specific molecular regions.

- **Feature Matrix Construction**: Fingerprints and descriptors were extracted from the dataset and converted to NumPy arrays. These arrays were concatenated along the horizontal axis to form a combined feature matrix X. The target variable, representing experimental $\log EC_{50}$ values, was extracted and stored in y. Ultimately, a total of 2258 features were collected.
- **Imputation of Missing Values**: In the literature, there are several proposed methods to impute missing values: mean imputation, median imputation, K nearest neighbor (KNN) imputation, predictive mean matching, Bayesian Linear Regression (norm), Linear Regression, non-Bayesian (norm. nob), and random sample [64]. Among these recommended methods, Mean imputation is the simplest and quickest imputation method. To handle missing values in the feature matrix X, the SimpleImputer class from scikit-learn was employed with the strategy set to 'mean'. In the original dataset, the number of missing values was 292, whereas in the augmented dataset, the number of missing values increased to 1744. The imputer was fitted to X, and the missing values were replaced with the mean value of their respective feature columns.

3.2. Feature selection with RFE

- **Initial Feature Evaluation and Selection**: To identify the most important features, RFE was applied using a RandomForestRegressor with 100 estimators as the underlying model. The process began with a broad evaluation of feature subsets, where the optimal number of features was determined using cross-validation with R^2 as the performance metric. This evaluation involved iteratively eliminating less important features and identifying the subset that provided the highest cross-validation score. The optimal number of features was found to be 650, based on the "elbow method" from the plotted cross-validation scores. The RFE process was then refined to select this optimal number of features, resulting in a new feature matrix ($X_{selected}$) that included only the most relevant features.
- **Feature Importance Ranking and Visualization**: The importance of the selected features was subsequently evaluated and ranked according to the RandomForestRegressor model's feature importance scores. The top features were visualized in a bar plot, highlighting the most significant contributors to the model's predictive performance. Fig. 5 illustrates the top 20 significant features, as determined by the RandomForestRegressor model utilizing Recursive Feature Elimination (RFE).

3.3. Data split

The dataset was split randomly into training and testing sets, with 80% of the data used for training and 20% for testing. To ensure reproducibility, the splitting process was controlled using a fixed *random_state value of 42*. The same splitting strategy was applied to the augmented dataset.

3.4. Hyperparameter tuning of base models

This section details the creation, hyperparameter tuning, and selection of several base models used in the study: RandomForestRegressor, Support Vector Regressor (SVR), CatBoost Regressor, and Chemception (a specialized convolutional neural network designed for chemical data analysis) (see Tables 2 and 3).

GridSearchCV was used for the RandomForestRegressor, SVR, and CatBoost Regressor with 5-fold cross-validation (`cv=5`) and parallel computation (`n_jobs=-1`). RandomizedSearchCV was employed for the Chemception model with 3-fold cross-validation (`cv=3`). GridSearchCV with 5-fold cross-validation was considered as an appropriate approach to RandomForestRegressor, SVR, and CatBoost Regressor in balancing the bias–variance in model selection during hyperparameter

Table 2
Hyperparameter grids and best parameters for RandomForestRegressor, SVR, and CatBoost Regressor.

Model	Type	Parameters
RandomForestRegressor	Hyperparameter Grid	n_estimators: [100, 200] max_depth: [None, 10, 20] min_samples_split: [2, 5]
	Best Parameters	n_estimators: 200 max_depth: None min_samples_split: 2
Support Vector Regressor (SVR)	Hyperparameter Grid	kernel: [‘rbf’] C: [0.1, 1, 10] gamma: [‘scale’, ‘auto’]
	Best Parameters	kernel: ‘rbf’ C: 10 gamma: ‘auto’
CatBoost Regressor	Hyperparameter Grid	depth: [6, 8] learning_rate: [0.1, 0.01] iterations: [100, 200]
	Best Parameters	depth: 8 learning_rate: 0.1 iterations: 200

Table 3
Hyperparameter grid and best parameters for the Chemception Model.

Model	Type	Parameters
Chemception Model	Layers	Convolutional Layer with ReLU activation MaxPooling Layer Flatten Layer Dense Layer with ReLU activation Output Layer for regression
	Hyperparameter Grid	filters: [32, 64] kernel_size: [3, 5] pool_size: [2, 3] dense_units: [50, 100] epochs: [10, 20] batch_size: [10, 20] learning_rate: [0.001, 0.01]
	Best Parameters	filters: 64 kernel_size: 3 pool_size: 2 dense_units: 100 epochs: 20 batch_size: 10 learning_rate: 0.001

tuning, which was computationally efficient and conforms to most widely accepted standards in machine learning. In contrast, for the Chemception model, a 3-fold cross-validated RandomizedSearchCV was preferred because training deep neural networks is computationally expensive, yet this fold size guarantees a fair evaluation of model performance for most problems when conducting hyperparameter tuning. This will allow model evaluation to be both reliable and efficient.

