# Solvation Meta Predictor

Faiza Abdillahi

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Chemical Engineering

University of Washington

2024

Committee:

David Beck

Lilo Pozzo

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

**Abstract**

Solvation Meta Predictor

Faiza Abdillahi

Chair of the Supervisory Committee:
David Beck

Department of Chemical Engineering

Predicting the solubility of aqueous mixtures is a critical task in cheminformatics, impacting fields such as drug discovery, chemical engineering, and environmental science. This study aims to enhance the predictive accuracy of machine learning models for solubility by employing advanced ensemble techniques. We evaluated the performance of three individual models: SMI, MDM, and GNN, and compared them to ensemble methods including simple averaging and an Optuna-optimized ensemble. Our results indicate that the Optuna-optimized ensemble model achieved the highest predictive accuracy, with an $R^2$ value of 0.8117, outperforming individual models and simple ensemble techniques. To further improve model performance, we propose the implementation of the Mixture of Experts (MoE) approach. This advanced ensemble technique leverages specialized experts and a gating network to optimize model predictions based on input features. MoE promises to enhance model flexibility, scalability, and specialization, making it a robust tool for handling complex and heterogeneous datasets. Future work will involve integrating additional models and exploring other ensemble strategies to further improve predictive accuracy. The findings of this study highlight the potential of ensemble methods to significantly improve the prediction of solubility in aqueous mixtures, offering valuable insights for various scientific and industrial applications.

# Table of Contents

# Introduction:

Solubility is the property of a solid, liquid, or gaseous chemical substance called solute to dissolve in a solid, liquid, or gaseous solvent to form a homogeneous solution of the solute in the solvent. The solubility of a substance fundamentally depends on the solvent used as well as on temperature and pressure. The extent of solubility of a substance in a specific solvent is measured as the saturation concentration where adding more solute does not increase its concentration in the solution [1]. Solubility can also be measured using solvation-free energy, an essential thermodynamic concept that describes the energy change accompanying the transfer of a molecule from one phase to another. Understanding solvation-free energy is crucial, as it relates to numerous physicochemical properties, that impact solute behavior in different environments [2].

In the pharmaceutical industry, solubility plays a pivotal role in drug efficacy. The solubility of a drug determines its bioavailability, the proportion that enters the circulation when introduced into the body, and is thus able to have an active effect. Drugs with poor solubility often suffer from low bioavailability, posing significant challenges in drug formulation and delivery [3] .

Similarly, in food production, protein solubility dictates the type of food that can be produced (solid, liquid), the stabilization of different phases (oil or air), the type of processing operation that is required (thermal processing, size reduction, mixing, etc.), and the time needed to carry out these operations (swelling, hydrolysis, stirring, etc.) [4]. It also affects different quality characteristics such as appearance, sedimentation, viscosity, and flavor generation, and impacts sensory properties and nutritional aspects upon consumption [5].

# Review of State of Art:

## First Principal Calculations

Traditionally, solvation-free energy was estimated using first-principles calculations, such as Hartree-Fock (HF), Density Functional Theory (DFT), and Molecular Dynamics (MD) simulations. These methods are grounded in fundamental physical laws and provide detailed insights into molecular interactions. First-principles calculations can predict solubility limits in various alloys, essential for material design due to their impact on mechanical properties. DFT, for instance, is a quantum mechanical method that calculates material properties using electron density. HF, another quantum mechanical method, approximates the quantum state of multi-electron systems. MD simulations offer dynamic insights into molecular behavior but are often limited by computational costs over extended timescales [6].

## Cosmos-RS

COSMO-RS (Conductor-like Screening Model for Real Solvents) emerged as a significant advancement, employing statistical thermodynamics to predict the thermophysical properties of fluids. Unlike first-principles

methods, COSMO-RS relies less on computational intensity and more on the quantum chemistry of surface segments, facilitating predictions in complex mixtures without extensive empirical parameters [7].

# Machine Learning

## Transition to Machine Learning

More recently, the advent of machine learning (ML) in cheminformatics has revolutionized the prediction of solvation-free energy and solubility. Machine learning models, such as Quantitative Structure-Activity Relationship (QSAR) and Quantitative Structure-Property Relationship (QSPR), have demonstrated remarkable efficiency and accuracy. These models predict properties based on empirical or structural features of compounds, offering a novel perspective compared to traditional computational approaches [8] [9].

## Advantages of Machine Learning Over Traditional Methods

Although we may not expect to obtain detailed chemical or physical insights other than the target property because this is a regression analysis in its nature, Structure-Property Relations (SPR) have demonstrated significant potential in terms of transferability and outstanding computational efficiency [8] [10] [11]. Unlike traditional methods, ML can handle large datasets and learn complex patterns without explicitly modeling the underlying physics. This capability allows for faster calculations and more precise estimations. Several ML models have achieved accuracies comparable to ab initio solvation models, especially in aqueous systems [8] [11]. However, these models do rely heavily on the availability and diversity of training data and may face challenges in generalizing to chemically diverse systems or new compounds [11].

## Recent Developments in Machine Learning Models:

The latest advancements in machine learning, as seen in models like Delfos, MLSolvA, SolvBERT, and AquaPred, have significantly enhanced the prediction of solvation-free energy and solubility. These models employ innovative approaches like deep learning, NLP techniques, and attention-based mechanisms, setting new benchmarks in the field.

### Deflos

Delfos is an innovative deep-learning model designed for accurately predicting solvation-free energies of organic solvents. This model represents a significant advancement in computational chemistry, offering a more efficient and accurate alternative to traditional quantum mechanical (QM) models. It is particularly valuable for its ability to handle a wide range of organic solvents, which is essential in drug discovery and material science.

The model integrates a Quantitative Structure-Property Relationship (QSPR) approach with a recurrent neural network (RNN). It comprises three sub-neural networks: two encoders for the solvent and solute, which extract dominant structural features from SMILES strings, and a predictor that calculates solvation energy. The Mol2Vec embedding model is employed for encoding chemical structures, and an attention mechanism is integrated within the neural network to enhance performance. Training and testing were done using the Minnesota Solvation Database (MNSOL), a comprehensive repository with experimental measures of solvation-free energies for a

variety of solutes and solvents. The database included data for 2495 pairs of 418 solutes and 91 solvents, providing a broad scope for the model's training.

The model's performance was evaluated using Mean Absolute Error (MAE). Notably, Delfos exhibits lower MAE in non-aqueous systems compared to aqueous systems. For example, the Delfos/BiLSTM and Delfos/BiGRU models have lower MAEs for non-aqueous solutions (0.24 kcal/mol and 0.21 kcal/mol, respectively) than for aqueous solutions (0.64 kcal/mol and 0.68 kcal/mol, respectively). This differential performance suggests a higher accuracy in non-aqueous solutions, likely due to water's unique structural characteristics that pose a challenge for the model. The model faces challenges in predicting solvation energies for structurally novel compounds, emphasizing the need for diverse and extensive databases. The performance with cluster cross-validation indicates areas for improvement and adaptation for handling a wider variety of chemical structures [12].

## MLSolvA

MLSolvA introduces a novel approach to predicting solvation-free energy using machine learning (ML), focusing on pairwise atomistic interactions. The model's architecture is centered on a linear regression task to calculate solvation-free energy. It employs a unique mechanism where two encoding functions extract atomic feature vectors from a given chemical structure, and the inner product of these vectors computes their interactions. This approach allows for a more direct and interpretable understanding of intermolecular interactions. Two types of neural networks, BiLM (based on RNN) and GCN (Graph Convolutional Neural Network), are compared for encoding input molecular structures.

This model was trained and tested using 6,239 experimental measures of solvation energies for 935 organic solvents and 146 organic solutes, collected from the FreeSolv and Solv@TUM databases. A pre-training process on a vast number of organic compounds from the ZINC15 database was also implemented to enhance the model’s accuracy.

MLSolvA demonstrated excellent prediction accuracy, with mean unsigned prediction errors (MUE) of 0.19 kcal/mol for the BiLM/LSTM encoder and 0.22 kcal/mol for the GCN model. It also showed notable. The model faces challenges in extrapolation scenarios, particularly when predicting solvation-free energies for new compounds not represented in the training data. Scaffold-based split methods were used to investigate this, revealing a degradation in prediction performance [13].

## SolvBERT

SolvBERT is a pioneering model that employs natural language processing (NLP) techniques, specifically a BERT-based model, for predicting solvation-free energy and solubility. This model is significant in the field of computational chemistry for its ability to predict properties of molecular complexes, a crucial factor in various chemical and pharmaceutical applications.

The model utilizes a BERT-based regression model, which is distinct in its approach of reading the SMILES (Simplified Molecular-Input Line-Entry System) representations of solute-solvent combinations, transforming them into vectorized representations. This approach allows for a more comprehensive understanding of molecular interactions. The model was pre-trained in an unsupervised fashion using computational solvation-free energies, allowing it to predict experimental solvation-free energy or solubility depending on the fine-tuning database.

Several datasets were used for training and validation, including the CombiSolv-QM dataset with about 1 million solute-solvent pairs, the CombiSolv-Exp dataset with experimental solvent-free energy data, and a solubility dataset curated by Boobier et al.

SolvBERT showed high accuracy in predictions with notable performance in out-of-sample tests. Its capability to predict solubility and solvation-free energy was highlighted by its effective transfer learning from pre-training on computational data to fine-tuning on experimental data. The model's ability to extrapolate to new compounds do not present in the training set was a key challenge, but it still showed promising results in terms of accuracy and reliability in these scenarios [14].

### AquaPred

This study introduces an advanced machine learning model, specifically an attention-based graph neural network (GNN), for predicting the molecular solubility of compounds. Solubility prediction is a critical aspect of drug discovery and development, making this model highly significant for pharmaceutical research. It aims to enhance efficiency and accuracy in predicting aqueous solubility.

