Machine Learning And Deep Learning Models For Solvation Energy Prediction

A Project Report

submitted by

TEJAS (MT21232)

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

Computational Biology
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

August 2023

THESIS CERTIFICATE

This is to certify that the thesis titled **Machine Learning And Deep Learning Models For Solvation Energy Prediction**, submitted by **Tejas**, to the Indraprastha Institute of Information Technology, Delhi, for the award of the degree of **Master of Technology**, is a bonafide record of the research work done by him under our supervision.The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr.N.Arul Murugan
Thesis Supervisor
Associate Professor
Dept. of Computational Biology
IIIT Delhi, 110020

Place: New Delhi

Date: 2nd August 2023

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and respect towards Dr.N.Arul Murugan from Indraprastha Institute of Information Technology, Delhi for being my supervisor and for exposing me to this wonderful topic of research and guiding me throughout. I would also like to thank the Department of Computational Biology. Lastly, I would also like to thank my family and friends for providing much-needed support and motivating me from time to time throughout the course of my thesis which enabled me to pursue my research in an efficient and structured manner.

ABSTRACT

Drug discovery is divided into 4 phases. The first phase is in-silico processes beginning with target identification and validation, followed by the hit discovery process, assay development, high throughput screening, preparing lead, and lead optimization. Screening through all the hits is experimentally unfeasible in terms of both time and resources. Henceforth in-silico models have been developed to predict the properties such as protein-ligand binding affinity, ligand permeability and so on. Here in this thesis, the property in focus is Solvation Energy which provides us with ligand dissolution energy during the protein-ligand binding process in an aqueous medium. The classical mechanics and quantum mechanics-based deterministic approaches can be employed to predict the solvation energies, but these approaches are computationally very demanding. Machine learning and deep learning methods can be used, which can provide reliable results and can be computationally less demanding. Here ten Graph-based deep learning models have been trained using graph representations in combination with two featurizers which help in featurizing the input molecules. Various unsupervised mechanisms like convolution, attention mechanisms, and supervised mechanism interaction networks are implemented for solvation energy prediction. These algorithms work upon graph representations constructed from input molecules by mapping atoms to vertices and bonds to edges. Apart from these three, machine learning models are also trained on different types of descriptors. Weave and CIGIN models perform best in graph-based deep learning algorithms trained on the FreeSolv dataset. Among different machine learning models, the random forest model is found to work best on both datasets of FreeSolv and MNSol. In this thesis, we establish that reliable machine learning and deep learning models can be developed for predicting the solvation energies

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ML	Machine Learning
DL	Deep Learning
MSP	Multi-level Self-supervised Pre-training
MLP	Multi Layer Perceptron
QML	Quantum Machine Learning
QC	Quantum Computing
AFP	AttentiveFP
MPNN	Message Passing Neural networks
GCN	Graph Convolutional Network
GAT	Graph Attention Network
GIN	Graph Interaction Network
CIGIN	Chemically Interpretable Graph Interaction Network
SVR	Support Vector Regression
RFR	Random Forest Regression
SVR	Linear Regression
QSVR	Quantum Support Vector Regression
VQR	Variational Quantum Regression
QNN	Quantum Neural network
COBYLA	Constrained Optimization By Linear Approximation
MNSOL	Minnesota Solvation Database

CHAPTER 1

Introduction

1.1 Virtual Screening

Searching for the best drug has been a point of interest; however, many algorithms have been designed based on available resources at the time[1][2]. With the advent of large-scale calculations with faster methods, new in-silico methods were designed for screening top candidate ligands, reducing time complexity and the large number of resources required. Virtual screening helps with fastening. In the beginning, deterministic methods were developed. Under deterministic methods, Quantum mechanical is most accurate, as biology is quantum mechanical by nature and has quantum effects in ligand-protein interactions. Hence Quantum mechanical calculations provide us with accurate calculations regarding ligand-protein interaction in different mediums[3]. Afterward comes Molecular Dynamics Simulation for the molecular mechanics model[4], which assumes Quantum properties to be static and performs classical physics-based calculations using various force fields. It provides faster calculations than QM but with less accuracy. These deterministic methods provide us with protein-ligand binding interactions with high accuracy but can be time-consuming for calculating physicochemical properties. With new-generation artificial intelligence algorithms performing well in different domains, new machine learning and deep learning-based models have been found to perform with high accuracy[5]. Such algorithms can be used for different properties prediction based on data concerning that property. This not only helps with getting a correlation between different physicochemical properties concerned with protein-ligand binding interaction. These techniques provide deep insights into different representations of molecules and how much can be learned from them. The fastening of the Virtual screening can open new paths for the drug discovery process, as we can run many more times ligands than experimental methods and thus understand ligand-protein binding interactions and the significance of various concerned properties. The basic principle for virtual screening is to find ligands on a structured basis; however, now, ML and DL-based techniques have started the data-driven approach in property prediction based on a given set of other properties. Here we study an essential property in virtual screening known as solvation energy.

1.2 Solvation Energy

For virtual screening, we study various physicochemical properties concerned with protein-ligand binding. These properties help us understand these interactions on different dimensions like thermodynamics, potential energy, kinetic energy, rate of reaction, solvation energy etc. Here we study solvation energy specifically to understand the protein-ligand binding in different mediums, enabling us to predict ligand's behavior in the host body[7]. Solvation energy is a change in Gibbs free energy, which describes the interaction of a solvent with dissolved solute or ligand molecules[6]. The strength and nature of protein-ligand interaction affect ligand properties like reactivity, solubility, and color.Properties like the medium's viscosity and density also get influenced by solvation energy. Solvation energy helps us study ligand desolvation in the thermodynamic process of protein-ligand binding. Solvation energy is directly related to chemical processes in condensed phases. It influences drug delivery, organic reactions, and electrochemical redox reactions. Hence it helps in determining these properties for drug development. Solvation energy is the difference between thermodynamic potentials regarding chemical species populations in different phases in equilibrium. Solvation energy is calculated as the chemical potential difference between two phases. Solvation energy provides free energy change emerging from molecule transfer between solvent and ideal gas. Solvation energy is related to various properties like solubilities, Henry's law constants, infinite dilution activity coefficients, and the distribution of chemical species between solvents. Solvation-free energy also provides information regarding solvent behavior in different environments. Solvation energy is a measure with combined calculations of different interactions and entropies with different calculation ranges. It provides the exacting test for quality determination of a force field which further contributes to discovering deficiencies in small molecule force fields. Solvation energy calculation for all the ligands takes much time, which can, in return, increase the time required for virtual screening; hence various Deep learning and machine learning-based models have been developed to learn and predict the solvation energy.