3.5. Creation of meta-features through cross-validated predictions

To improve the ensemble model’s predictive accuracy, cross-validated predictions from each optimized base model were generated and utilized as meta-features. In the context of stacked ensemble learning, ‘meta features’ refer to the predictions generated by base models during the training and testing phases. These features serve as inputs to a higher-level model, known as the meta-model. For instance, in our study, predictions from models such as RandomForest, SVR, CatBoost, and Chemception were stored as meta features. These features encapsulate the diverse learning patterns captured by the base models and enable the meta-model to learn and refine the final prediction. An empty array, ‘‘meta_features’’, was created to store these predictions. Using 5-fold cross-validation, predictions were obtained for the RandomForestRegressor, Support Vector Regressor (SVR), and CatBoost Regressor. For the Chemception model, predictions were also generated using 5-fold cross-validation, with careful reshaping to fit into the meta-feature array. This approach ensured that predictions for each training sample were made by models that had not encountered the sample during training, thus preventing overfitting and providing a dependable set of meta-features. These meta-features were then used as inputs for the meta-model in the stacking ensemble, leveraging the combined strengths of the base models to enhance overall predictive performance.

3.6. Hyperparameter tuning for the meta-model (XGBoost)

To optimize the meta-model in the stacking ensemble, an XGBoost regressor was fine-tuned using GridSearchCV. The hyperparameter grid and the best parameters identified are summarized in Table 4. The hyperparameter learning rate has a value, 0.2, which is higher than the typical values in such ranges. This value was obtained through vigorous hyperparameter tuning and found to be optimal. We experimented with a wide setting range going from as low as from 0.00001 up to 0.005. Nevertheless, from the cross-validation results, it was depicted that the learning rate equal to 0.2 offered the best performance level for this specific dataset. Similarly, other hyperparameters were also tuned over broad ranges to ensure optimal settings. Values, like the learning rate, were chosen empirically following systematic testing and not arbitrarily. These values best matched data characteristics – empirical

Table 4
Hyperparameter grid and best parameters for the XGBoost meta-model.

Parameter	Values	Best value
n_estimators	[100, 200]	100
max_depth	[3, 6, 9]	6
learning_rate	[0.01, 0.1, 0.2]	0.2
subsample	[0.7, 0.8, 1.0]	0.8
colsample_bytree	[0.7, 0.8, 1.0]	0.8

trends – that focused attention toward dataset-specific tuning rather than standard published ranges.

The best XGBoost regressor configuration was:

Best XGBoost Regressor Configuration

```
XGBRegressor(base_score=None, booster=None,
callbacks=None,
colsample_bylevel=None, colsample_bynode=None,
colsample_bytree=0.8, device=None,
early_stopping_rounds=None,
enable_categorical=False, eval_metric=None,
feature_types=None,
gamma=None, grow_policy=None, importance_type=None,
interaction_constraints=None, learning_rate=0.2,
max_bin=None,
max_cat_threshold=None, max_cat_to_onehot=None,
max_delta_step=None, max_depth=6, max_leaves=None,
min_child_weight=None, missing=nan,
monotone_constraints=None,
multi_strategy=None, n_estimators=100, n_jobs=None,
num_parallel_tree=None, random_state=42, ...)
```

This process ensured the meta-model effectively combined the strengths of the base models for enhanced predictive performance.

3.7. Creation of meta-features for the test set

To test the ensemble model on the test set, the meta_features were generated from all optimized base model predictions. An empty list, test_meta_features, was initialized to hold it. Results from the test set of "RandomForestRegressor", "SVR", and "CatBoost Regressor" were placed in the first three columns of test_meta_features. The "Chemception" test set was also reshaped as required for its predictions, and their predictions were placed in the fourth column. This was done with the assurance that the meta-features were derived from well-tuned models, thus providing reliable inputs for the meta-model in the stacking ensemble to increase predictive performance.

3.8. Final prediction with the meta-model

After generating meta-features from the optimized base models, the best XGBoost regressor was used to make the final predictions. The `test_meta_features` array, containing the base models' predictions, served as input for the XGBoost meta-model. The optimized XGBoost regressor then produced the final test set predictions, leveraging the strengths of all base models to enhance accuracy.

4. Results and discussion

4.1. Comparative model performance analysis

This section offers a comparison of the ensemble model's performance with and without data augmentation, assessing the impact of data augmentation by examining the alignment between predicted and actual values, along with the residuals' distribution. As summarized in Table 5, the performance metrics clearly demonstrate the advantages of data augmentation. The ensemble model with data augmentation shows a significant reduction in Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), alongside substantial improvements in R-squared and Pearson Correlation values, indicating a stronger fit and enhanced predictive accuracy.

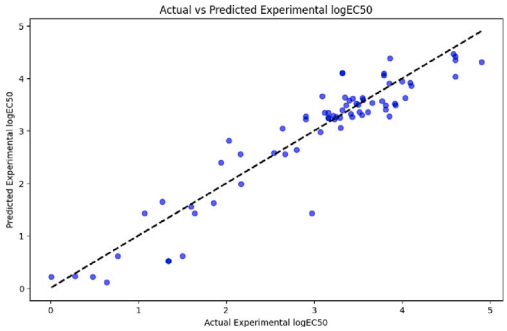

Fig. 6. Actual vs Predicted Experimental $logEC_{50}$ without data augmentation.

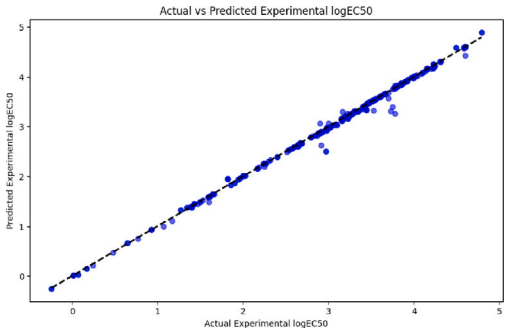

Fig. 7. Actual vs Predicted Experimental $logEC_{50}$ with data augmentation.