The model uses an attention-based GNN to predict molecular solubility. It involves the utilization of Simplified Molecular-Input Line-Entry System (SMILES) strings for molecular representation. The approach includes models developed using Simple Graph Convolution (SGConv), Graph Isomorphism Network (GIN), Graph Attention Network (GAT), and Attentive FP network. The Attentive FP-based network model was selected due to its superior performance, featuring a mechanism that captures intermolecular properties through information propagation and intramolecular properties by employing gated recurrent units (GRU).

The model was trained and tested on a dataset comprising 9,943 compounds. It demonstrated its proficiency on 62 anticancer compounds, showcasing robustness and versatility. It displays high accuracy with a Pearson correlation R2 value of 0.52 and a root-mean-square error (RMSE) of 0.61, indicating its efficacy in solubility prediction. These metrics underscore the model’s ability to accurately predict molecular solubility [15].

## Objective

The research presented in this paper builds upon foundational work from the study titled "Evaluation of Deep Learning Architectures for Aqueous Solubility Prediction" authored by Gihan Panapitiya, Michael Girard, Aaron Hollas, Jonathan Sepulveda, Vijayakumar Murugesan, Wei Wang, and Emily Saldanha [16]. The original study provides an insightful exploration of various deep learning architectures and their efficacy in predicting aqueous solubility, which is critical for many applications in chemical and pharmaceutical sciences. Our research aims to further this investigation by not only leveraging the machine learning models discussed but also by enhancing their predictability through sophisticated ensemble techniques.

We implement a multifaceted approach to improve solubility prediction: initially, we utilize the K-Nearest Neighbors (KNN) algorithm to select the most effective model for each instance in our test dataset. We then apply an ensemble method that involves simple averaging of model outputs. To refine the ensemble predictions, we employ Optuna for optimizing the weights of each model, ensuring that the contributions of individual models are proportionate to their predictive accuracy. This optimization is further supported by rigorous cross-validation, enhancing the validity and reliability of our findings.

To advance our methodology, we explore the Mixture of Experts (MoE) technique, which allows for a dynamic adjustment of model weights based on the specific features of each test instance. This approach promises a more tailored and potentially more accurate prediction system. Our research, rooted in the deep learning architectures detailed by Panapitiya et al., seeks to extend the current understanding and application of these models in solubility prediction, aiming to achieve superior accuracy and applicability in real-world scenarios.

# Methods:

## Data:

The dataset comprises 11,696 unique mixtures, meticulously assembled from diverse sources to ensure a robust foundation for analysis. Originally compiled by Gao et al., the dataset includes information on 11,868 molecules sourced from several reputable databases including OChem, Beilstein, and Aquasol. This dataset has been further enriched with additional data contributions from Cui et al. and a commercially available dataset from Reaxys, enhancing its comprehensiveness and utility for various scientific inquiries.

The molecular diversity within the dataset is substantial, with molecule sizes ranging from 1 to 273 atoms and molecular masses extending from 16 to 1,819 grams per mole. A significant range of aqueous solubilities is also documented, spanning from 3.4×10^-18 to 45.5 mol/L. The distributions of log solubility values, molecular mass, and atom count are illustrated in Figure 1. In this context, LogS represents the base-10 logarithm of the solubility (S) in moles per litre (mol/L), where 'L' denotes the volume of the solvent in litres.

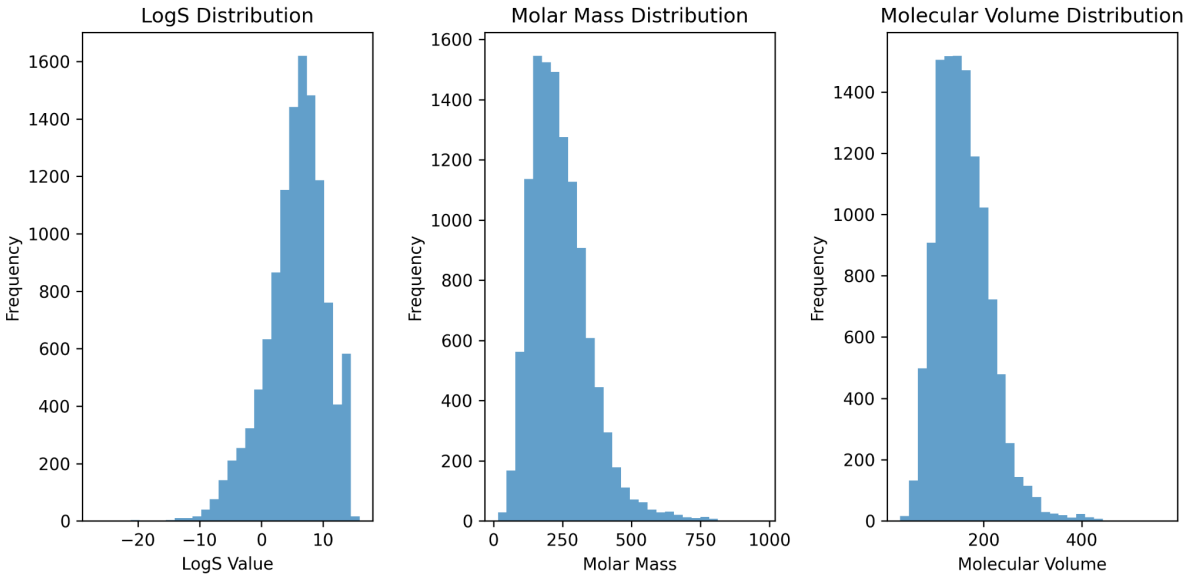


*Figure 1- **Distributions of Chemical Properties in a Dataset.** This figure presents histograms representing the distributions of three key chemical properties across a dataset: LogS, molar mass, and molecular volume. The first histogram (left) illustrates the distribution of LogS values, which show a notable frequency concentration around the values -10 and 0. The middle histogram depicts the distribution of molar mass, peaking between 250 and 500 g/mol. The third histogram (right) displays the distribution of molecular volume, with a prominent peak observed around 200 to 400 cubic Angstrom,*

# Features for each model:

## MDM Model Features

The Molecular Descriptor Model (MDM) leverages a dataset featuring 839 distinct descriptors, meticulously calculated to capture a broad spectrum of molecular properties. These descriptors encompass a blend of two-dimensional (2D) and three-dimensional (3D) molecular features, alongside functional group features and descriptors derived from Density Functional Theory (DFT)-based quantum calculations.

### 2D Descriptor:

Out of a potential 1,613 descriptors facilitated by the Mordred [17] package, 743 were successfully calculated and utilized for the model. The remainder, approximately 870 descriptors, were excluded due to generation failures across some molecules. The utilized 2D descriptors, derived from the two-dimensional molecular structure, include crucial information about atoms, bonds, and molecular connectivity without consideration of spatial configurations. Key categories and examples of these descriptors, as detailed in the provided diagrams, include:

**ABCIndex and BertzCT**: Calculations based on bond connectivity and graph theory principles.

**AtomCount and CarbonTypes**: Count specific types of atoms and bonds, respectively.

**Chi and TopologicalIndex**: Derived from graph theory, these describe the overall topology of the molecule.

**EState**: Indicators calculated from electronic states determined by molecular topology.

**Lipinski and LogS**: Reflect rules or properties such as LogP and the count of hydrogen bond donors and acceptors, pivotal in pharmacological profiling.

**Constitutional and RotatableBond**: Focus on the counts and types of specific structural elements crucial for molecular flexibility and reactivity [18].

For a full list of 2D descriptors, check Appendix A.

### 3D descriptors:

The dataset includes 37 unique three-dimensional (3D) descriptors that provide a comprehensive analysis of molecular structure by encoding information about the molecular shape. These 3D descriptors are crucial for understanding complex spatial arrangements and properties of molecules, which can be pivotal in various applications such as drug design and materials science.

**Atomic Coordinates and Optimization:** The atomic coordinates necessary for these calculations are generated using the Pybel package, a Python wrapper for the Open Babel cheminformatics toolkit. These coordinates are optimized using the Merck Molecular Force Field (MMFF94), undergoing 550 optimization steps to ensure accuracy in the resultant molecular models.

**Stratification of Atoms in Concentric Layers:** Following the optimization of atomic coordinates, the structure of the molecule is analyzed by stratifying atoms into six concentric layers around the molecule's centroid. This method, as detailed by Panapitiya et al., helps in understanding the distribution of atomic mass around the center, which can be indicative of the molecule's density and packing characteristics.

**Distribution of Atoms and Statistical Moments:** The distribution of atoms is assessed through a method proposed by Ballester and Richards. This involves calculating the distances of all atoms with respect to three strategically significant locations within the molecule: the centroid, the closest atom to the centroid, and the farthest atom from the centroid. Subsequently, the statistical moments of these atomic distance distributions are

calculated from the first to the tenth order. These moments provide insights into the symmetry, skewness, and kurtosis of the distribution, reflecting the molecule's spatial complexity and heterogeneity.

**Volume Enclosed by Atoms:** To quantify the spatial volume occupied by the molecule, the volume enclosed by all atoms is calculated using the ConvexHull function implemented in the SciPy package. This measurement is crucial for understanding the physical size of the molecule, its potential steric interactions, and its overall shape [18].

## Quantum Descriptors Derived from Density Functional Theory (DFT)

These descriptors were computed using the NWChem software package, a powerful tool for quantum chemical calculations. The DFT-based descriptors include:

**Solvation Energy (kcal/mol)**: Provides an estimate of the energy change when a molecule is transferred from the gas phase into a solvent, crucial for understanding solubility and reactivity in different environments.

**Molecular Volume (Å³)**: Measures the occupied three-dimensional space of the molecule, relevant for studying molecular packing and interaction potentials.

**Molecular Surface Area (Å²)**: Related to the exposed area of a molecule that can interact with its environment, important for binding and reactivity assessments.

**Dipole Moment (Debye)**: Quantifies the separation of positive and negative charges within a molecule, influencing molecular interactions and reactivity.

**Dipole Moment/Volume (Debye/Å³)**: Provides a normalized measure of the dipole moment, adjusting for the size of the molecule, useful in studies of molecular polarity and its effects on solubility and interactions.