1.3 Prediction Methods

Solvation energy calculation experimentally is time consuming process, and using in silico methods reduces time complexity and resource consumption by a major percentage. Biology is a subject with quantum mechanics in depth of its processes, and hence in the beginning, QM methods were designed which help run quantum mechanical simulations which are faster and more accurate than conventional methods.Quantum calculations describe accurately biochemical reactions like enzyme catalysis, and photobiological systems , basically systems where biomolecular force fields fail.Under QM/MM Models, a central part of the protein site is cut out.then the surroundings are modeled explicitly using force fields. In the end, QM and MM regions together.Tight binding DFT, Self-consistent charge approximation, and reactive force fields are used for accurate solvation energy calculation under QM/MM calculation. QM calculation provides us with an accurate model of the protein-ligand binding process; however, its time-consuming and resource expensive, and, henceforth MD Simulation with classical effects was developed. It takes quantum effects as static and computes the binding interaction classically using various force fields like AMBER[8], and CHARMM[9], which help us simulate different mediums, calculate the potential and kinetic energy of the molecules and determine how they will perform under various conditions.Now with the rise of Artificial intelligence and data-driven decision making, new data-based algorithms started getting developed. Different representations are being used to build data which can be used to train algorithms and predict physicochemical properties. Some of the representations used are 1D,2D,3D fingerprints,substructure mappings,Morgan Fingerprints etc. Here graph-based representation is used to describe molecules and train the various deep learning algorithms. With these, we learn about training on GPUs and CPUs using various algorithms,compare the performances, and study how and why computational and algorithmic differences affect solvation energy prediction. With quantum algorithms on rise,quantum effects can be studied with more accuracy than classical computers however limited quantum resources limit our capability to run quantum simulations on bigger molecules.However, QML algorithms do provide us with some algorithms but for multivariate analysis, we have only multiple linear regression algorithms.

CHAPTER 2

Datasets

2.1 FreeSolv Dataset

For solvation energy prediction using data-based algorithms, a dataset with smiles and calculated and experimented hydration-free energy values was prepared by Mobley Labs[10]. This dataset is built with 643 small molecules as solutes in water as solvent. These experimented values were calculated using GAFF small molecule force field. General Amber force field (GAFF). GAFF calculates partial atomic charges and force constants using empirical and heuristic models. As water has been used as a solvent, we need to calculate its classical and quantum effects; however, calculating water molecules' effects at each step adds to time complexity, henceforth, the TIP3P water model.TIP3P model stands for transferable intermolecular potential with 3 points. For parameters, the Lennard-jones site is used for the oxygen atom, and for two charge sites, the model by Vega and Miguel is used for calculating surface tension. Shear viscosity for water is calculated at 0.321 mPa.s at 298 K and 1 bar pressure. Electrostatic interactions play a significant role in calculating solvation energy, and the AM1-BCC charges model helps us accurately calculate atomic charges.AM1-BCC provides us with atomic charges calculating a molecule's HF/6-31G* electrostatic potential (ESP).FreeSolv dataset is calculated using GROningen MAchine for Chemical Simulations package.SO FreeSOlv has experimented and calculated values from alchemical free energy calculations. Here in this thesis, FreeSolv dataset has been run on graph-based deep learning and ML algorithms. In graph-based algorithms, the molecular representations were converted into graphs by using atoms as vertices and bonds as edges. Graph representation is then trained and tested using solvation energy values provided by FreeSOlv Dataset. Graph based representations differ from each other because of different models to be trained and also because of different featurizers. The FreeSolv dataset has been broken into the ratio of 80:10:10 for Training, Validating, and Testing, respectively.FreeSolv provided us with well enough information for comparisons between deep learning models and their effectiveness in solvation energy prediction, but a smaller sample size limits its training potential in ML algorithms.

2.2 MNSolv Dataset

The Minnesota Solvation Database has 3037 solvation energy values calculated experimentally. Seven hundred ninety unique solutes are used to calculate solvation energy[11].MNSol has 92 solvents. A combination of these solutes and solvents was chosen to calculate solvation energy.MNSolv also provides us with 25 descriptors. These descriptors include eps, phi, psi, alpha, beta, gamma, and combinations of commonly found molecules like hydrocarbons, carbon dioxide etc. These descriptors include gas-phase molecular geometries. Molecules with H, C, N, O, F, Si, P, S, Cl, Br, and I atoms are only selected for this purpose. Seven hundred ninety solutes are combined from 541 neutrals and 249 singly-charged ions.MNSol dataset gives better accuracy with Machine Learning algorithms than FreeSolv because of bigger sample space and better descriptors than those used for training on Machine Learning algorithms on the FreeSolv dataset.MNSol is also used here for Quantum machine learning as it provides better results than FreeSolv descriptors-based dataset. However, qubit limit capacity makes multiple linear regression the only multivariate algorithm possible. The MNSol dataset was converted into angle embedding values for parameterized circuits to be used in the quantum machine learning algorithm. So from the MNSol dataset, a new dataset of angle embeddings was implemented using the qiskit library one by one while implementing Quantum Multiple Linear Regression. Quantum machine learning algorithm was prepared, trained, and tested to test the supremacy of Quantum systems over classical ones in the machine learning field. Although both classical and quantum machine learning algorithms were performed, there is a considerable difference in accuracies resulting from input type differences and different algorithm circuits. Machine learning algorithms got better accuracies from MNSolv than FreeSolv, which helps us figure out the best descriptor types and hence enables us to learn the correlations between these descriptors and solvation energy which helps in enhancing the further research as we can understand the best ways to represent drug candidate molecules for solvation energy prediction.

2.3 1D,2D,3D Descriptors Dataset

FreeSolv dataset provided us with smiles, experimented, and calculated solvation energies. For graph-based deep learning models, these smiles were used to build molecular graphs, which were then used to calculate features using attentive fp and canonical featurizers. Afterward, all the deep learning algorithms were trained on the basis of these calculated features against solvation energy. However, a different set of 1D,2D and 3D descriptors were calculated in this thesis using RDKIT [12] library. RDKIT library calculates these descriptors on the basis of 2D graphs, physicochemical properties, and substructure similarity or presence. In this dataset, 1826 descriptors were calculated for all of the smiles. This dataset was used for training Machine learning algorithms. Here three machine learning algorithms have been trained on this algorithm. For a better understanding of descriptors-based learning, all three models were trained on subsets of descriptors that are only on 2D3D descriptors, with only 2D and 3D descriptors, and with all the descriptors as well. 2D3D descriptors gave better results in solvation energy prediction and showed better correlation than 1D descriptors. RDKIT calculates these descriptors by first calculating the distance-bound matrix for the given molecule. This matrix is generated based on a connection table and a set of rules. The bounds matrix is not ready to be used once calculated and requires smoothing. This is performed using the triangle bounds algorithm. After smoothing, we get a proper distance bounds matrix. Now a random distance matrix is generated, which can satisfy the already generated distance bound matrix. In the final step, 3D embedding is performed, under which the entire distance matrix is embedded into the 3D dimensions. This process provides us with coordinates for each atom. Once 3D coordinates are ready for all the atoms in the molecule, coordinate data is cleaned up. The cleaning process is performed using a distance geometry force field. This distance geometry force field is calculated based on the distance constraints. These distance constraints are calculated from the distance-bound matrix calculated in the first step. This way, RDKIT provides us with 1D,2D and 3D descriptors, which can be used to train Machine learning algorithms and predict solvation energy.