Table 5
Performance metrics comparison for the ensemble model with and without data augmentation.

Metric	Without augmentation	With augmentation
Root Mean Squared Error (RMSE)	0.383646	0.055850
Mean Absolute Error (MAE)	0.295523	0.020458
R-squared	0.878808	0.996990
Pearson Correlation	0.940181	0.998510

4.1.1. Actual versus predicted values

The relationship between the actual and predicted experimental $logEC_{50}$ values is illustrated in Figs. 6 and 7. Fig. 6 shows the outcomes for the model without data augmentation. The scatter plot demonstrates that the predicted values correspond fairly well with the actual values, though some deviations from the diagonal line are noticeable, particularly at higher $logEC_{50}$ values. This suggests that while the model performs adequately, there is room for improvement in predicting the more extreme values.

On the other hand, Fig. 7 presents the results for the model with data augmentation. The scatter plot reveals a much tighter clustering of points around the diagonal line, indicating a significant improvement in prediction accuracy. The improved alignment suggests that the model with data augmentation more effectively captures the underlying patterns in the data, resulting in more reliable predictions across the full range of experimental values.

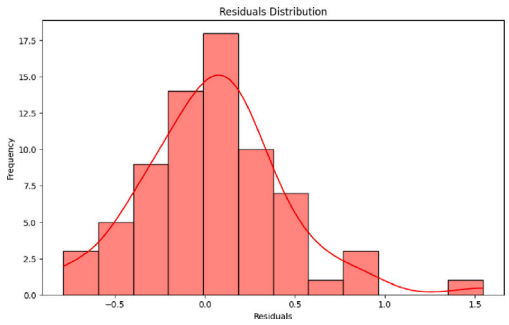

Fig. 8. Residuals Distribution without data augmentation.

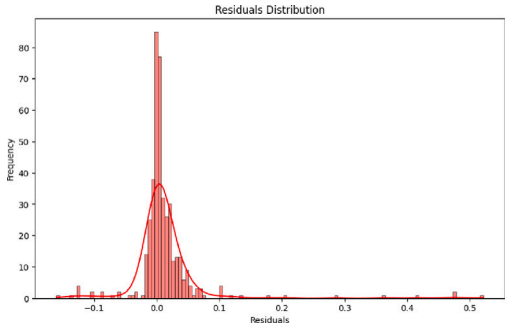

Fig. 9. Residuals Distribution with data augmentation.

4.1.2. Residuals distribution

The residuals distribution for both models is illustrated in Figs. 8 and 9. Fig. 8, which corresponds to the model without data augmentation, displays a broader distribution of residuals centered around zero. Although the residuals follow a normal distribution, the wider spread indicates greater variability in the model's errors, implying that the predictions are less consistent.

Conversely, Fig. 9 shows the residual distribution for the model with data augmentation. This distribution is much narrower and more sharply centered around zero, indicating a significant reduction in prediction errors. The decreased variability in residuals suggests that the augmented model not only enhances accuracy but also improves the consistency and reliability of the predictions.

4.2. Confidence interval analysis and statistical significance

To evaluate the precision and accuracy of the predictive models, both with and without data augmentation, an extensive statistical analysis was conducted. This involved calculating 95% confidence intervals for each predicted value and performing a paired t-test to compare the predictions against the actual experimental data.

Without Data Augmentation:

The model without data augmentation generated a mean prediction value of 2.9527, with a 95% confidence interval spanning from 2.6760 to 3.2294. Specific confidence intervals, such as (2.944, 3.488) and (3.230, 3.774), were relatively narrow, suggesting a moderate level of certainty in these predictions. However, the standard deviation of the errors was 0.396, indicating greater variability in the predictions. The paired t-test resulted in a t-statistic of −1.053 and a p-value of 0.296, which is higher than the standard significance level of 0.05. This indicates that there is no statistically significant difference between the predicted and actual values. While the model's predictions are generally consistent with the experimental data, the confidence in these predictions is somewhat lower due to the higher variability.

With Data Augmentation:

The model with data augmentation exhibited a marked improvement in prediction accuracy. The mean prediction value slightly increased to 3.0965, with a more precise 95% confidence interval ranging from 2.9994 to 3.1936. The confidence intervals for specific predictions, such as (3.468, 3.662) and (3.213, 3.408), were narrower, indicating a higher degree of certainty. Additionally, the standard deviation of the errors significantly decreased to 0.086, demonstrating less variability and more consistent predictions. The paired t-test for the augmented model yielded a t-statistic of −3.799 and a p-value of 0.00017, which is well below the 0.05 significance threshold. This result highlights a statistically significant difference between the predicted and actual values, with the augmented model showing a much stronger alignment with the experimental data (see Fig. 10).

P-Value Analysis: The p-value analysis of both models highlights the statistical significance of their predictions. For the model without data augmentation, significant p-values ($p < 0.05$) are scattered, indicating occasional alignment with actual values. In contrast, the model with data augmentation exhibits a concentrated area of significant p-values, reflecting a more reliable prediction performance. This difference underscores the impact of data augmentation in enhancing the model's accuracy and consistency.

4.2.1. Methodological transparency

To evaluate the reliability and robustness of the model's predictions, we conducted confidence interval (CI) analysis and statistical significance testing. These methods provide quantitative measures of prediction uncertainty and ensure the validity of the results.