**Quadrupole Moments**: Further describe the distribution of charge in a molecule, enhancing the understanding of molecular shape and its electronic environment.

These DFT-based descriptors offer deep insights into the electronic properties and behaviours of molecules under various conditions, thereby supporting more accurate simulations and predictions in chemical and pharmaceutical research.

## Additional Descriptors

In addition to quantum and structural descriptors, the dataset includes descriptors based on molecular fragments and functional groups. These features provide a qualitative aspect of molecular composition, which complements the quantitative data from other descriptors:

**Molecular Fragments**: Using RDKit, a versatile cheminformatics tool, we identified specific molecular fragments attached to benzene-like structures, which are common substructures in many organic molecules. The selection of fragments focused on those attached to a hexagonal ring with six atoms, a frequent motif in aromatic compounds.

**Commonly Found Functional Groups**: From the initial set of fragments, we further selected 52 of the most prevalent fragments across the dataset, along with seven additional functional groups that are ubiquitously found in chemical compounds. These functional groups often play critical roles in chemical reactivity and biological activity, making their identification vital for any comprehensive molecular analysis.

### GNN Model Features

The features used to train the Graph Neural Network (GNN) as listed in your document are detailed and specifically tailored to capture a wide range of atomic and molecular properties. These features are critical for effectively modeling and predicting the behavior of molecules in various scenarios, such as chemical reactions or physical interactions.

**Atomic Symbol**: Encoded as a one-hot vector representing each element in the periodic table commonly found in organic and inorganic compounds. This encoding includes a comprehensive set of elements such as Ag, Al, As, B, Br, C, Ca, Cd, Cl, Cu, F, Fe, Ge, H, I, K, Li, Mg, Mn, N, Na, O, P, Pb, Pt, S, Se, Si, Sn, Sr, Tl, Zn, and an 'Unknown' category for elements not listed.

**Degree of the Atom**: A one-hot encoded vector that quantifies the number of covalent bonds an atom has with other atoms (ranging from 0 to 10).

**Implicit Valence of the Atom**: Encoded similarly to a degree, this feature captures the typical number of chemical bonds formed by an atom.

**Formal Charge**: Represents the electrical charge of an atom, essential for understanding ionic structures and reactivity.

**Number of Radical Electrons**: Indicates unpaired electrons which are a key factor in radical reactions.

**Hybridization of the Atom**: Captured as a one-hot encoded vector indicating sp, sp2, sp3, sp3d, and sp3d2 hybridizations.

**Is the Atom Aromatic**: Boolean value Identifying whether an atom is part of an aromatic system.

**Total Number of Hydrogen Atoms**: One-hot encoded vector indicating the number of hydrogen atoms bonded to an atom, ranging from 0 to 4.

### SMI Model Features

The input feature for the SMI model is the SMILES string representation of each molecule. SMILES (Simplified Molecular Input Line Entry System) strings are a way to represent chemical structures using a line notation that encodes the structure of a molecule in a textual format. SMILES strings encode several key pieces of information about molecules such as the types of atoms and the connectivity between them, and representation of branches and ring structures in the molecular structure. SMILES can also encode information about the spatial arrangement of atoms (stereochemistry) [19].

The use of SMILES strings is a compact and efficient way to represent chemical structures in a textual format allowing for easy storage, sharing, and computational manipulation of chemical information [20].

## Base Learners

## MDM

The model is constructed using a Sequential neural network initiated with a dense layer comprising 128 neurons, integrated with a sigmoid activation function to introduce non-linearity into the model. Following the initial layer, a dropout mechanism, specified at approximately 10.7%, is implemented to mitigate the risk of overfitting by randomly omitting a subset of features during training.

The architecture progresses through additional dense layers, with neuron counts increasing to 576, these layers are accompanied by varied activation functions including ReLU and SELU, each tailored to optimize the transformation and transmission of data within the network. Dropout rates are incrementally adjusted, reaching up to 99.4% in subsequent layers, to further enhance model generalization.

The last layer of this sequential arrangement is a dense layer equipped with a linear activation function, aimed at producing a continuous output. Optimization of the model is achieved through the RMSprop algorithm, with a learning rate sourced from an external configuration, emphasizing the minimization of mean squared error.

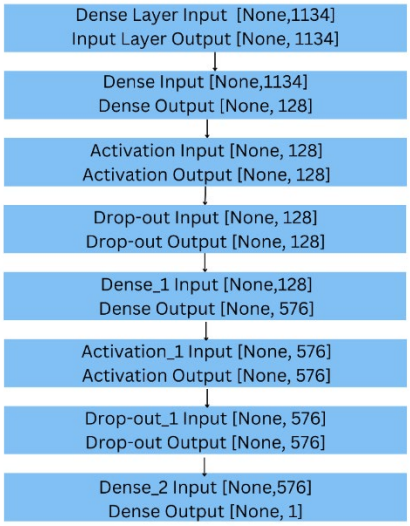


*Figure 2- **Architecture of a Fully Connected Neural Network used for MDM Model.** This figure details the structure of a deep fully connected neural network used for classification or regression tasks. The network architecture begins with an input layer that accepts a vector of size 1134. It progresses through multiple dense layers with corresponding activation and dropout layers designed to prevent overfitting and enhance model generalization. The first dense layer outputs a vector of size 128, followed by an activation and a dropout layer. Subsequent layers increase the dimensionality to 576, with similar activation and dropout stages. The final dense layer reduces the output to a single value, making the model suitable for tasks requiring a scalar output.*

## GNN

This Graph Neural Network (GNN) model is designed to handle complex graph-structured data effectively, utilizing a sophisticated architecture that combines multiple graph convolutional layers (GCN) with edge convolution (EdgeConv) and a series of fully connected layers. The model starts with several GCN layers, which facilitate the embedding of node features through localized graph convolutions, enabling the model to learn representative features where the atoms and bonds become nodes and edges of a graph, respectively. At each iteration, node features of node $i$ ($x_i$) are updated according to

$$x_i^t = \gamma^{t-1}(x_i^{t-1}, m_i^{t-1}) \qquad \textit{Equation 1}$$

For graph convolution networks (GCNs), $m_i^{t-1}$ takes the form:

$$\Sigma_{j\epsilon N(i)} \left(\frac{1}{\sqrt{deg(i)}}\right)\left(\frac{1}{\sqrt{deg(j)}}\right)\Theta \qquad \textit{Equation 2}$$

With a summation the update function [21] [22]. Thus, the node features in a GCN are updated as:

$$x_i^t = \Sigma_{j\epsilon N(i)} \left(\frac{1}{\sqrt{deg(i)}}\right)\left(\frac{1}{\sqrt{deg(j)}}\right)\Theta . x_i^{t-1} \qquad \textit{Equation 3}$$

Following the node feature transformation, an EdgeConv layer is employed to process edge attributes, which further enriches the node representations by incorporating neighboring relationships [23] for which the edge representations are updated according to,

$$x_i^t = \Sigma_{j \epsilon N(i)}\, h_\Theta(x_i^{t-1} || x_i^{t-1}, x_j^{t-1}) \qquad \textit{Equation 4}$$

*Table 1- Key Variables and Their Explanations for Graph Convolutional Network (GCN) equations 1-4.*

| Variables | Explanations |
|---|---|
| $\boldsymbol{\gamma^{t-1}}$ | Update function which is a differentiable function like a multi-layer perception |
| $\boldsymbol{m_i^{t-1}}$ | Aggregated messages coming from the neighbouring nodes |
| $\boldsymbol{N(i)}$ | The neighbouring atoms to atom i |
| $\boldsymbol{\Lambda}$ | Differentiable function that is used to aggregate the message of a given node with those of its neighbouring ones |
| $\boldsymbol{\emptyset\Theta}$ | Differentiable functions like a multi-layer perceptron |
| $\Theta$ | Weight matrix used to linearly transform the node features |
| deg(i) | Degree of the $i^{th}$ node |
| $\boldsymbol{h_\Theta}$ | An arbitrary neural network |
| $\|\|$ | Denotes concatenation of two vectors |

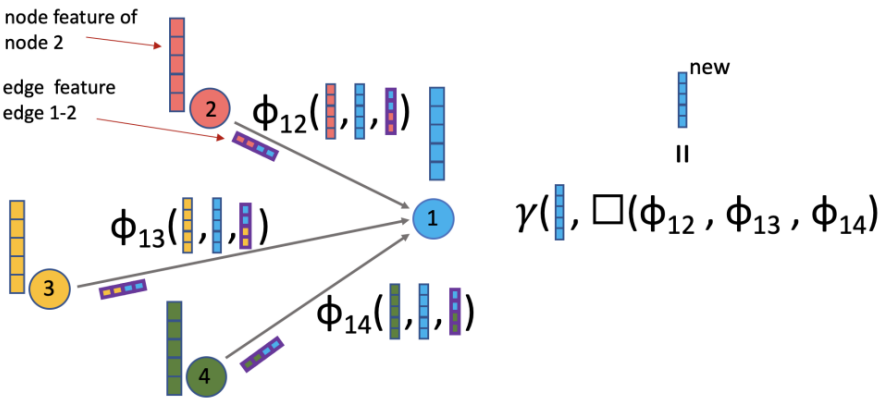


*Figure 3- **Mechanism of Message Passing and Aggregation in a Graph Neural Network**. This diagram demonstrates the core operations of message passing and aggregation within a graph neural network. Each node (labeled 1 through 4) is depicted along with its respective node features. Message functions ($\phi$) compute intermediate representations based on the attributes of the adjacent nodes and the corresponding edge features. For instance, $\phi_{12}$ denotes the message passed from node 2 to node 1, incorporating both the node features of node 2 and the edge features between nodes 1 and 2. The aggregation function ($\gamma$) then combines these messages into a new node representation for node 1, labeled $\eta_1$^new, which incorporates information from its neighborhood to update its state [18].*

To combat overfitting and ensure generalization, dropout is strategically applied after each convolutional and fully connected layer. The architecture also incorporates various activation functions (ReLU, SELU, Sigmoid), selected based on the layer and specific task, to introduce non-linear transformations and enhance learning capabilities.