CHAPTER 3

Models

3.1 Deterministic Models

Experimental methods for calculating protein-ligand binding interactions proved to be time and resource-intensive, limiting drug discovery speed. Henceforth new deterministic methods of protein-ligand binding were invented. In order to understand protein-ligand interactions binding free energy is the leading property to be predicted, which proved to be quantum mechanical and hence needed the Quantum mechanical calculations. Molecular Mechanics also provided a faster alternative for performing the binding free energy calculations with fewer resources and less time. Molecular mechanics gave results with lesser accuracy as it performed predictions on the classical level. Molecular dynamics simulation at the molecular mechanics level calculates protein-ligand interactions at a lesser accuracy but way faster. Molecular mechanics dynamics simulations provide a way to calculate protein-ligand binding properties by taking quantum effects as static, and only fewer variables are calculated.

3.1.1 Quantum Mechanical Model

Under this model, the molecule is represented as nuclei with electrons with interatomic bonds being out of consideration. Quantum mechanical methods solve the schrodinger equation using wave function related to each single atom and their electron density. Electron motion is first calculated on the basis of molecular structure, energy etc. Then the Schrodinger equation is solved for molecular calculations using electron motion. Solving the Schrodinger equation cannot be solved completely, even for one electron system, and hence we have to consider some approximations. An electron within an atom's shell cannot have any arbitrary energy or position, according to Heisenberg's principle. To solve this problem, a time-independent Schrodinger equation is used. Using this equation Hamiltonian operator is calculated(representative of the sum of kinetic energy) based on potential energy terms. Density functional theory and semiempirical calculations are used as ab initio methods for quantum mechanical calculations. However, these calculations provide us with results with low accuracy. Electron correlation methods like CCSDT, MP2 etc. are used for higher accuracy results. Using

these high-accuracy calculations, we can predict equilibrium structure, vibrational frequencies, free energy, and dipole moments. Quantum mechanical calculations prove to be really expensive when calculating activated complex and reaction pathway identification. As a solution to this, DFT and semiempirical methods are used for solving such problems using approximations. Quantum mechanical models, although highly accurate, being highly expensive in temporal and computational resources puts a limit on our computational capability.

3.1.2 Molecular Mechanics

Contrary to the Quantum mechanical approach, Molecular mechanics can be applied at a large scale with lesser computational and temporal resources. Such efficiency allows us to measure molecular structures and relative potential energies of an atom arrangement faster. Molecular mechanics consider atoms based approach and does not work on the electron level, and all the contents of an atom are worked upon collectively as a single unit. Born-Oppenheimer approximation allows for such calculations stating the possible uncoupling of nuclear motions from electronic vibration-based motions. This approximation does not provide absolute potential energy values like a quantum mechanical model; instead results in considerable differences in results for predicting energy values for different conformations. The molecular mechanics model works as ball and spring atoms represented as balls and bonds as springs. Potential energy functions and structural features like bond angles and torsional angles help us calculate such intramolecular forces. In this model, potential energy is measured as the sum of bond stretching energy, bond torsion energy, bond angular bending energy, and energy recorded from unbound atoms interactions(Van der Waals and electrostatic interaction). Molecular Mechanics does not provide us with the exact theory of working like the Schrodinger equation in quantum mechanics and considers a molecule as a set of bonded atoms. Chemical bonding-related explicit description and information about the structures of molecules is required by molecular mechanics as here molecules are being considered as calculated geometry on the basis of van der Waals and Coulombic forces. Unlike the quantum mechanical model, quantization of some physical properties like quantum entanglement, wave–particle duality, and

Heisenberg uncertainty principle is ignored by molecular mechanics, and hence this model is only able to compute molecule's energy as only a function of the nuclear positions. This results in lower accuracy in predicting physicochemical properties like solvation-free energy.

3.2 Graph Based Deep Learning Models

3.2.1 Message Passing Neural Network

MPNN [13]works with undirected graphs G with node and edge features.MPNN has two phases a message-passing phase and a readout phase. The message passing phase is defined in terms of message functions Mt and vertex update functions Ut where t is time steps. Message Passing phase includes the updation of hidden states at each node based on messages according to G. In the readout phase, a feature vector is computed for the whole graph using the readout function R. Message function Mt, vertex update function Ut and readout function R are all learned differentiable functions. For MPNN to be invariant to graph isomorphism, readout function R has to be invariant to permutations of the node states.

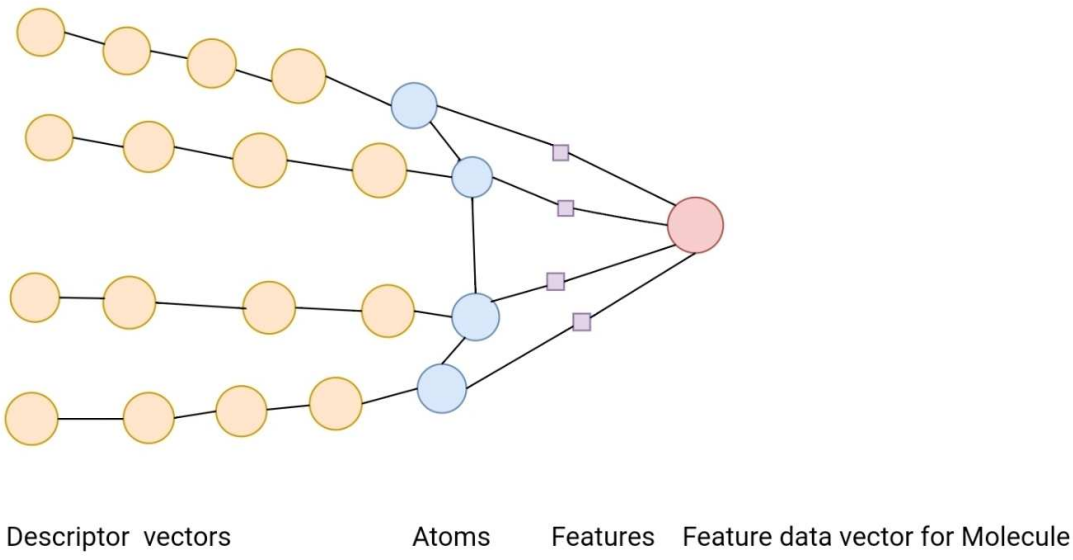

Figure 3.1: MPNN

3.2.2 WEAVE

The WEAVE model encodes the molecule into a list of atomic features and a pair matrix based on bonds[14]. Four sets of fully connected layers (corresponding to four paths from two original to two updated features) fed Weave features. Then in the next step, these sets are concatenated to form new atomic and pair features. A gathered layer made of stacked weave modules combines atomic features to form molecular features that are task-specific layers. The weave module is only different from MPNN for keeping the bond features.