1. Confidence Interval Analysis: Confidence intervals were calculated to quantify the uncertainty and reliability of model predictions. The process involved the following steps:

- **Mean and Standard Error Calculation:** For each prediction, the mean ($\bar{y}$) and the standard error (SE) were computed. The standard error was derived using the formula:

$$SE = \frac{\sigma}{\sqrt{n}}$$

  where σ is the standard deviation of the predictions, and n is the sample size.
- **Confidence Interval Estimation:** A 95% confidence interval (CI) was calculated using the t-distribution to account for small sample sizes:

$$CI = \bar{y} \pm t_{(1-\alpha/2,df)} \cdot SE$$

  Here, $t_{(1-\alpha/2,df)}$ is the critical t-value for the desired confidence level, and df represents the degrees of freedom.
- **Interpretation:** The CI provided a range within which the true prediction values are expected to fall with 95% confidence. This analysis helped assess the reliability of predictions across various EC_{50} ranges, particularly for extreme values.

2. Statistical Significance Testing: Statistical significance testing was conducted to evaluate model performance and ensure the robustness of results:

- **Paired t-Test:** A paired t-test was performed between the predicted values and the experimental $\log EC_{50}$ values to assess predictive accuracy. This test evaluated whether the mean difference between predicted and actual values was statistically significant.
- **Key Metrics:** The t-statistic and p-value were reported, with a p-value below 0.05 considered statistically significant. This indicated that the model's predictions were unlikely to be due to random chance.

3. Purpose and Benefit to the Community: These methods were employed to strengthen the robustness and reliability of model predictions:

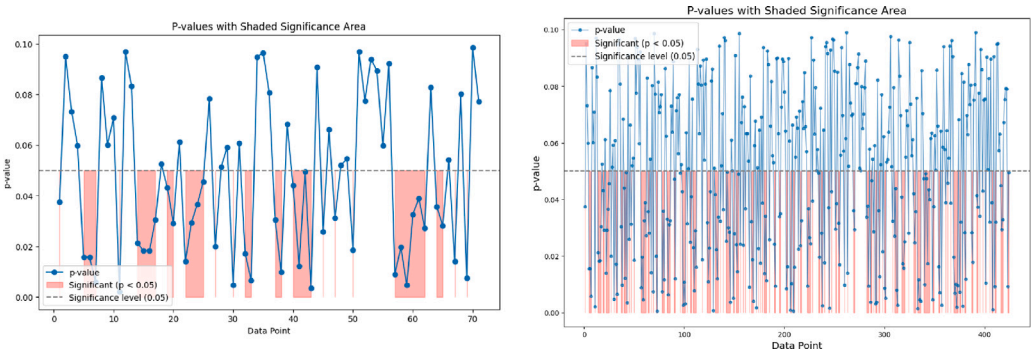

Fig. 10. P-values across various data points for models without (left) and with (right) data augmentation. The shaded regions represent significant areas where p-values are below the 0.05 significance level (dashed line). The model with data augmentation shows a more consistent and statistically significant alignment of predictions with actual values, as evidenced by the higher density of p-values below the threshold.

Table 6
Comparison of model performance metrics.

Model	RMSE	MAE	R-squared	Pearson correlation
FNN (Training Set) [22]	0.2906	0.2111	0.9227	–
FNN (Test Set) [22]	0.3732	0.3028	0.8917	–
SVM (Training Set) [22]	0.2787	0.1762	0.9289	–
SVM (Test Set) [22]	0.3204	0.2628	0.9202	–
MLR [69]	0.51	–	0.77	–
MLR [70]	0.43	0.34	–	–
Proposed Model (without DA)	0.383646	0.295523	0.878808	0.940181
Proposed Model (with DA)	0.060812	0.024410	0.996432	0.998301

- **Quantifying Model Uncertainty:** CI analysis provides a way to report not just point predictions but also the associated uncertainty. This is particularly valuable for datasets with high variability, as seen in studies involving ionic liquids.
- **Ensuring Robustness:** Statistical significance testing validates the reliability of reported results, serving as a benchmark for future ML studies.
- **Fostering Transparency:** By detailing these methods, we provide a framework for other researchers to adopt, promoting transparency and reproducibility in ML-based research on ionic liquids.

The use of confidence interval analysis and statistical significance testing strengthens the robustness and reliability of model predictions. These methods not only validate our results but also offer a reproducible approach that benefits the broader research community. For readers seeking more detailed information on confidence interval analysis and statistical significance testing, we recommend consulting established resources such as [65,66] for confidence intervals, and [67, 68] for statistical testing methods.

4.3. Model comparison

The proposed model, particularly with the incorporation of data augmentation, shows a remarkable enhancement in predicting the toxicity of ionic liquids compared to earlier models. Without data augmentation, the model achieves an RMSE of 0.38, an MAE of 0.29, an R-squared value of 0.87, and a Pearson correlation of 0.94. However, when data augmentation is applied, the model's performance significantly improves, achieving an RMSE of 0.06, an MAE of 0.02, an R-squared value of 0.99, and a Pearson correlation of 0.99. These results suggest that the data-augmented model not only surpasses the Feed-forward Neural Network (FNN) and Support Vector Machine (SVM) models presented by Wang et al. (2020), but also significantly exceeds the Multiple Linear Regression (MLR) models by Sosnowska et al. (2017) and Wu et al. (2020). The notable decrease in RMSE and MAE, coupled with the almost perfect R-squared and Pearson correlation values, underscores the robustness and predictive accuracy of the proposed model, establishing it as a superior tool for the computer-aided molecular design of environmentally friendly ionic liquids (see Table 6).