Global pooling is utilized to aggregate node features across the entire graph, which is particularly beneficial for tasks requiring graph-level outputs. The model concludes with several fully connected layers that map the aggregated and processed features to the final output, demonstrating the model's capability to adapt from node-level to graph-level tasks [18].

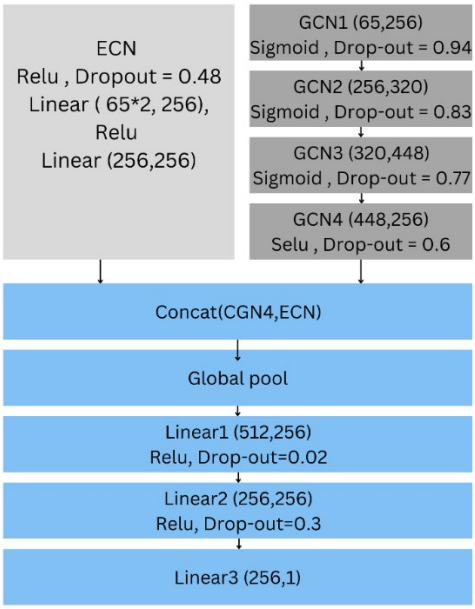


*Figure 4-* ***Detailed Architecture of a Hybrid Neural Network used for GNN Model****. This schematic illustrates a complex neural network model combining convolutional and global pooling layers with multiple densely connected layers. The model starts with an embedding convolution network (ECN), followed by four graph convolutional network layers (GCN1 to GCN4) with sigmoid activation functions and varying dropout rates to prevent overfitting. Outputs from GCN4 and the initial ECN are concatenated before being subjected to global pooling, optimizing the feature extraction process. The model concludes with three linear layers, each utilizing a ReLU activation function and different dropout rates to enhance generalization. The final output is a single neuron, suggesting the model's utility in binary classification tasks.*

## SMI

The simplest model in the ensemble is SMI mode. It is based on using the SMILES string representation of each molecule as an input to a character-level long short-term memory neural network. It is designed to process sequential data such as the character sequences that comprise SMILES strings.


*Figure 5- **Architecture of a Deep Learning Model for Sequence Processing used for SMI model**. This diagram provides a detailed view of the multi-layer neural network used for sequence data analysis. The model architecture includes an initial embedding layer that transforms input data (shape: [None, 248]) into a higher dimensional space (shape: [None, 248, 960]). It is followed by an LSTM layer and a bidirectional LSTM layer to capture temporal dependencies in both forward and reverse directions. Subsequent layers include dropout layers for regularization, and multiple dense layers with varying outputs to progressively refine the features extracted from the sequence data. The output of the final dense layer (shape: [None, 1]) represents the model's prediction.*

# Improving ML Model's Performance:

## K-Nearest Neighbours:

The k-nearest neighbours (KNN) algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point [24]. In this approach (KNN) aims to determine the best predictive model from a set of machine learning models—MDM, GNN, and SMI—based on their performance on validation data. This selection process utilizes a set of features that reflect the chemical and physical properties of molecules, such as molecular weight (MW), surface area, volume, and the number of bonds. These features are crucial for accurately assessing similarities between molecules in the dataset.

For each molecule in the test dataset, we identify KNN from the validation dataset using the Euclidean distance calculated based on the features using equation 5.

Features of point A [a1, a2, a3, .., an], features of point B [b1, b2, b3, .., bn]

$$Euclidean = \sqrt{((b1 - a1)^2 + (b2 - a2)^2 + (b3 - a3)^2 + \dots\dots + (bn - an)^2)} \quad \text{Equation 5}$$

Once the nearest neighbors are identified, we evaluate the predictions made by the MDM, GNN, and SMI models for these neighbors by comparing the predicted values of the logarithm of the solubility (LogS) against the experimental values. The absolute error is calculated for each model's predictions, and the model with the lowest average absolute error across the K-nearest neighbours is deemed the best performer for that particular test molecule.

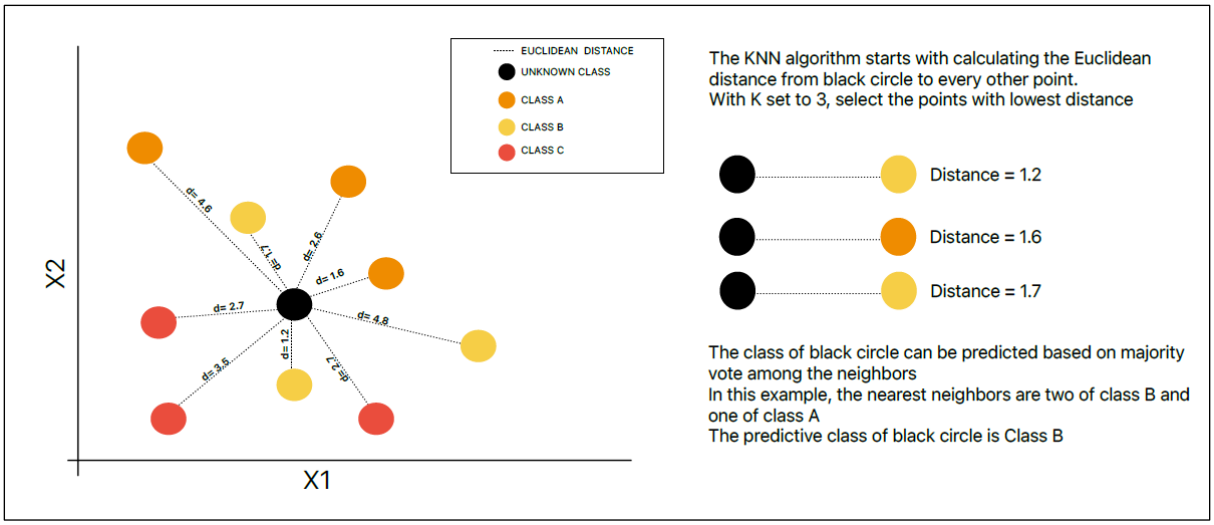


*Figure 6-**Classification of an Unknown Data Point Using the K-Nearest Neighbors (KNN) Algorithm**. This diagram illustrates the application of the KNN algorithm for classifying an unknown class (black circle) based on the Euclidean distances to labeled data points from three different classes (Class A in yellow, Class B in orange, Class C in red). With K set to 3, the algorithm identifies the three closest points (distances: 1.2, 1.6, 1.7) to the unknown point, represented by colored circles corresponding to their respective classes. The class of the unknown point is determined through a majority vote among these nearest neighbors, resulting in a classification of Class B. This example highlights the use of Euclidean distances for spatial classification and the decision-making process within the KNN framework.*

This approach tailors the model selection to the specific characteristics of each molecule, leveraging the local similarity within the dataset. By dynamically selecting the best model for each molecule based on empirical error rates within its immediate neighborhood, this method aims to enhance the overall prediction accuracy and robustness of the solution. The goal of using a KNN-based model selection is to effectively harness the strengths of different models depending on the molecular context, optimizing the performance of machine learning applications in chemical property prediction [25].

# Ensemble techniques

## Simple Averaging as an Oracle

Simple averaging is a classic example of ensemble learning, where multiple machine learning models are combined to improve the overall performance, often achieving better results than any single model could on its own.

Each of the trained models is used to predict the output variable, in this case, the logarithm of the solubility (Log S) for each molecule in the test dataset. Once predictions are obtained from all models, they are combined by calculating the arithmetic mean as in equation 6. Specifically, for each test instance, the predicted values from each model are added together and then divided by the number of models in the ensemble.

$$Average\ Ensembe = \frac{Prediction\ from\ MDM + Prediction\ from\ GNN + Prediction\ from\ SMI}{3} \quad \text{Equation 6}$$

This yields a single prediction result for each instance based on the collective intelligence of all the participating models. The performance of this simple average ensemble is then evaluated to assess how closely the predictions match the actual data and determine whether combining the models has reduced the prediction error compared to individual model outputs.

A simple average ensemble provides many advantages in comparison to independent ML models. By averaging the outputs of multiple models, the ensemble can cancel out some of the noise and errors present in individual model predictions. This often results in lower variance and more stable predictions. Ensembles typically perform better than single models as they aggregate the diverse perspectives of different models, leading to more accurate predictions. The simple average ensemble is straightforward to set up and doesn't require complex integration or additional training steps beyond the initial training of individual models.

On the other hand, simple averaging faces many limitations as it assumes that all models are equally reliable, which might not be the case if some models are significantly better or worse than others. It lacks the weighted contributions of each model by not accounting for the relative performance of each model. Models that perform better are not given more importance than those that perform poorly. Despite these limitations, simple averaging is a popular and effective ensemble method, especially when model independence can be assumed. It serves as a powerful technique to enhance prediction reliability in various practical applications, including chemical property prediction as in your use case.

## Weighted averaging improves the Oracle

### Optuna

Optuna is an automatic hyperparameter optimization software framework, designed to construct the search spaces for the hyperparameters [26]. It is an open-source library with its latest version v3.6.1 can be found on GitHub [GitHub - optuna/optuna: A hyperparameter optimization framework]. We used the latest version v3.6.1 to fine-tune the weights of the different predictive models in an ensemble setup to minimize prediction error. This process involves several key steps, each crucial for optimizing the ensemble's performance in predicting molecular solubility.

The core of the optimization process is defined in the `objective` function, where Optuna is tasked with finding the optimal combination of weights assigned to each model's predictions. Optuna initiates a trial process where it suggests different sets of weights for the models within the range of 0 to 1. These weights are then normalized to ensure that their sum equals one, maintaining a proper balance in their contribution to the final prediction.

In each trial, predictions from the GNN, SMI, and MDM models are combined using the suggested weights to produce a weighted average prediction for the validation set. The mean squared error (MSE) between these ensemble predictions and the actual values is calculated, serving as the objective metric to be minimized. Optuna seeks to find the weight configuration that results in the lowest MSE, iterating through a predefined number of trials to explore various combinations efficiently.

