IMPLEMENTATION OF GNN FOR PROPERTY PREDICTION USING MOLECULAR STRUCTURE

BY

KANAD SEN (22M1674)
ABHISHEK RAJ (22M1677)

GUIDED BY: PROF. ALANKAR ALANKAR

DEPARTMENT OF MECHANICAL ENGINEERING , IIT BOMBAY

ME-793

TEAM 10

OBJECTIVES OF THE PROJECT

- To understand the collected molecular dataset [**3**] given in the **SMILES string** form along with the molecular property.
- **Predict** the given **molecular property** using suitable **deep learning** models based on the derived **molecular structure.**

MOTIVATION OF OUR PROJECT

- Traditionally, the main method of modelling for molecular property is by using the **Quantitative Structure-Property relationships (QSPR)** method where we apply simple statistical model with ML. This involves developing a mathematical model that correlates the chemical and physical properties of a set of molecules to their biological or chemical activity, often using machine learning algorithms.
- However, with the advantage of increasingly sophisticated hardware and our requirements of very accurate models especially in **highly sensitive tasks** like preparation of vaccines, where there is a **very little margin of error,** we need a highly sophisticated model for our task.
- Some of the major **disadvantages of QSPR :**
  - **Availability of features:** QSPR molecules depend heavily on the availability of the sub-molecule groups' properties/features which is often not available to us. Even for same type of molecule with different variations of structure we find a large variation in property which cannot be detected by the model. Thus, QSPR is heavily dependent on its input feature size.
  - **Dataset size:** QSPR models typically require larger datasets to achieve good performance, because it is not dependent on the basic properties of molecules/atom level features like structure but only on the overall or molecular level features.
- Owing to the above mentioned disadvantages we have decided to use to **GNN models** for our molecular property prediction. This is due to the following advantages :
  - GNN models **doesn't depend on different molecular properties** or features but depend on the molecule's own structure and atom property which is easily known to us and can work with **small training sets**.
  - Gives **good accuracy on test sets** containing very different molecules as it depends on the dependent atom and bond properties of that same molecule.

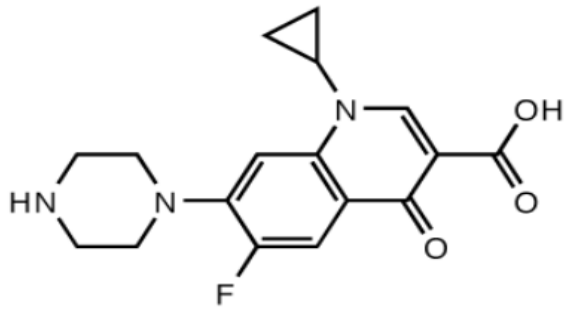
SMILES DATASET AND IT'S REPRESENTATION

S.No.	SMILES	Boiling point
1	CCC(O)CCC(F)(F)F	413.8
2	CCC(CC)ON(=O)=O	413.15
3	Clc1ccc(Cl)c(c1)C(=O)O	574.15
.....	.....	.....
5274	Clc1ccc(cc1)CC2CCC(C)(C)C2(O)CN3C=NC=N3	558.15
5275	CCOC(=O)C(Cl)Cc1cc(c(F)cc1Cl)N2N=C(C)N(C(F)F)C2=O	625.65
5276	CSC(=O)c1cccc2N=NSc12	540.15

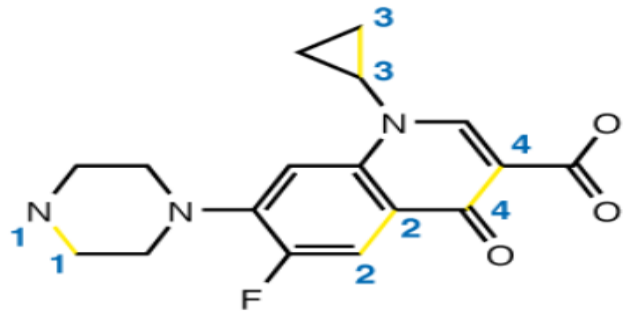
Function	Symbol
Single bond	-
Double bond	=
Triple bond	#
Aromatic bond	:
Positive charge	[C+]
Negative charge	[C-]
Aromatic carbon	c (lower case c)

- In SMILES, hydrogen are typically implicitly implied and atoms are represented by their atomic symbol enclosed in brackets unless they are elements of the "organic subset" (B, C, N, O, P, S, F, Cl, Br, and I), which do not require brackets unless they are charged. So gold would be [Au] but chlorine would be Cl.
- If hydrogen are explicitly implied brackets are used.
- A formal charge is represented by one of the symbols + or -. Single and aromatic bonds may be, and usually are, omitted. Branches are specified by enclosures in parentheses and can be nested or stacked.
- Rings are represented by breaking one single or aromatic bond in each ring, and designating this ring-closure point with a digit immediately following the atoms connected through the broken bond.