4.4. Discussion of top features

The top 20 features identified during the modeling process, as shown in Fig. 5, provide significant insights into the molecular characteristics influencing toxicity. Among these, the most important descriptors are:

SMR_VSA5: This feature represents the van der Waals surface area (VSA) contributions weighted by molar refractivity, capturing dispersion forces.

fr_unbrch_alkane: The count of unbranched alkane fragments, which reflects molecular flexibility.

VSA_EState7: An electrotopological state descriptor summarizing both electronic and geometric properties.

Dispersion Forces (SMR_VSA5): Molecular regions with high refractivity are associated with strong dispersion interactions. These regions can enhance molecular binding to biological targets, potentially increasing toxicity.

Fragment-Based Descriptors (fr_unbrch_alkane): Unbranched alkanes, characterized by reduced steric hindrance, can influence bioavailability and membrane permeability, impacting how the molecule interacts with biological systems.

Electrotopological and Surface Area Descriptors (VSA_EState7): These descriptors highlight regions of significant electronic activity and molecular reactivity. Such regions often correlate with interactions with enzymes, such as acetylcholinesterase, directly affecting toxicity.

4.5. Impact of data augmentation on high EC50 values

The observed improvement in model performance for compounds with high EC_{50} values following data augmentation can be attributed to the following factors:

1. **Balancing the Dataset Distribution:** High EC_{50} values, often associated with low-toxicity compounds, are typically underrepresented in toxicity datasets. This imbalance can result in a model biased toward more common lower EC_{50} values. Data augmentation methods, such as random SMILES generation and

tautomer enumeration, enriched the dataset by introducing diverse yet valid chemical representations. This increased representation of high EC_{50} compounds enabled the model to better capture patterns associated with low-toxicity compounds.

2. **Increased Chemical Diversity:** Augmentation techniques expanded the dataset with structurally and chemically diverse samples. This diversity:
   - Highlighted subtle structural features relevant to high EC_{50} values that might be underexplored in the original dataset.
   - Improved the model's ability to generalize across sparsely populated regions of the chemical space, particularly those corresponding to high EC_{50} values.
3. **Improved Representation of Low-Toxicity Patterns:** High EC_{50} values are indicative of low-toxicity compounds, which may share distinct structural or physicochemical properties (e.g., high molecular weight, low lipophilicity). Data augmentation generated more examples of these specific patterns, enabling the model to:
   - Differentiate low-toxicity compounds from high-toxicity ones more effectively.
   - Capture features that were underrepresented in the original dataset.
4. **Mitigation of Overfitting:** By introducing variability into the dataset, data augmentation inherently reduces overfitting. This forces the model to focus on generalizable features rather than memorizing specific instances. For high EC_{50} values, this variability enhanced the model's ability to identify underlying trends and structural characteristics associated with low toxicity.
5. **Amplifying Signal for Sparsely Represented Regions:** In the original unaugmented dataset, high EC_{50} compounds contributed less to the overall loss function during training due to their smaller representation. Augmentation amplified the signal from these sparsely represented regions, ensuring that the model learned effectively from them, thereby improving predictive performance.

These factors collectively contributed to the observed improvement in predictive performance for high EC_{50} values, reducing bias toward more common toxicity levels and improving the model's ability to generalize across the entire EC_{50} spectrum.

5. Conclusion

This study presents a cutting-edge meta-ensemble learning framework designed to predict the toxicity of ionic liquids (ILs) with remarkable accuracy, utilizing molecular descriptors and fingerprints. By combining the strengths of multiple machine learning models – such as Random Forest, Support Vector Regression, CatBoost, and Chemception – with an XGBoost meta-classifier, the framework achieves notable improvements compared to traditional approaches. Efficiency and precision are further enhanced through Recursive Feature Elimination for feature selection and hyperparameter tuning via GridSearchCV.

Data augmentation techniques, including random SMILES generation, canonical SMILES, and tautomer enumeration, play a pivotal role in refining model performance. These methods reduce prediction errors and enhance consistency. The framework demonstrates significant improvements in RMSE, MAE, R^2, and Pearson correlation coefficients over models that do not employ augmentation, highlighting its robustness and reliability.

Beyond its technical contributions, this research offers a powerful tool for computer-aided molecular design of environmentally sustainable ILs, aligning with the principles of green chemistry. Moreover, it provides a reproducible framework for advancing QSAR modeling by integrating data-driven insights with molecular property prediction.

Future developments could extend this framework by exploring new data augmentation methods, incorporating experimental validation, and applying it to broader molecular datasets. Additionally, a comparison of the performances of the base models can be investigated to see their strengths separately as well. Such efforts would further establish its versatility and value in cheminformatics and environmental sciences.

CRediT authorship contribution statement

Safa Sadaghiyanfam: Writing – original draft, Validation, Software, Methodology, Formal analysis. **Hiqmet Kamberaj:** Writing – review & editing, Writing – original draft, Conceptualization. **Yalcin Isler:** Writing – review & editing, Writing – original draft, Supervision, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was also supported by Izmir Katip Celebi University Scientific Research Council Agency as project number 2024-TDR-FEBE-0024 for Safa Sadaghiyanfam's doctoral thesis studies.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.aichem.2025.100087. We share Python source code files and additional data files for the dataset used, the "Tanimato similarity matrix", the "augmented data file", and the "fingerprint descriptors" at https://github.com/islerya/ionic-liquid-toxicity-with-data-augmentation.