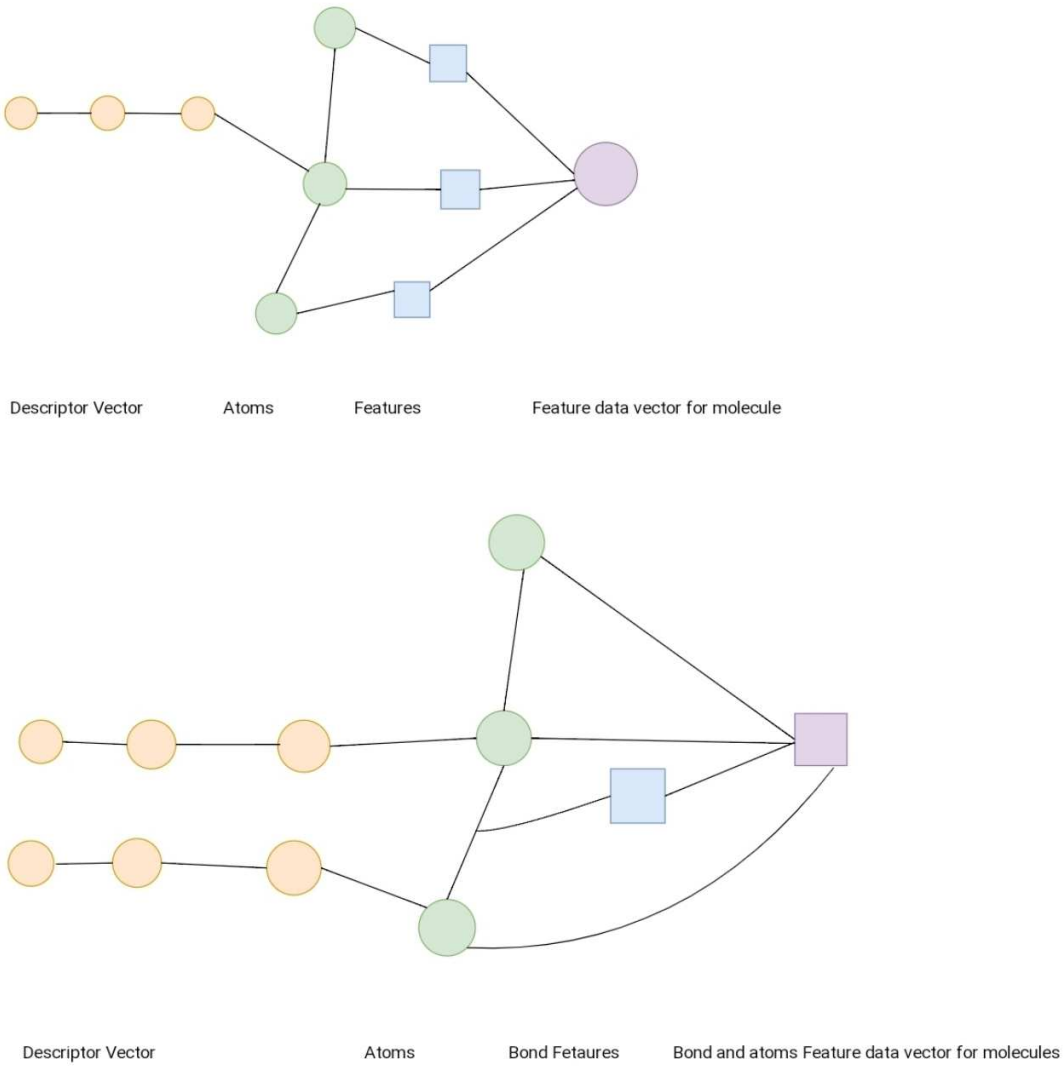

Figure 3.2: Weave Model

3.2.3 GCN

GCN is a version of circular fingerprint[15]. GCN uses differentiable network layers instead of fixed hash functions. Each convolutional layer extends central atom features by applying convolutional functions on itself and its neighbors' atoms. A neighbor list and a set of initial feature vectors corresponding to a single atom are used for molecule representation. Feature vector has the atom's local chemical environment, which includes atom types, hybridization types, and valence structures.

