ETH zürich

Scan to see our Code

529-0150-00L
Digital Chemistry
Prof. K. Jorner

Predicting Solubility of Organic Molecules in Water and Organic Solvents

Training and development of a neural network and a gradient boosting model to predict the solubility of organic molecules in various solvents using molecular fingerprints and/or calculated descriptors.

Daniel Isler[1], Tim Jürss[2], Céleste Kessler[3], Sabine Palm[4], David Walk[5]

[1]islerd@ethz.ch; [2]tjuerss@ethz.ch; [3]cekessler@ethz.ch; [4]palmsa@ethz.ch; [5]dawalk@ethz.ch;

1 Introduction

Solubility of organic molecules is crucial across multiple fields: While solubility is important for various analytical techniques, in the area of material science, and chemical synthesis, it also gives valuable insight into the bio-availability and systemic properties of drugs [1]. The accurate prediction of the solubility in different solvents for newly synthesized molecules could minimize the need for costly and time-consuming experimental studies. While several attempts have been made to employ machine learning for this purpose, it remains a challenge [2, 3].
The objective of the project is to apply machine learning methods to predict the solubility of organic molecules in various solvents in an attempt to assess the quality of inductive reasoning/generalization of solubility as a molecular feature.

2 Methods

Machine learning models were developed to predict the solubility of organic molecules using extensive datasets. The models leveraged molecular fingerprints and descriptors as input features, and rigorous hyperparameter optimization was performed to enhance predictive accuracy. Neural networks (NN) and gradient boosting (GB) techniques were applied to discover the best approach. The models were optimized and evaluated using separate validation and test sets or cross-validation (CV) methods where applicable.

Data Preparation

Two different datasets with experimental solubility values: AqSolDB [4] (aqueous solubility, 115039 compounds after curation) and BigSolDB [5] (multiple solvents at various temperatures, 52936 entries after curation), were prepared by duplicate filtering and the logarithmic values (base 10) of the solubility in mol L^{-1} was defined as the target data. The input to the models was prepared by calculating a selection of the fingerprints listed in table 1, using concatenation if more than one was selected. In an attempt to do some kind of delta-learning, the dipole moment & solvent-accessible surface area were calculated, but the optimization of the conformer ensembles was computationally expensive, so the training with these descriptors was restricted to a data set size of 368 solutes.

Table 1 Implemented input types for the ML models.

Molecular Fingerprints	Morgan, RDKit, atomic pair, and topological torsion
RDKit Descriptors	Selection of 210 available descriptors from rdkit.Chem.rdMolDescriptors [6]
QM Descriptors	Dipole moment, solvent-accessible surface area

Machine Learning Methods

Two different machine learning types were attempted on both datasets: a feed-forward neural network (implemented with PyTorch [7]) and a decision tree-based gradient boosting model (implemented with LightGBM [8], optimisation using Optuna [9]).
NN model: For the neural network, the dataset was divided into training, validation, and test sets and the input was scaled. Hyperparameter optimization was achieved by a grid search over given input parameters, optimizing the performance on the validation set. The model was evaluated statistically, using three different random states for the train/validation/test splits. In case multiple solvents are provided, for each solvent an individual model is trained and the final program can predict the solubility of the input molecule in each of these solvents.

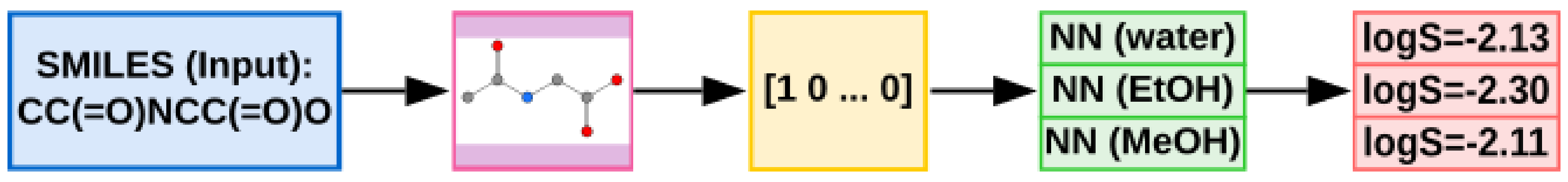
SMILES (Input): CC(=O)NCC(=O)O
[1 0 ... 0]
NN (water)
NN (EtOH)
NN (MeOH)
logS=-2.13
logS=-2.30
logS=-2.11

Fig. 1 Illustrated workflow for the neural network model. The SMILES input leads to a molecule object, from which molecular fingerprints are obtained. Different models for different solvents predict the solubility (only implemented for room temperature).