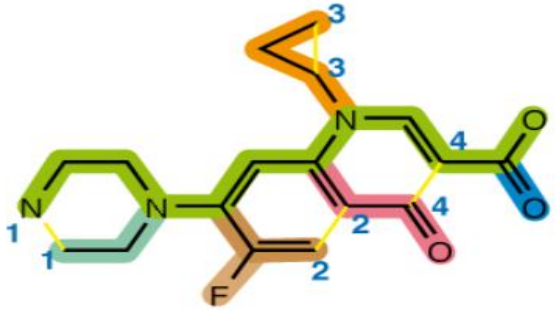
A



B



C



D

N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O

SMILES generation algorithm for A. "Ciprofloxacin":[1]
B. Break cycles, name as 1,2,3 & 4 for the four rings formed
C. Write as branches off a main backbone(green colour coded)**.**
D. Branches are of different colours kept inside the bracket.

DATA COLLECTION

- Initially we are searching for some kind of structural property relationships in the form of adjacency matrix to be used with our GNN model since the data format should be in the form of a undirected graph.
- However it was very difficult to find such kind of representations as it requires an in depth knowledge of the dataset with some complicated coding. In our search process we came across the SMILES data format wherein we have got the idea to create our own graph dataset from the molecule directly from scratch.
- While searching for our SMILES data which fits all the required criteria and a molecular property to be predicted, we have come across the paper [**3**] which has created the desired dataset and we chose to use this dataset in our current project.