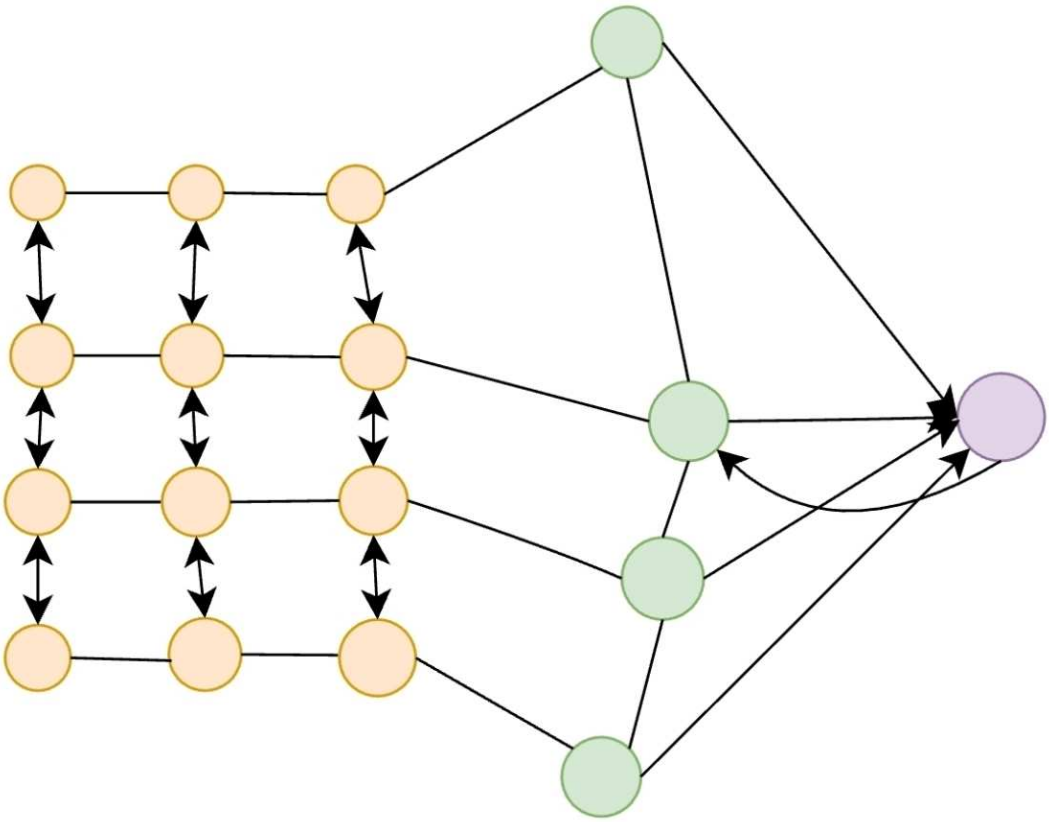

Figure 3.3: GCN Model

3.2.4 GAT

GAT is built from the graph attention layer[16]. The attention layer uses a soft-attention mechanism for mapping a set of node features to a new set of node features. The graph attention layer outputs a weighted average for all of the nodes

and their neighbors where the weights are not fixed and are input independent. Non-linearity is applied to linear combinations. Weights in attention mechanisms are determined based on graph features. Weights are computed using the softmax formula for normalization.

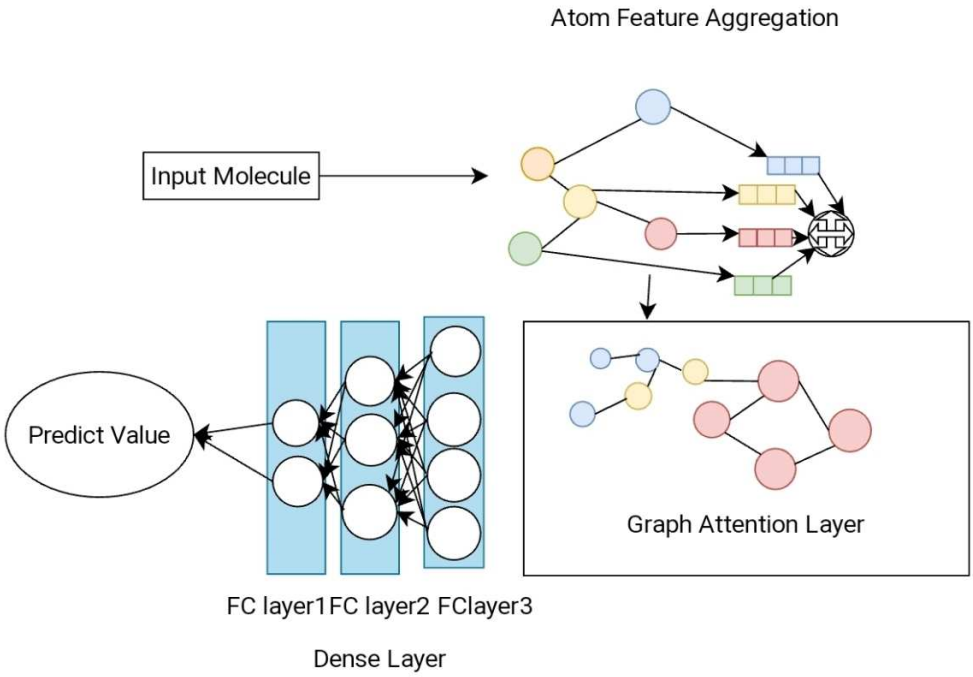

Figure 3.4: GAT Model

3.2.5 AttentiveFP

AttentiveFP considers bond and atomic features from a molecule to be extracted with RDKit and encoded[17]. First, fully connected layers are made, and initial state vectors are generated for each atom and its neighbors. Initial state vectors and stacked attentive layers are embedded for node embedding. Atomic messages from its neighborhoods are aggregated progressively. A new state vector is generated in each node embedding the attentive layer. To combine the individual atom state vectors into a full-molecule state vector, the entire molecule is a super virtual node that connects every atom in a molecule and is embedded. This is done for molecule embedding and generates a state vector for the whole molecule. This process is performed on stacked attentive layers. Structural information regarding

molecular graphs is encoded, and a learned representation is prepared from this as the final state vector.

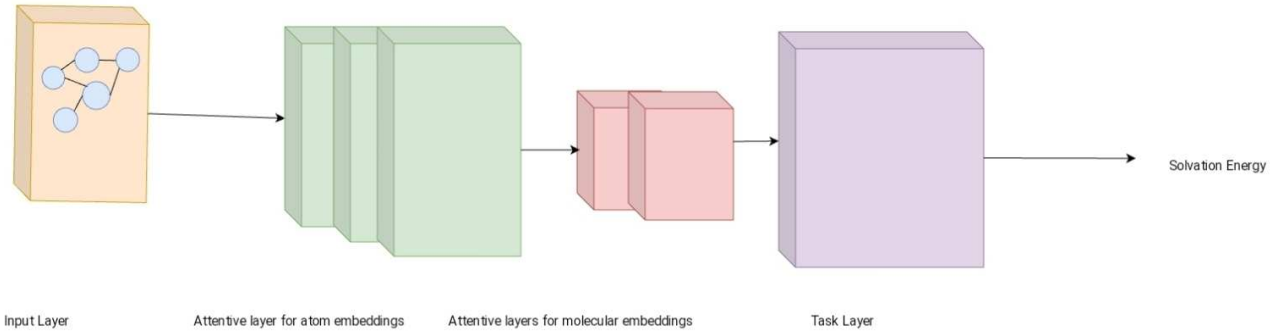

Figure 3.5: Attentive FP Model

3.2.6 GIN

GIN transforms SMILES to a graph G=(V, E) based on RDKIT41[18]. Then we decompose this graph into a series of motifs. Decomposition adds a rule based on BRICS42 i.e. break large ring fragments and select their minimum rings as generated motifs. All the nodes are incorporated into V and edges between the corresponding atoms and motifs Em are merged into E. Next, the graph level node Vg is augmented, and edges between all motifs and Vg are appended into the initial graph G. This graph is fed to GNNs which learn hierarchical presentations. In the MSP process, atomic representations are used to predict atom types, bond links, and bound types. Cross entropy loss and smooth L1 loss are respectively applied to optimize the atom and molecule level learning. For fine-tuning, the graph-level representations pass through a two-layer MLP to predict molecular properties. The pre-trained weights are transferred to the mfine tuning model and updated under the labels' supervision.