GB model: For the gradient boosting model, hyperparameter optimization was performed on the entire dataset using k-fold CV for AqSolDB and grouped (according to solvent) k-fold CV for BigSolDB. The final model takes the target molecule, target solvent and target temperature as input and predicts the solubility for these conditions.

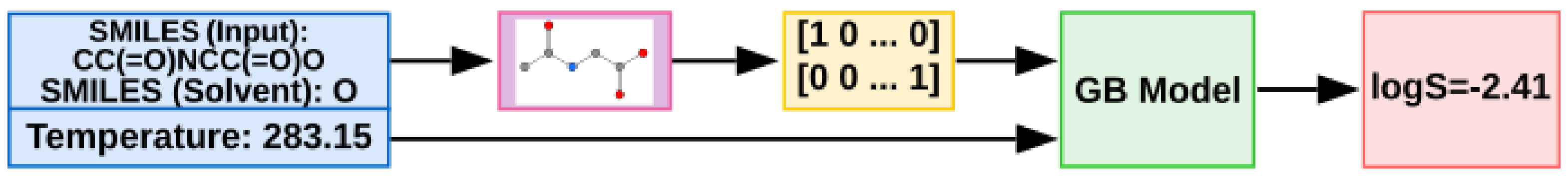
SMILES (Input): CC(=O)NCC(=O)O
SMILES (Solvent): O
Temperature: 283.15
[1 0 ... 0]
[0 0 ... 1]
GB Model
logS=-2.41

Fig. 2 Workflow for the gradient boosting model. From the SMILES string of the solvent and solute two molecule objects are calculated, from which molecular fingerprints are obtained, which are concatenated along with the temperature as features for the model.

5 References

[1] Pilar Ventosa-Andrés and Yolanda Fernández Gadea. “DRUG SOLUBILITY : IMPORTANCE AND ENHANCEMENT TECHNIQUES”. In: 2016.
[2] P. Llompart et al. “Will we ever be able to accurately predict solubility?” In: *Scientific Data* 11.1 (2024), p. 303.
[3] Arash Tayyebi et al. “Prediction of organic compound aqueous solubility using machine learning: a comparison study of descriptor-based and fingerprints-based models”. In: *Journal of Cheminformatics* 15.1 (2023), p. 99.
[4] J. Meng. *SolCuration*. `https://github.com/Mengjintao/SolCuration`. 2020.
[5] Lev Krasnov et al. “BigSolDB: Solubility Dataset of Compounds in Organic Solvents and Water in a Wide Range of Temperatures”. In: (Apr. 2023).
[6] Greg Landrum et al. *rdkit/rdkit: 2024_03_1 (Q1 2024) Release*. Version Release_2024_03_2. May 2024.
[7] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.
[8] Guolin Ke et al. “Lightgbm: A highly efficient gradient boosting decision tree”. In: *Advances in neural information processing systems* 30 (2017), pp. 3146–3154.
[9] Takuya Akiba et al. “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.
[10] Xiaomeng Ju and Matías Salibián-Barrera. “Robust boosting for regression problems”. In: *Computational Statistics & Data Analysis* 153 (2021), p. 107065. ISSN: 0167-9473.

3 Results

All trained models performed significantly better than dummy models, which predict the mean of the data in the training dataset.
As expected, thanks to more training data available, the performance was much better on the AqSolDB dataset for both models. Addition of RDKit descriptors and the more expensive dipole moment and solvent-accessible surface area to the input data increased the performance for the GB model, but did not improve the performance of the neural network. Also this was expected, as neural networks usually perform badly on little datasets. The mean squared error (MSE) loss values for the different models are provided in table 2.