References

[1] D. Rogers, M. Hahn, Extended-connectivity fingerprints, J. Chem. Inf. Model. 50 (5) (2010) 742–754, http://dx.doi.org/10.1021/ci100050t.
[2] Z. Lei, B. Chen, Y.M. Koo, M.D. R., Introduction: Ionic liquids, Chem. Rev. 117 (2017) 6633–6635, http://dx.doi.org/10.1021/acs.chemrev.7b00246.
[3] Y. Zhang, Z. Su, K. Xue, J. Xing, D. Fan, J. Qi, Z. Zhu, Y. Wang, Efficient separation of methyl tert-butyl ether using ionic liquids from computational thermodynamics to process intensification, Ind. Eng. Chem. Res. 61 (48) (2022) 17631–17643, http://dx.doi.org/10.1021/acs.iecr.2c03056.
[4] H. Ran, J. Wang, A.A. Abdeltawab, X. Chen, G. Yu, Y. Yu, Synthesis of polymeric ionic liquids material and application in CO2 adsorption, J. Energy Chem. 26 (2017) 909–918, http://dx.doi.org/10.1016/j.jechem.2017.06.001.
[5] E. Piatti, L. Guglielmero, G. Tofani, A. Mezzetta, L. Guazzelli, F. D'Andrea, S. Roddaro, C.S. Pomelli, Ionic liquids for electrochemical applications: correlation between molecular structure and electrochemical stability window, J. Mol. Liq. 364 (2022) 120001, http://dx.doi.org/10.1016/j.molliq.2022.120001.
[6] R.A. Sheldon, Biocatalysis in ionic liquids: state-of-the-union, Green Chem. 23 (2021) 8406–8427, http://dx.doi.org/10.1039/D1GC03145G.
[7] M.J. Earle, J.M. Esperança, M.A. Gilea, J.N. Canongia Lopes, L.P. Rebelo, J.W. Magee, K.R. Seddon, J.A. Widegren, The distillation and volatility of ionic liquids, Nature 439 (2006) 831–834, http://dx.doi.org/10.1038/nature04451.
[8] A.A. Quintana, A.M. Sztapka, V.D. Santos Ebinuma, C. Agatemor, Enabling sustainable chemistry with ionic liquids and deep eutectic solvents: a fad or the future? Angew. Chem. Int. Ed. 61 (2022) e202205609, http://dx.doi.org/10.1002/anie.202205609.
[9] P.T. Anastas, M.M. Kirchhoff, Origins, current status, and future challenges of green chemistry, Acc. Chem. Res. 35 (9) (2002) 684–694, http://dx.doi.org/10.1021/ar010065m.
[10] S.S. de Jesus, R. Maciel Filho, Are ionic liquids eco-friendly? Renew. Sustain. Energy Rev. 157 (2022) 112039, http://dx.doi.org/10.1016/j.rser.2021.112039.

[11] C. Pretti, C. Chiappe, D. Pieraccini, M. Gregori, F. Abramo, G. Monni, L. Intorre, Acute toxicity of ionic liquids to the zebrafish (Danio rerio), Green Chem. 8 (2006) 238–240, http://dx.doi.org/10.1039/B511554J.

[12] B. Pawłowska, A. Telesiński, R. Biczak, Phytotoxicity of ionic liquids, Chemosphere 237 (2019) 124436, http://dx.doi.org/10.1016/j.chemosphere.2019.124436.

[13] R. Arunkumar, A.N. Abraham, R. Shukla, C.J. Drummond, T.L. Greaves, Cytotoxicity of protic ionic liquids towards the HaCat cell line derived from human skin, J. Mol. Liq. 314 (2020) 113602, http://dx.doi.org/10.1016/j.molliq.2020.113602.

[14] M. Turek, B. Pawłowska, E. Rozycka-Sokołowska, R. Biczak, J. Skalik, K. Owsianik, B. Marciniak, P. Bałczewski, Ecotoxicity of ammonium chlorophenoxyacetate derivatives towards aquatic organisms: Unexpected enhanced toxicity upon oxygen by sulfur replacement, J. Hazard. Mater. 382 (2020) 121086, http://dx.doi.org/10.1016/j.jhazmat.2019.121086.

[15] N.V. Plechkova, K.R. Seddon, Applications of ionic liquids in the chemical industry, Chem. Soc. Rev. 37 (1) (2008) 123–150, http://dx.doi.org/10.1039/B006677J.

[16] J. Hemmerich, E.G. F., In silico toxicology: From structure–activity relationships towards deep learning and adverse outcome pathways, Wiley Interdiscip. Reviews: Comput. Mol. Sci. 10 (4) (2020) e1475, http://dx.doi.org/10.1002/wcms.1475.

[17] A.B. Raies, V.B. Bajic, In silico toxicology: computational methods for the prediction of chemical toxicity, Wiley Interdiscip. Reviews: Comput. Mol. Sci. 6 (2) (2016) 147–172, http://dx.doi.org/10.1002/wcms.1240.

[18] D. Krewski, D. Acosta, M. Andersen, H. Anderson, J.C. Bailar, K. Boekelheide, R. Brent, G. Charnley, V.G. Cheung, S. Green, K.T. Kelsey, Toxicity testing in the 21st century: a vision and a strategy, J. Toxicol. Environ. Heal. Part B: Crit. Rev. 13 (2010) 51–138, http://dx.doi.org/10.1080/10937404.2010.483176.