Upon completion of these trials, the best-performing set of weights is identified and printed out. These weights represent the most effective way to combine the individual model predictions into a single ensemble prediction that minimizes error on the validation dataset. This approach not only leverages the strengths of individual models but also mitigates their weaknesses, leading to more robust and accurate predictions. By systematically exploring the weight space, we ensure that the final ensemble model is well-tuned to the specific characteristics of the data, thus enhancing prediction accuracy and reliability in practical applications.

### Cross Validation (CV) with Optuna:

CV systematically divides the data into multiple subsets or folds each acting as a test set at different points, while the remaining data serve as the training set. This method allows the ensemble model to be trained and validated across all available data, reducing the bias that could arise from a single random train-test split [27]. Using cross-validation in the context of an ensemble model serves multiple crucial purposes, particularly in enhancing model reliability and ensuring the generalizability of the model predictions [28].

It also reduces overfitting by using each part of the data as both training and validation. This process identifies any instance where the model might perform exceptionally well on one subset of data but poorly on another, indicating overfitting. It forces the model to prove its efficacy across multiple independent data scenarios, promoting a more robust model that is less likely to overfit [28].

The repeated training and validation cycles across different data subsets provide statistical evidence of how well the ensemble model is expected to perform in practical settings. The consistency of the model's performance across these folds can be a reliable indicator of its generalizability. If the model performs well across diverse sets of data, we can be more confident in its ability to handle new, unseen data effectively.

In our ensemble method, using Optuna alongside cross-validation allows for the fine-tuning of model parameters, in this case, the weights assigned to each model's predictions. As it is crucial for finding the most effective combination of model weights that work well universally, rather than just for a specific subset of data.

# Results

These models were trained with a set of distinct chemical and physical features outlined in the prior sections of this report, which provide a comprehensive representation of the molecular characteristics relevant to solubility (LogS). Upon training, the optimized versions of each model were utilized to predict the aqueous solubility of the test data that includes 1755 data points. The efficacy of each model was quantified using several well-established performance metrics: the Coefficient of Determination ($R^2$), Spearman's rank correlation coefficient, Root Mean Square Error (RMSE), and Mean Absolute Error (MAE), chosen to provide a holistic assessment of each model's accuracy and predictive consistency.

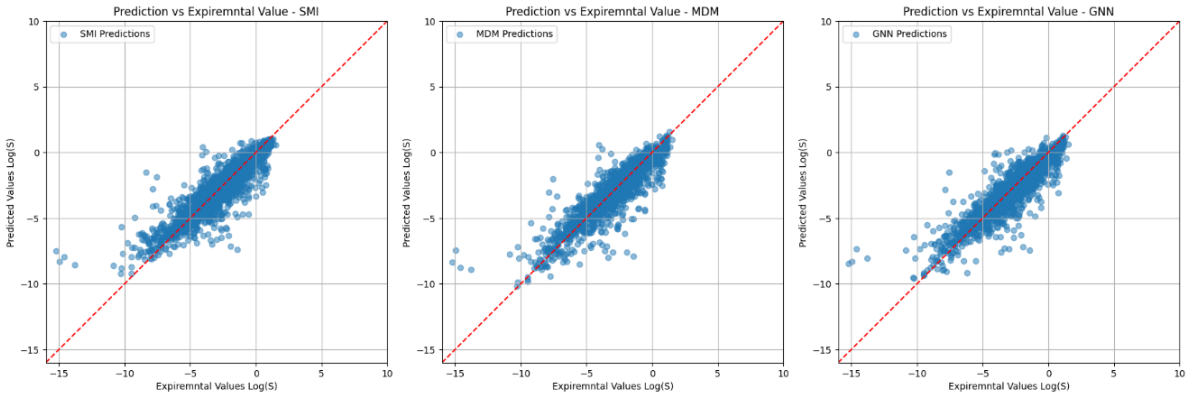


*Figure 7-**Experimental vs. Predicted Values of LogS**. Scatter plots comparing the predicted LogS values against the experimental LogS values for three different models: (left) Simplified Molecular-Input Line-Entry System (SMI), (center) Molecular Descriptor Model (MDM), and (right) Graph Neural Network (GNN). Each point represents a prediction for a solute, with the red dashed line indicating perfect predictions (where predicted values equal experimental values). The alignment of the points along the red dashed line indicates the accuracy of each model's predictions.*

The performance results, as shown in Table 2, reveal that the MDM model outperformed its counterparts across the board. Specifically, it demonstrated the highest $R^2$ value of 0.7973, indicating its superior ability to explain the variance within the dataset. Additionally, it achieved the best Spearman correlation coefficient of 0.8983, suggesting a strong monotonic relationship between the predicted and actual LogS values. In terms of error metrics, the MDM model also reported the lowest values with an RMSE of 1.0140 and an MAE of 0.6699, underscoring its predictive precision and reliability.

*Table 2- Performance metrics of individual ML models, Optuna ensemble, and Cross-Validation with Optuna ensemble*

| **Model\Metric** | **$R^2$** | **Spearman** | **RMSE** | **MAE** |
|---|---|---|---|---|
| SMI | 0.7726 | 0.8837 | 1.0741 | 0.7255 |
| MDM | 0.7973 | 0.8983 | 1.0140 | 0.6699 |
| GNN | 0.7941 | 0.8953 | 1.0219 | 0.6854 |
| K-NN | 0.7932 | 0.8958 | 1.0241 | 0.6834 |
| Simple averaging | 0.8105 | 0.9052 | 0.9804 | 0.6446 |
| Ensemble using Optuna | 0.8117 | 0.9058 | 0.9773 | 0.6413 |
| Ensemble using Optuna and Cross-Validation | 0.8116 | 0.9058 | 0.9775 | 0.6406 |

GNN model also performed well, with an $R^2$ value of 0.7941 and a Spearman correlation of 0.8953, as shown in Table 2. Its RMSE and MAE were slightly higher than those of the MDM model, at 1.0219 and 0.6854, respectively. The SMI model, while still performing strongly, showed slightly lower metrics across the board, with an $R^2$ of 0.7726, a Spearman correlation of 0.8837, an RMSE of 1.0741, and an MAE of 0.7255.

K Nearest Neighbors (KNN) algorithm was used to improve the prediction of individual models by choosing the best-performing model for each test data point. KNN approach yielded an $R^2$ value of 0.7932 and a Spearman correlation of 0.8958, with an RMSE of 1.0241 and an MAE of 0.6834, as shown in Table 2. These results indicate that K-NN performed comparably to the GNN model but did not surpass the performance of the MDM model.

Simple averaging of the predictions from the individual models resulted in a notable improvement in performance metrics. $R^2$ and the Spearman correlation values increased to 0.8105 and 0.9052 respectively, as shown in Table 2. Indicating a higher level of explained variance and a stronger monotonic relationship compared to the individual models and KNN approach. Furthermore, the RMSE and MAE values dropped to 0.9804 and 0.6446, respectively, demonstrating enhanced predictive accuracy and reduced error margins. This suggests that simple averaging is an effective method for leveraging the strengths of multiple models to achieve better overall performance.

Further efforts to optimize model performance involved the use of Optuna for hyperparameter tuning, both with and without cross-validation. The ensemble model using Optuna alone achieved an $R^2$ value of 0.8117 and a Spearman correlation of 0.9058. The RMSE and MAE values were 0.9773 and 0.6413, respectively, indicating marginal improvements over simple averaging.

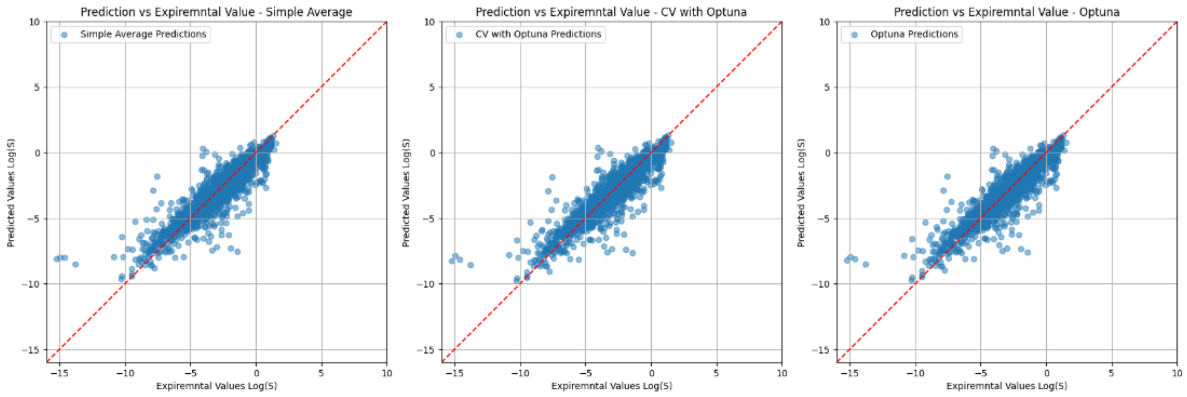


*Figure 8-**Ensemble techniques Experimental vs. Predicted Values of LogS.** Scatter plots comparing the predicted LogS values against the experimental LogS values for three different models: (left) Simple Averaging, (center) Cross-Validation and Optuna, and (right) Optuna. Each point represents a prediction for a solute, with the red dashed line indicating perfect predictions (where predicted values equal experimental values). The alignment of the points along the red dashed line indicates the accuracy of each model's predictions.*

When combining Optuna with cross-validation, the ensemble model exhibited similar performance, with an $R^2$ of 0.8116 and a Spearman correlation of 0.9058. The RMSE and MAE were 0.9775 and 0.6406, respectively, as shown in Table 2. These results suggest that while the inclusion of cross-validation did not significantly enhance the metrics compared to Optuna alone, the ensemble models tuned with Optuna consistently outperformed the individual models, KNN, and simple averaging in terms of predictive accuracy and error reduction.