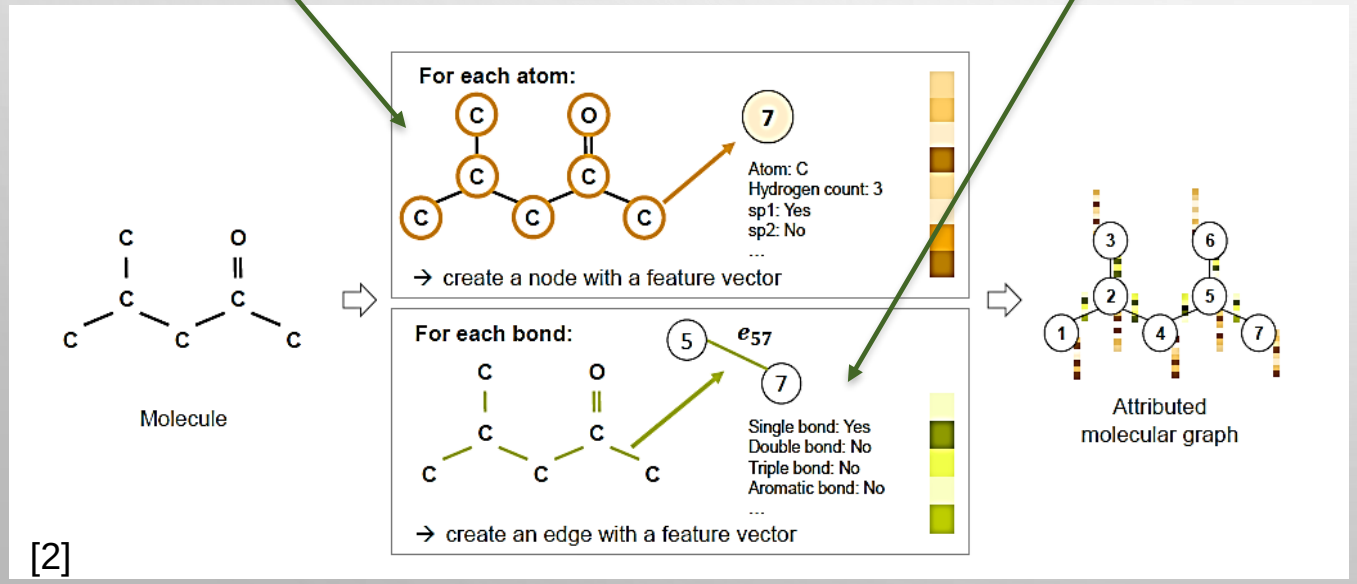
DATASET AND DATA PREPROCESSING

- Normal boiling point data set contains **5,276 boiling points for 5,089 molecules** of various classes, i.e., pure hydrocarbons, hydrocarbons with additional atoms such as oxygen, nitrogen, or fluorine, as well as multi-functional molecules.
- We will randomly select 10% of the data set for testing, i.e., 509 molecules are set aside. The test set is kept unchanged in the following. The remaining 4,749 data points will be split randomly into **90% training set** and **10% validation** . The training set and the validation set may be changed as required.
- We will convert the molecules provided as SMILES strings to attributed molecular graphs with the atom and bond features provided in **Table 1** and **Table 2 (as depicted in next slide)** respectively, with the help of **RDKit**.
- We extend the atom features stated in table 1 to the atom types occurring in the data set **(C, O, N, F, S, Cl, P, I, Br, Si)** and exclude atom features that do not occur or do not vary, i.e., **sp3d,** resulting in **29 atom features and 6 bond features** in total.
- We will **standardize the boiling** point values to a gaussian with a zero mean and standard deviation of one (z-score normalization) for training purposes. Lastly we scale the boiling points **between (0,1)** for an efficient neural network.

Atom features for initial node feature vector** [2]

Feature	Description/Exemplary values
Atom type	Type of atom (C, N, O, P, S, F, Cl, Br, I, Si)
Is in ring	Whether the atom is a part of ring
Is aromatic	Whether the atom is a part of an aromatic system
Hybridization	sp, sp2,sp3, sp3d2
Charge	Formal charge of the atom
Bonds	Number of bonds the atom is involved in
Hs	Number of bonded hydrogen atoms

Table 1



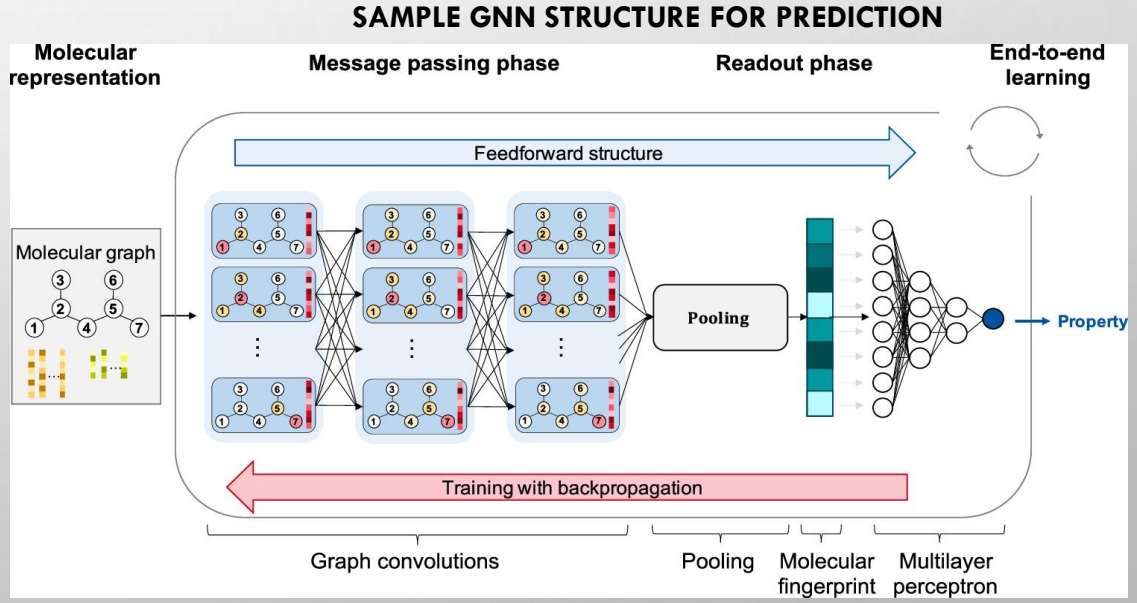
Bond features for initial edge feature vector** [2]

Feature	Description/Exemplary values
Bond type	Single, double, triple or aromatic
Conjugated	Whether the bond is conjugated
Is in ring	Whether the bond is a part of a ring

Table 2

Stereoisomer's (cis/trans) form of the molecules have not been included in the bond feature due to lack of information in our SMILES dataset.