[19] P. Zhu, X. Kang, Y. Zhao, U. Latif, H. Zhang, Predicting the toxicity of ionic liquids toward acetylcholinesterase enzymes using novel QSAR models, Int. J. Mol. Sci. 20 (2186) (2019) http://dx.doi.org/10.3390/ijms20092186.

[20] L. Cao, P. Zhu, Y. Zhao, J. Zhao, Using machine learning and quantum chemistry descriptors to predict the toxicity of ionic liquids, J. Hazard. Mater. 352 (2018) 192–202, http://dx.doi.org/10.1016/j.jhazmat.2018.03.025.

[21] Q. Yuan, Z. Wei, X. Guan, M. Jiang, S. Wang, S. Zhang, Z. Li, Toxicity prediction method based on multi-channel convolutional neural network, Molecules 24 (2019) 3383, http://dx.doi.org/10.3390/molecules24183383.

[22] Z. Wang, Z. Song, T. Zhou, Machine learning for ionic liquid toxicity prediction, Processes 9 (1) (2020) http://dx.doi.org/10.3390/pr9010065.

[23] D. Peng, F. Picchioni, Prediction of toxicity of ionic liquids based on GC-COSMO method, J. Hazard. Mater. 398 (2020) 122964, http://dx.doi.org/10.1016/j.jhazmat.2020.122964.

[24] D. Peng, J. Zhang, H. Cheng, L. Chen, Z. Qi, Computer-aided ionic liquid design for separation processes based on group contribution method and COSMO-SAC model, Chem. Eng. Sci. 159 (2017) 58–68, http://dx.doi.org/10.1016/j.ces.2016.05.027.

[25] I. Baskin, A. Epshtein, Y. Ein-Eli, Benchmarking machine learning methods for modeling physical properties of ionic liquids, J. Mol. Liq. 351 (2022) 118616, http://dx.doi.org/10.1016/j.molliq.2022.118616.

[26] D. Fan, K. Xue, R. Zhang, W. Zhu, H. Zhang, J. Qi, Z. Zhu, Y. Wang, P. Cui, Application of interpretable machine learning models to improve the prediction performance of ionic liquids toxicity, Sci. Total Environ. 908 (2024) 168168, http://dx.doi.org/10.1016/j.scitotenv.2023.168168.

[27] G.A. Tabaaza, B.N. Tackie-Otoo, D.B. Zaini, D.A. Otchere, B. Lal, Application of machine learning models to predict cytotoxicity of ionic liquids using VolSurf principal properties, Comput. Toxicol. 26 (2023) 100266, http://dx.doi.org/10.1016/j.comtox.2023.100266.

[28] S. Danush, A. Dutta, Machine learning-based framework for predicting toxicity of ionic liquids, Mater. Today: Proc. 72 (2023) 175–180, http://dx.doi.org/10.1016/j.matpr.2022.06.380.

[29] S.P. Mousavi, R. Nakhaei-Kohani, S. Atashrouz, F. Hadavimoghaddam, A. Abedi, A. Hemmati-Sarapardeh, A. Mohaddespour, Modeling of H2S solubility in ionic liquids: comparison of white-box machine learning, deep learning and ensemble learning approaches, Sci. Rep. 13 (1) (2023) 7946, http://dx.doi.org/10.1038/s41598-023-34193-w.

[30] D. Fan, K. Xue, Y. Liu, W. Zhu, Y. Chen, P. Cui, S. Sun, J. Qi, Z. Zhu, Y. Wang, Modeling the toxicity of ionic liquids based on deep learning method, Comput. Chem. Eng. 176 (2023) 108293, http://dx.doi.org/10.1016/j.compchemeng.2023.108293.

[31] I. Semenyuta, V. Kovalishyn, D. Hodyna, Y. Startseva, S. Rogalsky, L. Metelytsia, New QSTR models to evaluation of imidazolium- and pyridinium-contained ionic liquids toxicity, Comput. Toxicol. 30 (2024) 100309, http://dx.doi.org/10.1016/j.comtox.2024.100309.

[32] N.H. Abdullah, D. Zaini, B. Lal, Prediction of ionic liquids toxicity using machine learning models for application to gas hydrate, Process. Saf. Prog. 43 (S1) (2024) S199–S212, http://dx.doi.org/10.1002/prs.12599.

[33] R. Todeschini, V. Consonni, Molecular Descriptors for Chemoinformatics: Volume I: Alphabetical Listing/Volume II: Appendices, References, John Wiley & Sons, 2009, URL https://www.wiley.com/en-us/Molecular+Descriptors+for+Chemoinformatics%2C+2+Volume+Set-p-9783527628766.

[34] Z.H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012, URL https://www.crcpress.com/Ensemble-Methods-Foundations-and-Algorithms/Zhou/p/book/9781439830031.

[35] G. Chen, Z. Song, Z. Qi, K. Sundmacher, Generalizing property prediction of ionic liquids from limited labeled data: A one-stop framework empowered by transfer learning, Digit. Discov. 2 (3) (2023) 591–601, http://dx.doi.org/10.1039/d3dd00040k.

[36] P. Willett, J.M. Barnard, G.M. Downs, Chemical similarity searching, J. Chem. Inf. Comput. Sci. 38 (6) (1998) 983–996, http://dx.doi.org/10.1021/ci9800211.

[37] Kopal, Violin plots: A tool for visualizing data distributions, Anal. Vidhya (2024) Available at: https://www.analyticsvidhya.com/blog/2024/08/violin-plots/.