Overall, the application of ensemble techniques and hyperparameter optimization through Optuna demonstrated the most effective improvements in model performance, as evidenced by the highest $R^2$ values and the lowest error metrics among all evaluated methods.

# Discussion

This research aims to explore techniques to improve the performance of existing machine learning models for predicting the solubility of aqueous mixtures through variance ensemble techniques. The major finding is that the weighted ensemble model, optimized using Optuna, outperformed the individual models it comprises. Specifically, the ensemble model using Optuna, and cross-validation achieved the highest $R^2$ value of 0.8117 and the lowest RMSE and MAE values of 0.9773 and 0.6413, respectively. These results indicate that the ensemble approach provides a significant improvement in predictive accuracy and reliability compared to the individual models.

The findings demonstrate the effectiveness of ensemble learning in improving the predictive performance of solubility models. By combining the strengths of multiple models, the ensemble approach reduces the variance and bias inherent in individual models, leading to more accurate and robust predictions. The use of Optuna for hyperparameter optimization further enhances the performance by fine-tuning the weights assigned to each model in the ensemble. This is important because accurate solubility predictions are crucial for various applications in drug discovery, chemical engineering, and environmental science, where solubility plays a key role in the behavior and efficacy of compounds.

The performance of the ensemble model in this study is consistent with findings from similar studies in the literature. For instance, AqSolPred developed by Murat Cihan Sorkun, J.M. Vianney A. Koelman, Süleyman Er 2021, achieved an $R^2$ of 0.94 and an RMSE of 0.483 [29], as shown in Table 3, highlighting the potential of advanced machine learning models for solubility prediction. The results of this study align with these findings, further confirming that ensemble techniques can harness the strengths of various models to achieve superior performance.

*Table 3- Comparison between ensemble with Optuna and state of art ML models*

| **Model\Metric** | **$R^2$** | **RMSE** | **Data set size** |
|---|---|---|---|
| Ensemble using Optuna | 0.8117 | 0.9773 | 11,696 |
| SCHNET [16] | 0.6946 | 1.2429 | 10,000 |
| UG-RNN-CR+LogP [30] | 0.81 | 0.72 | 74 |
| UGR-NN+LogP [30] | 0.91 | 0.61 | 1026 |
| AqSolPred [29] | 0.94 | 0.483 | 1290 |

However, it is notable that models like UGR-NN+LogP and AqSolPred, which performed exceptionally well with R2 values of 0.91 and 0.94 respectively, were trained on much smaller datasets (1,026 compounds for UGR-NN+LogP and 1,290 compounds for AqSolPred). This raises the possibility that these models may be memorizing the data rather than generalizing well to unseen data. In contrast, the ensemble model in this study, trained on a larger dataset of 11,696 compounds, aims to generalize better, even though its performance metrics are slightly lower. This underscores the importance of dataset size and diversity in developing robust predictive models.

While the weighted ensemble model demonstrated superior performance, alternative explanations for these findings could include the inherent diversity of the individual models used in the ensemble. The combination of models such as SMI, MDM, and GNN, which capture different aspects of the data, may inherently lead to improved performance. Additionally, the specific dataset used in this study, which contains 11,696 compounds, may also influence the results, as the diversity and size of the dataset can impact the generalizability of the models.

Despite the promising results, this study has several limitations. First, the dataset used, while large, may not encompass the full range of chemical diversity found in real-world applications. Second, the models and ensemble techniques tested may not capture all relevant factors influencing solubility, such as specific interactions between solutes and solvents. Third, the computational complexity and resource requirements of ensemble methods, especially those involving advanced optimization techniques like Optuna, may limit their practical applicability in some scenarios.

Future research should explore the inclusion of additional models to the ensemble, such as more sophisticated neural network architectures and other state-of-the-art techniques like transfer learning. Additionally, other ensemble techniques, such as Mixture of Experts or stacking, should be investigated to further improve predictive performance. Finally, expanding the dataset to include a more diverse range of compounds and testing the models

on external validation sets will be crucial for assessing the generalizability and robustness of the proposed methods.

# Future Work and Recommendation:

This research aimed to enhance the predictive performance of machine learning models for the solubility of aqueous mixtures through the application of variance ensemble techniques. The primary objective was to determine whether combining multiple models could yield more accurate and reliable predictions than individual models alone. The findings of this study indicate that ensemble models, particularly those optimized using advanced methods like Optuna, outperform individual models. The weighted ensemble model demonstrated superior performance metrics, such as a higher $R^2$ and lower RMSE and MAE values, compared to the standalone models of SMI, MDM, and GNN. These results suggest that ensemble learning is a powerful approach for improving model accuracy in predicting solubility, aligning with similar trends observed in past research.

The significance of these findings lies in their potential to influence the field of cheminformatics, particularly in applications such as drug discovery, chemical engineering, and environmental science. Accurate solubility predictions are crucial for understanding the behavior of compounds, and the demonstrated effectiveness of ensemble techniques can lead to more reliable and accurate predictions. This study underscores the importance of leveraging multiple models and advanced optimization methods to achieve superior predictive performance.
Future research should explore the introduction of new machine learning models and the application of innovative ensemble techniques such as AqSolPred and UGR-NN+LogP, those models despite being trained on smaller datasets, achieved high-performance metrics, highlighting the potential to further enhance predictive accuracy. Additionally, experimenting with other ensemble strategies, such as Mixture of Experts or stacking, could provide new insights and improvements. Expanding the dataset to include a more diverse range of compounds and validating the models on external datasets will also be essential to ensure the robustness and generalizability of the findings.

## Adding ML models

The current ensemble performance is hindered by common outliers among the models that do not have accurate predictions from any of the models. This limitation is illustrated in Figure Y, which shows the predicted solubility LogS values against the experimental solubility LogS values for the common outliers. The absence of accurate predictions for these outliers highlights the need for additional models that can learn from the unique characteristics of these molecules and provide more accurate predictions. Therefore, to enhance the performance of the ensemble, it is essential to incorporate new models capable of better handling these challenging cases.

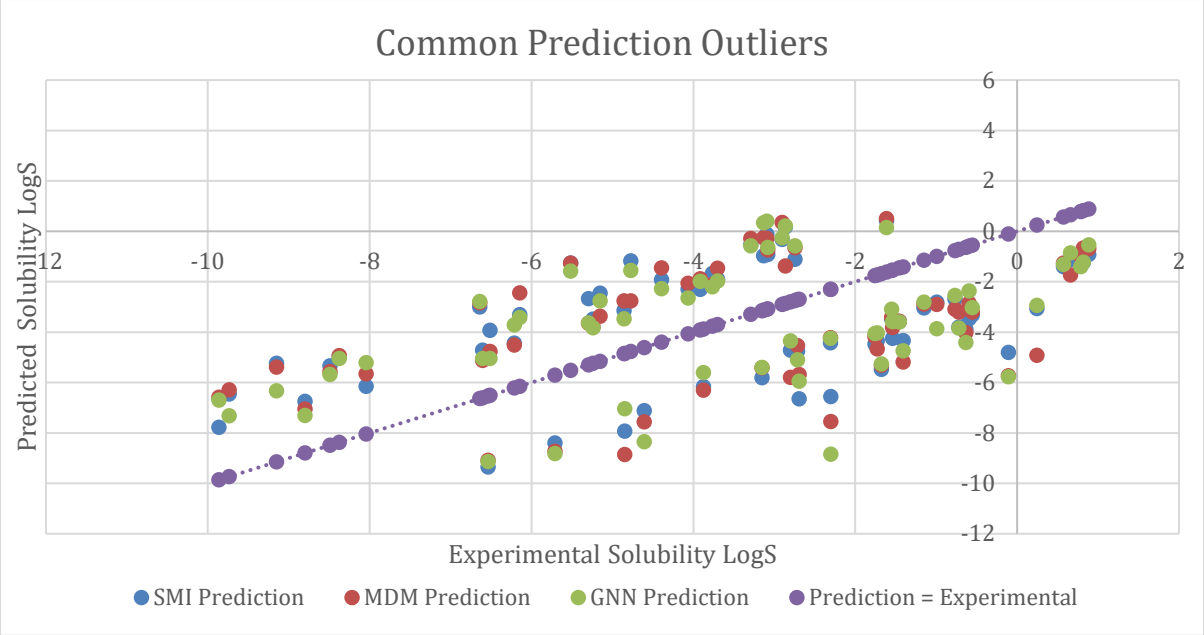


*Figure 9-**Common Prediction Outliers**. This scatter plot compares the predicted solubility LogS values against the experimental solubility LogS values for the SMI, MDM, and GNN models. The purple dotted line represents the ideal case where predicted values equal experimental values. The clustering of points away from the line indicates the inaccuracy of predictions for certain outliers.*

# Mixture of Experts (MoE)

The Mixture of Experts (MoE) approach is an advanced ensemble technique that can potentially enhance the performance of an ensemble model by optimally combining the predictions of multiple machine learning models. This technique divides the problem space into regions and assigns a specialized expert to handle each region, effectively leveraging the strengths of different models for different types of input data.

The critical component of MoE is the gating network, which is essentially a model itself, often implemented as a soft classifier. It decides the weighting of each expert's output based on the input features. The gating network learns to assign higher weights to the most competent experts for a given input instance during the training phase. Training an MoE system involves simultaneously optimizing the expert models and the gating network. The objective is to minimize the overall prediction error, where each expert's prediction is weighted by the gating network's output. This can be challenging as it requires balancing the training of multiple models and the gating mechanism.

MoE provides many advantages over traditional ensemble techniques, experts can specialize in different regions of the input space, making MoE particularly effective for heterogeneous datasets with diverse patterns and relationships. MoE can integrate various types of models as experts, making it versatile and capable of handling complex, multi-faceted problems. New experts can be added as needed for new types of data or to improve performance in specific areas, allowing the model to scale in complexity with the problem.

In conclusion, this research has demonstrated the substantial benefits of ensemble techniques in improving the predictive performance of solubility models. The results provide a strong foundation for further exploration and innovation in this area, with the potential to significantly impact various scientific and industrial applications.