**** All features are implemented as one-hot encoders.**



```python
# encoding of all variables

def one_of_k_encoding(x, allowable_set):
    if x not in allowable_set:
        raise Exception("input {0} not in allowable set{1}:".format(x, allowable_set))
    return list(map(lambda s: x == s, allowable_set))

def one_of_k_encoding_unk(x, allowable_set):
    """Maps inputs not in the allowable set to the last element."""
    if x not in allowable_set:
        x = allowable_set[-1]
    return list(map(lambda s: x == s, allowable_set))
```

- Due to limited space I have added only the **one-hot encoding function** used to encode the atom and bond features into numerical functions.

Why have we not done feature engineering such as (PCA,SVD) on the atom and node features ?

- In case of feature engineering such as PCA/SVD we generally use it filter out the noise or find out redundant datasets. However, one major condition is that the features and the dependent variable should be linearly dependent. In case it is non-linear we have to use t-SNE.

- We have tried to use t-SNE to apply PCA on our data. However, for that we would have to flatten the n-dimension graph data to 2 D data which will result in loss of structural information. It will be very difficult to re-convert the 2-D data into the original graphical data.

Edge indices are direction of flow of information connecting two nodes.

Torch Tensor Data to be used as input in GNN model

Node Features. One hot encoding

Edge Features. One hot encoding

```
[array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0]), array([0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 0, 0]), array([0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 0, 1, 0, 0, 0]), array([0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 0, 1, 0, 0, 0]), array([0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 0, 1, 0, 0, 0]), array([0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 0, 1, 0, 0, 0]), array([0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 0, 0]), array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0])]
[array([1, 0, 0, 0, 0, 0]), array([0, 0, 0, 1, 1, 1]), array([0, 0, 0, 1, 1, 1]), array([0, 0, 0, 1, 1, 1]), array([0, 0, 0, 1, 1, 1]), array([0
Data(x=[8, 29], edge_index=[2, 16], edge_attr=[16, 6])
```

FUTURE PLAN

- As an extension to the current project in Stage 2 we are going to use **3 Edge Conditioned Convolution network** and use it to predict the molecular property using a **Fully connected Linear layer** as mentioned in **[2].** We are also going to compare this model to other models with **different update methods.**
- We are also going to perform **hyperparameter tuning** on the model parameters.
- Use this model to predict a different molecular property and take a look at how it **generalizes on different properties**.
- We will try to compare our results obtained with a higher dimensional graph model which we will try to create to increase our accuracy.

ANSWERS TO QUESTION F.

- **Describe what activity was performed by which team member**

Kanad Sen : The majority of the programming part of the dataset like one-hot encoding part, conversion of SMILES string to bond and edge features, conversion of dataset to an acceptable input form to be used in GNN has been done by me. Also I have also contributed equally to finding the optimum dataset to be worked upon as well as finding related research papers to the topic. I have also contributed equally to prepare the slides for this presentation.
Abhishek Raj: He equally contributed to finding research papers, new ideas for the project and also in finding the dataset. He also equally contributed in preparing the slides and designing them.

- **Overall, in your opinion, how much contribution to the project was made by each member (in %)**

Kanad Sen : **60%** Abhisek Raj: **40%**

CONCLUSION

- We have learnt that GNN deep learning framework is a powerful tool to evaluate the molecular property from its structure directly.
- Careful consideration of the type of SMILES encoding is required for proper molecular information to be extracted.
- However, the process of converting the molecule into a structured data is a very time consuming process and needs to be done with careful considerations.
- PCA need not be applied on structured data as it may result in loss of information. However, another method called UMAP exists which allows feature engineering of topological data. Due to its complexity it has not been applied by us.
- Also since the model we are applying is a deep learning framework we need not apply feature engineering.

REFERENCES:

1. https://en.Wikipedia.Org/wiki/simplified_molecular-input_line-entry_system
2. https://arxiv.org/abs/2208.04852 Graph neural networks for the prediction of molecular structure-property relationships.
3. Next generation pure component property estimation models: with and without machine learning techniques
4. Yann LeCun, Yoshua Bengio, and Georey Hinton. Deep learning. Nature, 521(7553):436{444, 2015.
5. Shuo Zhang, Yang Liu, and Lei Xie. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures. arXiv preprint arXiv:2011.07457, 2020.
6. Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1):4{24, 2021.
7. Greg Landrum. RDKit: Open-source cheminformatics software. accessed on 01.04.2022.
8. Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? ArXiv preprint arXiv:1810.00826v3, 2018.
9. Kristof T. Schutt, Huziel E. Sauceda, Pieter-Jan Kindermans, Alexandre Tkatchenko, and Klaus-Robert Muller. SchNet - A deep learning architecture for molecules and materials. Journal of Chemical Physics, 48(24):1{11, 2018.