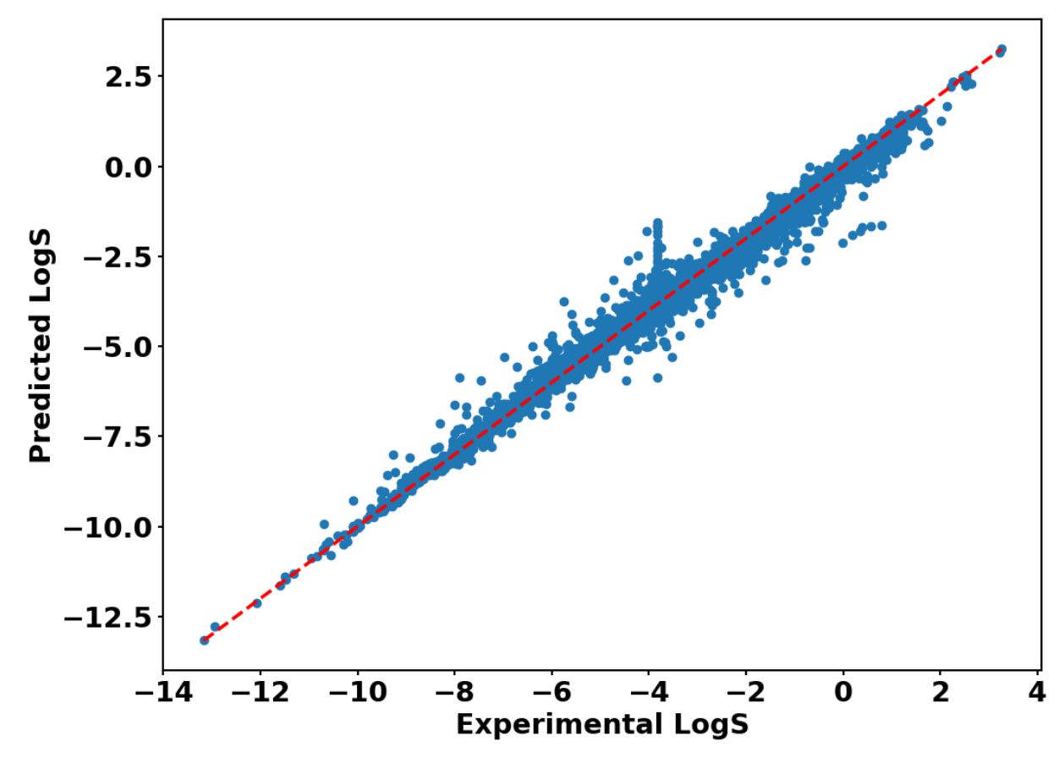

Fig. 3 Predicted LogS values plotted against their experimental LogS value for the best performing GB model trained on AqSolDB.