# References


[1] Leon Lachman , Herbert A. Lieberman, Joseph L. Kanig, The Theory And Practise of Industrial Pharmacy, 3rd edition, Lea & Febiger, 1986.

[2] Dongdong Zhang, Song Xia, Yingkai Zhang, “Accurate Prediction of Aqueous Free Solvation Energies Using 3D Atomic Feature-Based Graph Neural Network with Transfer Learning,” *Journal Of Chemical Informattion And Modeling,* vol. 62, no. 8, pp. 1840-1848, 2022.

[3] Y. SRK, “Pharmaceutical technologies for enhancing oral bioavailability of poorly soluble drugs,” *Journal Of Bioequivalence And Bioavailability,* vol. 2, no. 2, pp. 28-36, 2010.

[4] Thierry Hellebois, Claire Gaiani, Sébastien Planchon, Jenny Renaut, Christos Soukoulis, “Impact Of Heat Treatment On The Acid Induced Gelation Of Brewers’ Spent Grain Protein Isolate,” *Food Hydrocolloids,* vol. 113, 2020.

[5] Audrey Cosson, Lydie Oliveira Correia, Nicolas Descamps, Anne Saint-Eve, Isabelle Souchon, “Identification And Characterization Of The Main Peptides In Pea Protein Isolates Using Ultra High-Performance Liquid Chromatography Coupled With Mass Spectrometry And Bioinformatics Tools,” *Food Chemistry,* vol. 367, 2022.

[6] Takafumi Mochizuki, Tokuteru Uesugi, Yorinobu Takigawa, “Prediction System for Solid Solubility Limits of Ag-, Cu-, Al-, and Mg-Based Alloys Using Artificial Neural Networks and First-Principles Calculations,” *Material Transactions,* vol. 61, no. 11, 2020.

[7] K. Paduszyński, “An Overview Of The Performance Of The COSMO-RS Approach In Predicting The Activity Coefficients Of Molecular Solutes In Ionic Liquids And Derived Properties At Infinite Dilution,” *PubMed,* vol. 19, no. 19, pp. 11835-11850, 2017.

[8] Artem Cherkasov, Eugene N Muratov, Denis Fourches, Alexandre Varnek, Igor I Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C Martin, Roberto Todeschini, Viviana Consonni, Victor E Kuz'min, Richard Cramer, Romualdo Benigni, Chihae Yang, James, “QSAR Modeling: Where Have You Been? Where Are You Going To?,” *PubMed,* vol. 57, no. 12, pp. 4977-5010, 2014.

[9] J. B. O. Mitchell, “Machine Learning Methods In Chemoinformatics,” *Wiley Interdiscip Rev Comput Mol Sci,* vol. 4, no. 5, pp. 468-481, 2017.

[10] T. A, “Best Practices For QSAR Model Development Validation And Exploitation,” *Molecular Informatics,* vol. 29, no. 6-7, pp. 476-488, 2010.

[11] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, Vijay Pande, “MoleculeNet: A Benchmark for Molecular Machine Learning,” *Chemical Science ,* vol. 9, no. 2, pp. 513-530, 2017.

[12] Hyuntae Lim, YounJoon Jung , “Delfos: Deep Learning Model For Prediction Of Solvation Free Energies In Generic Organic Solvents,” *Chemical Science,* vol. 10, pp. 8306-8315, 2019.

[13] Hyuntae Lim, YounJoon Jung, “MLSolvA: solvation free energy prediction from pairwise atomistic interactions by machine learning,” *Journal of Cheminformatics,* 2021.

[14] Jiahui Yu, Chengwei Zhang, Yingying Cheng, Yun-Fang Yang,Yuan-Bin She, Fengfan Liu, Weike Su,An Su , “SolvBERT for solvation free energy and solubility prediction: a demonstration of an NLP model for predicting the properties of molecular complexes,” *Royal Society of Chemistry,* no. 2, 2023.

[15] Waqar Ahmad, Hilal Tayara, Kil To Chong, “Attention-Based Graph Neural Network for Molecular Solubility Prediction,” *ACS Omega,* vol. 8, pp. 3236-3244, 2023.

[16] Gihan Panapitiya, Michael Girard, Aaron Hollas, Jonathan Sepulveda, Vijayakumar Murugesan, Wei Wang,Emily Saldanha, “Evaluation of Deep Learning Architectures for Aqueous Solubility Prediction,” vol. 7, no. 18, pp. 15695-15710, 2022.

[17] Moriwaki, H.; Tian, Y.-S.; Kawashita, N.; Takagi, T., “Mordred: a molecular descriptor calculator,” *Journal of Cheminformatics,* vol. 10, no. 4, 2018.

[18] Gihan Panapitiya, Michael Girard, Aaron Hollas, Jonathan Sepulveda, Vijayakumar Murugesan, Wei Wang, Emily Saldanha, “Evaluation of Deep Learning Architectures for Aqueous Solubility Prediction,” *Published by American Chemical Society,* p. 15695–15710, 2022.

[19] D. Weininger, “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules,” *Journal of Chemical Information and Computer Sciences,* vol. 28, no. 1, pp. 31-36, 1988.

[20] “Simplified molecular input line entry specification,” Chem Europe, [Online]. Available: https://www.chemeurope.com/en/encyclopedia/Simplified_molecular_input_line_entry_specification.html. [Accessed 16 05 2024].

[21] Thomas N. Kipf,Max Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in *ICLR 2017*, 2017.

[22] Matthias Fey, Jan Eric Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” in *ICLR 2019 (RLGM Workshop)*, 2019.

[23] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, Justin M. Solomon, “Dynamic Graph CNN for Learning on Point Clouds,” in *CoRR*, 2018.

[24] “What is the KNN algorithm?,” IBM, [Online]. Available: https://www.ibm.com/topics/knn. [Accessed 15 05 2024].

[25] K. Taunk, S. De, S. Verma and A. Swetapadma, , “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification,” in *International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019.

[26] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, Masanori Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” 25 7 2019. [Online]. Available: https://arxiv.org/pdf/1907.10902. [Accessed 15 05 2024].

[27] J. Brownlee, “A Gentle Introduction to k-fold Cross-Validation,” Machine Learning Mastery, 04 10 2023. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/. [Accessed 18 05 2024].

[28] Stephen Bates, Trevor Hastie, Robert Tibshiranic, “Cross-Validation: What Does It Estimate and How Well Does It Do It?,” *Journal of the American Statistical Association,* pp. 1-12, 2023.

[29] Murat Cihan Sorkun, J.M. Vianney A. Koelman, Süleyman Er, “Pushing the limits of solubility prediction via quality-oriented data selection,” *IScience,* vol. 24, no. 1, p. 101961, 2021.

[30] Alessandro Lusci, Gianluca Pollastri, Pierre Baldi, “Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules,” *Journal of Chemical Information and Modeling,* vol. 53, no. 7, pp. 1563-1575, 02 07 2013.

# Appendix A: Full list of 2D descriptors calculated from Morder library

*Table 4-2D descriptors calculated from Morder library*

| Descriptor category | Descriptor category |
|---|---|
| **ABCIndex** | ABC, ABCGG |

| | |
|---|---|
| **AcidBas** | nBase, nAcid |
| **Aromatic** | nAromAtom, nAromBond |
| **AtomCount** | nI, nX, nHeavyAtom, nCl, nBr, nS, nSpiro, nH, nF, nHetero nC, nP, nB, nBridgehead, nO, nN, nAtom |
| **Autocorrelation** | ATSC3i, ATS5are, ATS0are, ATS8are, ATS3dv, ATSC1Z, ATSC5m, ATSC6dv, ATS3are, ATS7i, AATS0dv, ATSC6m, ATS6d, AATSC0pe, ATS7pe, AATS0pe, ATSC7m, ATSC0d,ATS3v, ATSC6i, ATSC5i, AATSC0v, ATS4m, ATSC1m,ATS7Z, ATS8dv, ATSC4are, ATSC4p, ATSC5d, ATSC4m,ATSC7i, ATSC5are, ATSC6v, ATS2i, ATS5pe,ATSC5dv, ATSC1are, AATS0p, ATS0dv, ATS5v, ATSC8are, ATS6are, ATSC8v, ATSC2p, ATSC0p, ATS1m, ATS0d, ATSC6Z, ATS5Z, ATS4Z, ATS2Z, ATSC3v,ATS2pe, ATS4p, ATS1v', 'ATSC8dv', 'ATS4pe', 'ATSC4i', 'ATS0m', 'ATSC2m', ATS0i, ATS1are, AATS0m, ATSC6d, ATSC8pe, ATS8m, ATSC3p, ATS5m, ATS5d, ATS5p, ATSC8d, ATSC0i, ATS1pe, ATSC6pe, ATS0pe, ATSC6are, ATSC0pe, ATSC5p,ATS7m, ATS6dv, ATS1i, AATSC0Z, ATS8i, AATSC0p, ATSC0Z, ATSC7pe, ATSC7v, ATSC1p, ATS1p, ATS1Z, ATS3Z, ATS6i, ATSC0dv, ATSC6p, ATS8p, ATSC8p, ATS4i, ATSC0are, AATS0Z, ATS3d, AATS0d, ATS2m, ATS5dv, ATS7p, ATSC2d, ATS6v, ATS1d,ATSC4pe, ATS3pe, ATS8Z, ATS1dv, ATS8d, AATSC0d, ATSC7dv, AATSC0m, ATSC0v, ATSC7d, ATSC8Z, ATSC1i, ATSC1d,ATSC5pe, ATSC1dv, ATSC2dv, ATS7dv, ATSC4dv, ATS7v, ATSC3pe, ATSC3Z, ATS3m, ATS2d, ATSC1pe, ATS0p, ATS4d, ATS8v, ATS2p, AATSC0are, AATSC0i, AATS0are, ATSC8m, ATS3p, ATS5i, ATSC4d, ATSC2v, ATS3i, ATSC2pe, ATS7are, AATS0i, ATSC2Z, ATSC4Z, ATSC3are, AATSC0dv, ATS8pe, ATS0v, ATS2dv, ATSC7Z, ATS4are, ATSC5v, ATSC7are, ATS0Z, ATS7d, ATSC1v, ATS4dv, ATSC3dv,ATS4v, ATS6pe, ATSC0m, ATSC2i, ATS6m, ATS2v, ATS2are, ATSC4v, ATS6Z, ATSC7p, ATSC3m, ATSC2are, ATSC5Z, ATSC8i, ATS6p, ATSC3d, AATS0v |