[38] Atlassian, A complete guide to violin plots, Atlassian (2024) Available at: https://www.atlassian.com/data/charts/violin-plot-complete-guide.

[39] J.L. Hintze, N.R. D., Violin plots: A box plot-density trace synergism, Amer. Statist. 52 (2) (1998) 181–184, http://dx.doi.org/10.1080/00031305.1998.10480559.

[40] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Comput. Sci. 28 (1) (1988) 31–36, http://dx.doi.org/10.1021/ci00057a005.

[41] E.J. Bjerrum, SMILES enumeration as data augmentation for neural network modeling of molecules, 2017, http://dx.doi.org/10.48550/arXiv.1703.07076, arXiv Preprint arXiv:1703.07076.

[42] T. Sterling, J.J. Irwin, ZINC 15–ligand discovery for everyone, J. Chem. Inf. Model. 55 (11) (2015) 2324–2337, http://dx.doi.org/10.1021/acs.jcim.5b00559.

[43] T. Sterling, J.J. Irwin, ZINC 15–ligand discovery for everyone, J. Chem. Inf. Model. 55 (11) (2015) 2324–2337.

[44] N.V. Plechkova, K.R. Seddon, Applications of ionic liquids in the chemical industry, Chem. Soc. Rev. 37 (1) (2008) 123–150.

[45] Z. Chen, R. Bertin, G. Froldi, EC50 estimation of antioxidant activity in dpphradical dot assay using several statistical programs, Food Chem. 138 (2013) 414–420.

[46] X. Jiang, A. Kopp-Schneider, Summarizing EC50 estimates from multiple dose-response experiments: A comparison of a meta-analysis strategy to a mixed-effects model approach, Biom. J. 56 (2014) 493–512.

[47] E. Perales, L. Lomba, M. Garcia-Escudero, E. Sarasa, C.E. Lafuente, B. Giner, Toxicological study of some ionic liquids, Green Process. Synth. 7 (2017) 287–295.

[48] RDKit: Open-source cheminformatics software, 2025, https://www.rdkit.org.

[49] S.A. Wildman, G.M. Crippen, Prediction of physicochemical parameters by atomic contributions, J. Chem. Inf. Comput. Sci. 39 (5) (1999) 868–873.

[50] P. Ertl, B. Rohde, P. Selzer, Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties, J. Med. Chem. 43 (20) (2000) 3714–3717.

[51] K. L., Molecular Connectivity in Chemistry and Drug Research, vol. 14, Elsevier, 2012.

[52] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422, http://dx.doi.org/10.1023/A:1012487302797.

[53] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.

[54] M. Christoph, Interpretable machine learning: A guide for making black box models explainable, 2020, Leanpub.

[55] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (2004) 199–222, http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88.

[56] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, CatBoost: unbiased boosting with categorical features, Adv. Neural Inf. Process. Syst. 31 (2018) http://dx.doi.org/10.48550/arXiv.1706.09516.

[57] G.B. Goh, C. Siegel, A. Vishnu, N.O. Hodas, N. Baker, Chemception: A deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models, 2017, http://dx.doi.org/10.48550/arXiv.1706.06689, arXiv PreprintarXiv:1706.06689.

[58] S. Kwon, H. Bae, J. Jo, S. Yoon, Comprehensive ensemble in QSAR prediction for drug discovery, BMC Bioinformatics 20 (2019) 1–12.

[59] M. Zaslavskiy, S. Jégou, E.W. Tramel, G. Wainrib, ToxicBlend: virtual screening of toxic compounds with ensemble predictors, Comput. Toxicol. 10 (2019) 81–88.

[60] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, 2016, http://dx.doi.org/10.48550/arXiv.1603.02754, arXiv preprint arXiv:1603.02754.

[61] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2) (2012) URL https://jmlr.org/papers/v13/bergstra12a.html.

[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830, http://dx.doi.org/10.48550/arXiv.1201.0490.

[63] C.R. Harris, K.J. Millman, S.J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, et al., Array programming with NumPy, Nature 585 (2020) 357–362, http://dx.doi.org/10.1038/s41586-020-2649-2.

[64] A. Jadhav, D. Pramod, K. Ramanathan, Comparison of performance of data imputation methods for numeric dataset, Appl. Artif. Intell. 33 (10) (2019) 913–933.

[65] J.L. Fleiss, B. Levin, M.C. Paik, Statistical Methods for Rates and Proportions, third ed., John Wiley & Sons, 2013.
[66] D. Berengut, Statistics for Experimenters: Design, Innovation, and Discovery, Taylor & Francis, 2006.
[67] P. Bruce, A. Bruce, P. Gedeck, Practical statistics for data scientists: 50+ essential concepts using R and Python, O'Reilly Media, 2020.
[68] J. Gareth, W. Daniela, H. Trevor, T. Robert, An Introduction to Statistical Learning: with Applications in R, Spinger, 2013.
[69] A. Sosnowska, M. Grzonkowska, T. Puzyn, Global versus local QSAR models for predicting ionic liquids toxicity against IPC-81 leukemia rat cell line: The predictive ability, J. Mol. Liq. 231 (2017) 333–340, http://dx.doi.org/10.1016/j.molliq.2017.02.025.
[70] T. Wu, W. Li, M. Chen, Y. Zhou, Q. Zhang, Estimation of ionic liquids toxicity against leukemia rat cell line IPC-81 based on the empirical-like models using intuitive and explainable fingerprint descriptors, Mol. Informatics 39 (2020) 2000102, http://dx.doi.org/10.1002/minf.202000102.