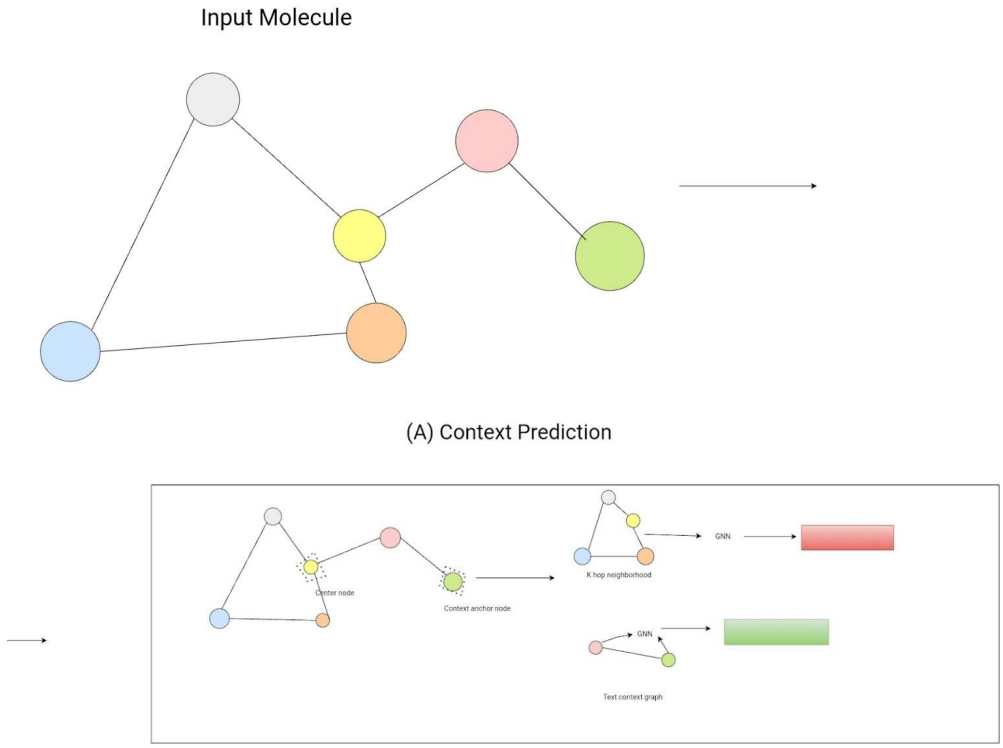

Figure 3.6: GIN Model

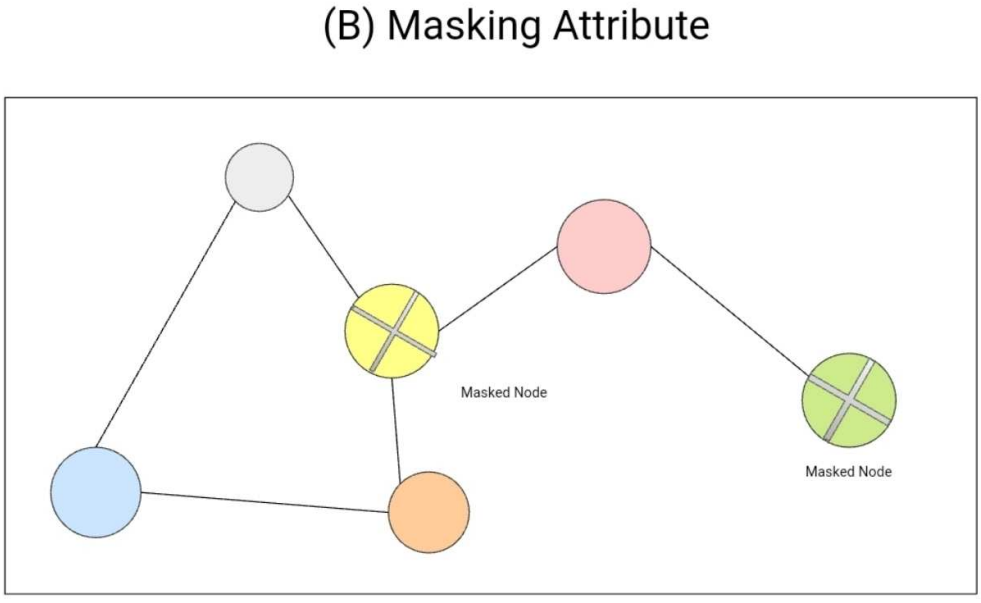

Figure 3.7: GIN Model Masking Attribute

3.2.7 CIGIN

Feature Representation:

Atom Features	Description
Atom Type	H, C,N, O, F.. (one-hot)
Implicit Valence	Has Implicit Valence (Binary)
Radical Electrons	Has Radical Electrons (Binary)
Chirality	R,S or None (one-hot)
Number of Hydrogens	Number of neighboring Hydrogen atoms (one-hot)
Hybridization	sp, sp<, sp*, spšd (one-hot)
Acidic	Acidic in Nature (Binary)
Basic	Basic in Nature (Binary
Aromatic	Part of aromatic group (Binary)
Donor	Donates electron (Binary)
Acceptor	Accepts electron (Binary)

CIGIN has a message-passing phase, and the interaction phase is based on the MPNN model. The only difference between CIGIN and MPNN is that it uses MPNN for both solute and solvent[19].The prediction phase predicts solvation energy. For this message passing phase and interaction phase are run for both solute and solvent and combined based on atoms. Then, a readout layer R is used to combine the feature vectors across all the atoms in a manner invariant to graph isomorphism in order to obtain a one-dimensional vector.A"=Rsolute(A, A') B" = Rsolvent(B, B')Here, we have two choices of R, the first one is sum pooling along the atom dimension, and the second is a set2set layer[24], The outputs A" and B" are concatenated and are passed through three fully connected layers to predict the free energy of solvation. The intermediate layers have ReLU as the activation function. A readout layer combines the feature vectors across all the atoms in a manner invariant to graph isomorphism and calculates a 1D Vector.