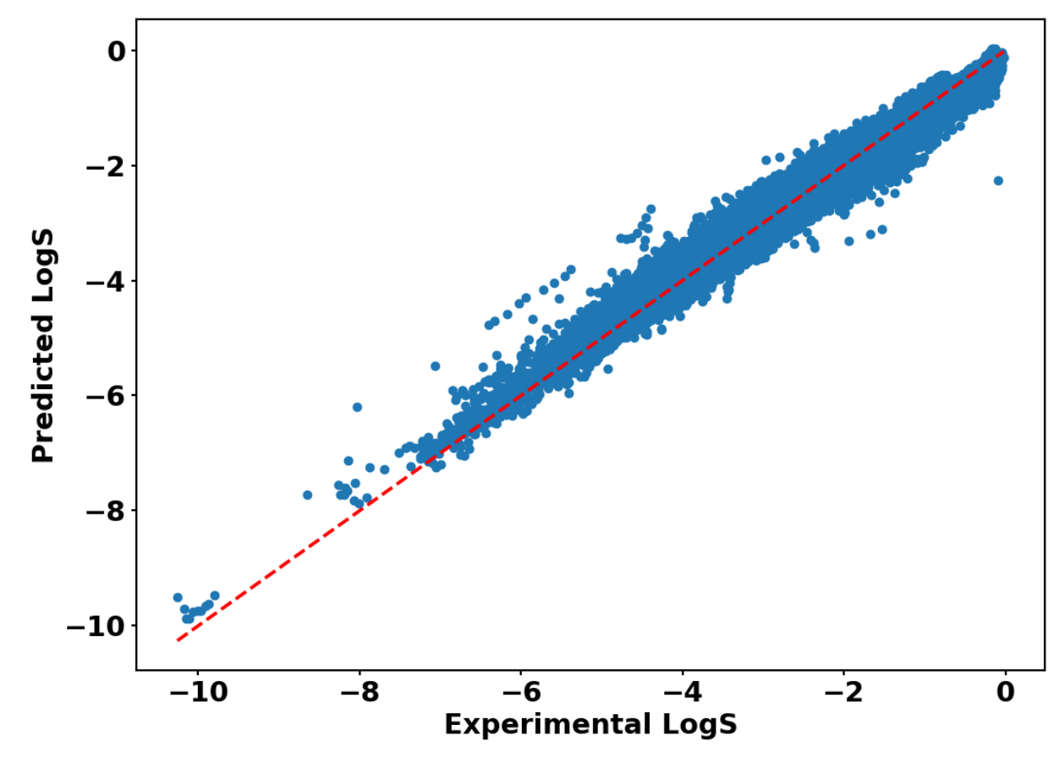

Fig. 4 Predicted LogS values plotted against their experimental LogS value for the best performing GB model trained on complete BigSolDB.

Table 2 MSE of different trained models. If available, standard deviation is indicated.

		Fingerprints	Solvents	Descriptors	MSE
AqSolDB	NN	RDKit	Water	-	$0.53(5)$
		Morgan	Water	-	$0.55(6)$
	GB	Atomic Pair	Water	-	$0.341(3)$
		Atomic Pair	Water	RDKit (15)	$0.319(3)$
BigSolDB	NN	All 4 Fingerprints	Methanol	RDKit (all)	$1.5(8)$
		All 4 Fingerprints	Water	RDKit (all)	$1.2(6)$
	GB	Atomic pair	All	RDKit (15)	$0.78(8)$
		Atomic Pair	Alcohols	RDKit (15)	$0.70(6)$

As shown in table 2, the GB model performed significantly better than the neural network. Decision tree-based GB models perform well for tabular regression problems, such as the one at hand. They do not require a large dataset for good performance, are more robust to outliers and do not need excessive feature engineering [10]. The poor performance of the neural network can be mainly attributed to the dataset size, especially for the models trained on the BigSolDB, as filtering for specific solvents resulted in very small dataset sizes. Also predicting experimental values is a difficult task and may include outliers.

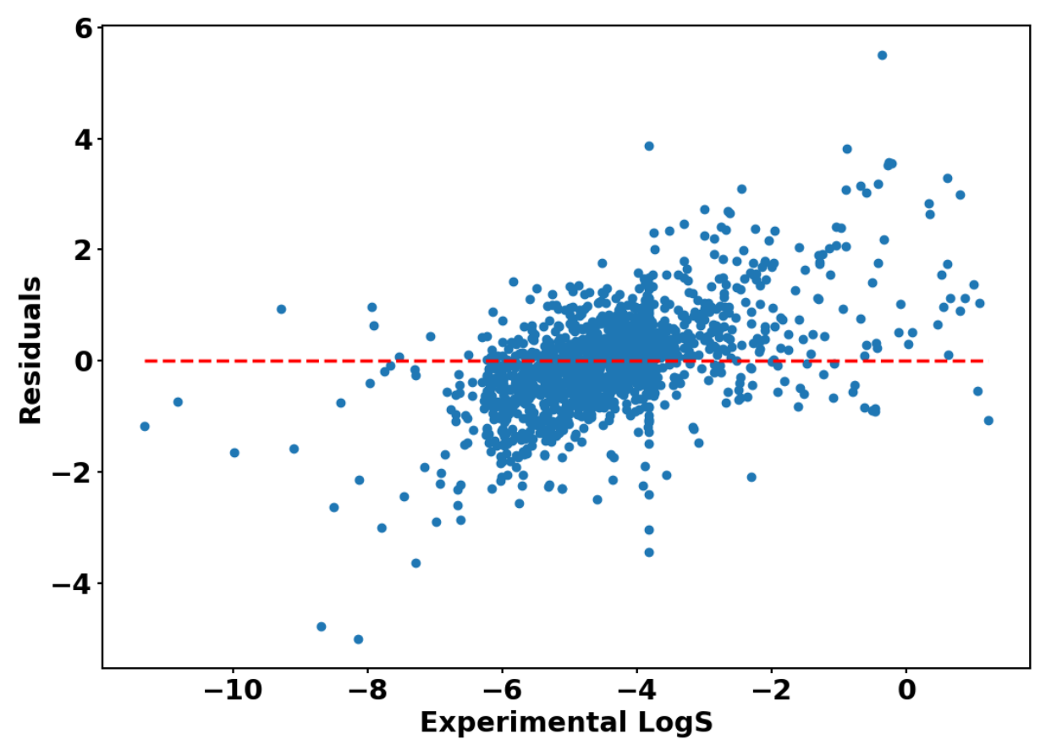

Fig. 5 Residuals as a function of experimental LogS values for the NN model trained on AqSolDB with RDKit fingerprints.