*Table 5- continue. 2D descriptors calculated from Morder library*

| **Descriptor category** | **Descriptor category** |
|---|---|
| **BalabanJ** | BalabanJ |
| **BertzCT** | BertzCT |
| **BondCount** | nBondsS, nBondsO, nBonds, nBondsA, nBondsT, nBondsKD, nBondsM, nBondsKS, nBondsD |
| **CarbonTypes** | C1SP2, C1SP1, C3SP2, FCSP3, C3SP3, C2SP1, C1SP3, C2SP3, C4SP3, C2SP2 |

| | |
|---|---|
| **Chi** | Xp-5dv, Xc-5dv, Xpc-4dv, Xp-4dv, Xp-3d, Xch-3dv, Xpc-5dv, Xp-7d, Xc-3dv, Xp-6d, Xch-3d, Xp-6dv, Xpc-6d, Xch-4dv, Xch-5d, Xc-4d, Xp-2d, Xc-5d, Xch-7dv, Xc-3d, Xch-6dv, Xp-2dv, Xch-4d, Xch-6d, Xp-3dv, Xpc-4d, Xp-1dv,Xch-5dv, Xc-4dv, Xp-4d,Xpc-6dv, Xp-5d, Xp-7dv, Xch-7d, Xpc-5d, Xc-6dv, Xp-1d, Xc-6d |
| **Constitutional** | Spe, Sare, Si, Mpe, Mm, Sv, Mv, SZ, MZ, Mare, Sm, Mi, Sp, Mp |
| **EStat** | NssS, NssssSn, NsssPbH, SssssSi, NsGeH3, SaaNH, NsNH3, SddsN, NsSeH, StN, SssSe, SsNH2, NaasC, NdsCH, SsCl, SdCH2, SaaS, SsssdAs, NdSe, NssPH, SdSe, StsC, NaaN, NdssC, SsssCH, SssBe, NaaCH, NaaO, SsCH3, SssS, SdsCH, SssPH,SsssPbH, NsCH3, SaaCH, NsF, NsssssAs, SssBH, NddC, NdsssP, NssGeH2, SdssC, SdsssP, NssPbH2, SssssB, NsBr, NaaNH, SaaaC, SsssSnH, NssssBe, NssNH2, SssssssP, SssssSn, NsssB, NsI, NdS, SaasC, NssBe, SsSnH3, NdssSe, NddsN, NdssS, NtN, SssCH2, NtsC, SsssNH, NsssSnH, SssssGe, NssSnH2, NsssssP, NddssSe, NsSiH3, NddssS, SaasN, SssssssAs, SddssS, NaaSe, SssssBe, NssSiH2, NssAsH, NsPH2, SaaN, SsssN, SssO, SssssC, NaaaC, NsssNH, NsSnH3, SsPH2, SaaSe, SddC, NssssGe, SssssB, SssssN, SssssP, SsssGeH, SdsN, SddssSe, NdCH2, SsSeH, NsssGeH, SsGeH3, SdssSe, NssNH, SssAsH, SsLi, NdNH, SssGeH2, NsOH, SdO, NdO, NtCH, NssssB, NssSe, SsssAs, SssPbH2, NssssN, NsCl, SdS, SsSH, SsOH, NsssSiH, StCH, SsI, SssNH, SsSiH3,SsAsH2, NssssPb, SdNH, NdsN, NssBH, SsPbH3, SsssSiH, SaaO, NaaS, SsF, NsAsH2, NsssdAs, NsSH, NssssSi, SssSnH2, SssSiH2, NsPbH3,NsssP,NssCH2, NaasN, SsBr, NsssCH, SsNH3, NsNH2, NssO, SssssPb, NssssC, SdssS, NsssN, NsLi, SssNH2, NsssAs |
| **EccentricConnectivityIndex** | ECIndex |
| **FragmentComplexity** | fragCpx |
| **Framework** | fMF |
| **HydrogenBon** | nHBDon, nHBAcc |

*Table 6- continue. 2D descriptors calculated from Morder library*

| Descriptor category | Descriptor category |
|---|---|
| **InformationContent** | CIC5, TIC2, ZMIC3, IC5, ZMIC1, MIC5, ZMIC5, IC0, TIC3, MIC3, ZMIC2, TIC5, MIC0, CIC1,CIC0, MIC4, ZMIC0,TIC0, IC2, IC3,TIC1, MIC2, IC4, CIC4, TIC4, CIC3, ZMIC4, MIC1, IC1, CIC2 |
| **Lipinski** | Lipinski, GhoseFilter |
| **LogS** | FilterItLogS |
| **McGowanVolu** | VMcGowan |

| | |
|---|---|
| **MoeTyp** | VSA_EState2, SlogP_VSA11, SlogP_VSA9, SMR_VSA6, EState_VSA6, PEOE_VSA13, SlogP_VSA6, PEOE_VSA9, VSA_EState7, SMR_VSA7, SlogP_VSA2, VSA_EState6, SlogP_VSA8, PEOE_VSA6, EState_VSA9, EState_VSA1, VSA_EState8, EState_VSA3, SlogP_VSA5, EState_VSA4, VSA_EState9, VSA_EState1, PEOE_VSA1, SMR_VSA5, SMR_VSA8, SlogP_VSA10, PEOE_VSA7, VSA_EState4, SlogP_VSA7, VSA_EState5, EState_VSA8, SMR_VSA2,PEOE_VSA10, PEOE_VSA3, LabuteASA, PEOE_VSA8, SMR_VSA3, EState_VSA5, EState_VSA10, SlogP_VSA1, PEOE_VSA2, SMR_VSA4, PEOE_VSA11, VSA_EState3, PEOE_VSA4, SMR_VSA1, PEOE_VSA5, EState_VSA7, SlogP_VSA4, SlogP_VSA3, PEOE_VSA12, SMR_VSA9, EState_VSA2 |
| **PathCount** | piPC1, TMPC10, piPC10, piPC3, MPC10, MPC9, MPC8, MPC5, piPC6, piPC5, piPC8, piPC9, piPC7, MPC2, piPC2, MPC7, TpiPC10, piPC4, MPC6, MPC4, MPC3 |
| **Polarizability** | bpol, apol |
| **RingCount** | n9FAHRing, n5aRing, nG12FHRing, n5HRing, n11AHRing, n5AHRing, n3HRing, n12AHRing, n9ARing, n6FHRing, nARing, n4HRing, n10aRing, n9Ring, n5FaHRing, nRing, naRing, n6FaRing, n10HRing, n6FaHRing, n6ARing, nFaRing, n9HRing, n6aRing, n5FARing, n8FARing, n4FaRing, nG12Ring, n4ARing, n11FRing, n4AHRing, n12aRing, n12FRing, n9aRing, n12FARing, n5FaRing, n3AHRing, naHRing, n9FaHRing, n5FRing, n6aHRing, n12FaRing, n11FHRing, n8FHRing, n7FaRing, n3aRing, n7FARing, n8FaHRing, n4FARing, n6FAHRing, n5FHRing, n12FaHRing, n3aHRing, n9aHRing, nFARing, nFaHRing, n6Ring, n11FaRing, nG12FaHRing, n9FARing, nFAHRing, n12FHRing, n12HRing, n12Ring, n10aHRing, n4aRing, nG12FARing, n3Ring, n10FaHRing, n8FRing, n9FRing, nAHRing, n10AHRing, n9FHRing, n7AHRing, n8aHRing, nFHRing, n12ARing, n12aHRing, n9AHRing, nG12FAHRing, n8FAHRing, nG12AHRing, nG12FRing,n11FARing, n8ARing, n8FaRing, n4aHRing, n4FHRing, n11HRing, n10Ring, n10FRing, n4FRing, n7HRing, nHRing, n5FAHRing, n7FRing, n6HRing, n12FAHRing, n7aHRing, n10FaRing, n4FaHRing, n11aRing, nG12ARing, n10FARing, n5Ring, n3ARing, n10ARing, n11FAHRing, n11FaHRing, n9FaRing, n7FAHRing,nG12FaRing, n6FRing, n10FAHRing,n8AHRing, nG12aRing, nFRing, n5aHRing, n11Ring, n6FARing, nG12aHRing, nG12HRing, n4FAHRing, n7FHRing, n7aRing, n11aHRing, n11ARing, n7ARing, n8HRing, n4Ring, n6AHRing, n5ARing, n8Ring, n10FHRing, n8aRing, n7FaHRing, n7Ring |

*Table 7-continue. 2D descriptors calculated from Morder library*

| **Descriptor category** | **Descriptor category** |
|---|---|
| **RotatableBon** | [nRot] |

| | |
|---|---|
| **SLogP** | [SLogP, SMR] |
| **TopoPSA** | TopoPSA, TopoPSA(NO) |
| **TopologicalCharg** | GGI8, JGI7, GGI10, JGI5, JGI6, JGI1, JGI3, JGI8, JGI2, GGI9, GGI5, GGI6, GGI4, JGI4, JGI9, JGT10, JGI10, GGI2, GGI1, GGI3, GGI7 |
| **TopologicalIndex** | Diameter, Radius |
| **WalkCount** | MWC02, MWC03, TSRW10, SRW03, SRW08, MWC04, SRW04, MWC06, TMWC10, SRW07, MWC08, SRW10, MWC01, MWC10, SRW09, SRW02,MWC05, MWC09, SRW06, MWC07, SRW05 |
| **Weight** | AMW, MW |
| **WienerIndex** | WPol, WPath |
| **ZagrebIndex** | Zagreb1, mZagreb2, Zagreb2 |