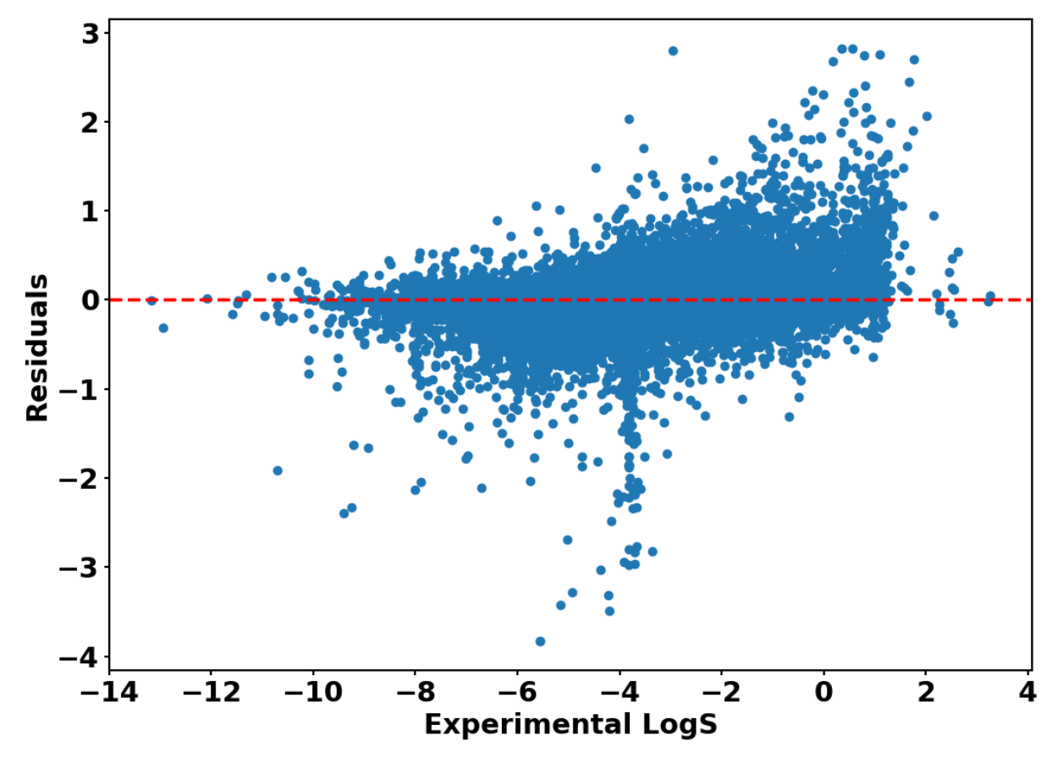

Fig. 6 Residuals as a function of experimental LogS values for the GB model trained on AqSolDB with atomic pairs fingerprints.

For both models trained on AqSolDB, there seems to be non constant variance of the residuals indicating a systematic error. They underestimate ($r > 0$) the solubility above LogS$= -4$ and overestimate ($r < 0$) it below. This suggests that the model itself might be biased by the training data in which the solubility is close to LogS$= -4$ for the majority of compounds. This effect is reduced, but still apparent, when descriptors are also used as features.

4 Conclusion

NN model: The neural network took very long to train, which is why it was not optimized for each of the fingerprint combinations. The best performance was observed for RDKit and Morgan fingerprints.
GB model: For the GB model, the atomic pair fingerprints achieved highest accuracy, which was also the case for the smaller dataset with QM descriptors. Additional RDKit descriptors lead to a significant improvement of the model.
The models in general performed worse on the BigSolDB, which is expected due to the significantly harder problem to solve and smaller data size. Further improvements could be performed by adding more relevant descriptors or by filtering for the quality of the solubility data, which is given by a weight-score. The obtained models with best performance have a MSE which is comparable to aqueous solubility models reported in the last three years (MSE of 0.27-5.0) [2].