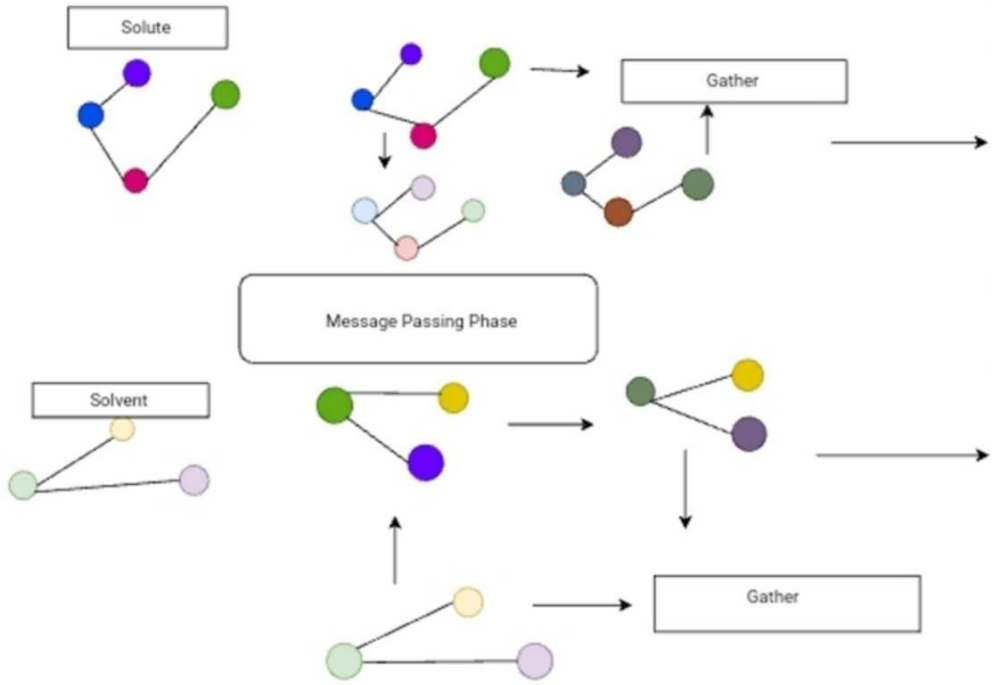

Figure 3.8: CIGIN Model Message Passing Phase

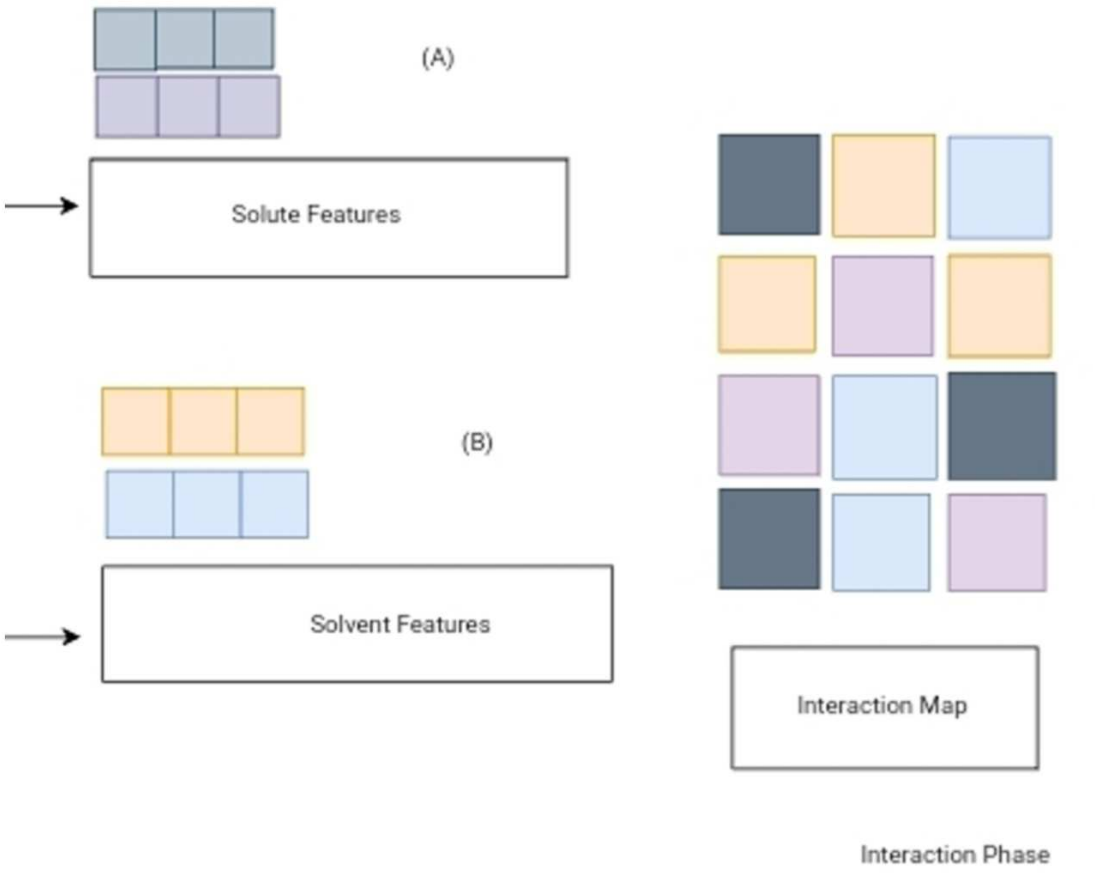

Figure 3.9: CIGIN Model Interaction Phase

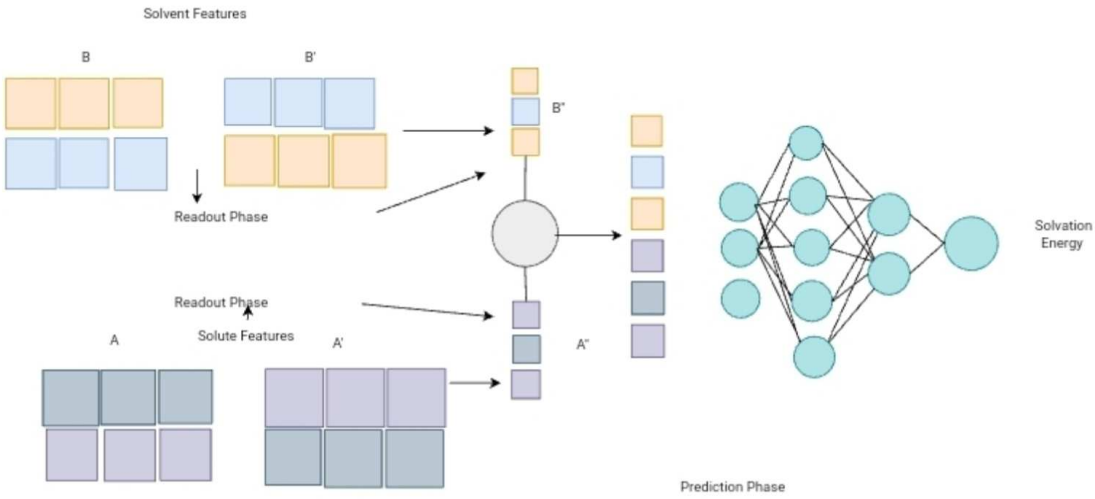

Figure 3.10: CIGIN Model Prediction Phase

3.3 Featurizers

3.3.1 Canonical Featurizer

This featurizer has following features[20]:

- One hot encoding of the atom type.
- The supported atom types include C, N, O, S, F,Si, P, cl, Br, Mg, Na,ca,Fe,As, Al,I,B,V,K,TI,Yb,Sb,Sn,Ag,Pd,Hg,Pb,Zn,Ti,Cd,Pt,Zr,Ge,Co,Se,In,Ni,Au,M n,Cr,Cu,Li,H.v
- One hot encoding of the atom degree. The supported possibilities include 0-10
- One hot encoding of the number of implicit Hs on the atom. The supported possibilities include 0-6
- Formal charge of the atom.
- Number of radical electrons of the atom.
- One hot encoding of the atom hybridization. The supported possibilities include SP,SP2,SP3 , SP3D , SP3D2
- Whether the atom is aromatic.
- One hot encoding of the number of total Hs on the atom. The supported possibilities include 0-4.

3.3.2 AttentiveFP Featurizer

The atom features include:

- One hot encoding of the atom type. The supported atom types include B, C, N,si, P,S, C1, As, Se, Br, Te, I, At ,F and other
- One hot encoding of the atom degree. The supported possibilities include 0-5
- Formal charge of the atom
- Number of radical electrons of the atom
- One hot encoding of the atom hybridization. The supported possibilities include SP,SP2,SP3 , SP3D, SP3D2, and other
- Whether the atom is aromatic
- One hot encoding of the number of total Hs on the atom. The supported possibilities include 0-4.

- Whether the atom is chiral center
- One hot encoding of the atom chirality type.The supported possibilities include R,and S.

3.4 :Descriptor based ML Models

Support Vector Regression:

Support Vector Machine[21] calculates a hyper-plane and creates a classification between the data types. In 2-D space, this classification is represented by a line. We plot data items in an N-dimensional space, where N is the number of features/attributes in the data. Next optimal hyperplane is calculated for data separation. We use this for regression on various classes of Descriptors(1D,2D and 3D) together and separately.

Random Forest Regression:

Random Forest can perform both regression and classification using multiple decision trees, Bootstrap and Aggregation. We combine multiple trees instead of relying on individual trees[22]. Random row sampling and feature sampling are performed. This part is called Bootstrap. We use this for regression on various classes of Descriptors(1D,2D and 3D) together and separately.

Linear Regression:

This function computes a linear relationship between a dependent variable with one or more independent variables[23]. There are two types, univariate and multivariate, based on no. of variables. Linear regression finds an equation that provides a straight line that represents. The goal of this algorithm is to find the best linear equations and predict the values of independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s). We use this for regression on various classes of

Descriptors(1D,2D, and 3D).

Quantum Machine Learning

For Quantum machine learning QSVR, Qiskit Estimator Regressor, and VQR algorithms are built for univariate regression by IBM Qiskit. The only Quantum machine learning multivariate algorithm possible is Multiple Linear regression, in which we calculate the slope for all the variables, then multiply them with their predicted results and add all the values. These result values are then compared to experimental results for evaluation. However, the results were quite similar for the three algorithms, as suggested by qiskit. The results for all three algorithms gave Test PCC 0.046 with both CPU and GPU and performed really poorly for multiple linear regression if compared to classical ML and DL algorithms run on CPU and GPU, respectively. Multiple linear regression is the only option available for such ML algorithms as these regressors are run using single quantum bits prepared using angle embedding and optimized using COBYLA optimizer from IBM. Preparing equal qubits for all variables will be required, which still needs to be feasible with current-state quantum computers. Hybrid Quantum Machine learning algorithms were also run against various descriptors using GPU and CPU; however, the CPU wasn't able to convert parameterized values to classical values and vice versa.GPU was able to do so, but the classical to parameterized circuit values conversion took a big part of the total time taken per run. Hence batch processing was used, but still, the Hybrid algorithm could have provided better results as it had the same problem of running on a single qubit and hence the same multivariate(multiple linear regression) algorithms. For Quantum machine learning Purposes, the MNSol dataset was used all these variables were converted into embedded qubits. Each variable was embedded in qubits. A Qubit can be physically any quantum particle like a boson, photon etc. This quantum particle is prepared as a qubit by various forms of embedding, which encodes our information as a quantum state on a single quantum particle. A univariate quantum machine learning regression model is prepared by qiskit to be run on a parameterized circuit. This approximation, however, results in predicted values being way less correlated to solvation energy than classical descriptor based machine learning algorithms.

CHAPTER 4

Methodology

4.1 Graph based deep learning model

4.1.1 Message Passing Neural Network

Graph based deep learning models:

MPNN:

First, we train the Message Passing Neural Network model. For this model, we generate features either on Attentive FP or Canonical featurizer. These featurizers provide us with information about input molecules. Here in the MPNN model, we generate features for the solute molecule from the smile file. These features provide us with information regarding classical information for each atom. For solvent, we have the TIP3P water model as we are calculating only water for solvent. This phase is called the Message passing phase. Afterward, features from the TIP3P model and solute features calculated from the first phase interacted with each other. This phase provides us with a collective information vector for a combination of solute molecules with solvent molecules. This is required for solvation energy correlation as solvation energy changes with changes in either solute or solvent features. This is known as the Interaction phase. In the last phase, the output vector from the interaction phase and respective solvation energy are fed to the Neural network in which we have a dense layer of neurons that get the vector values as input to the input neural layer. This neural network in the last layer predicts the solvation energy. This value is compared to the experimental value, and weights and biases are changed, respectively.MPNN model is later saved and then used for solvation energy prediction for input solute molecule automatically considering water as solvent.

4.1.2 WEAVE Model

Next WEAVE Model is implemented for solvation energy prediction. In the WEAVE model, first, the features are generated from solute molecules based on the WEAVE featurizer. Weave featurizer takes as a basis either Attentive FP featurizer or Canonical featurizer. Weave Featurizer has a unique quality that it first

generates feature vectors for each atom on the basis of aforementioned featurizers but also builds a vector on the basis of bonds between atoms. Hence Weave model prepares the vector for each atom on the basis of the atom itself and its neighboring atoms. This type of featurization helps the Weave model better understand molecular geometry. Now all the vectors from each atom are concatenated into one complete vector, which provides us with a complete features list for an entire molecule. This vector is inspired by the Message passing phase in MPNN. In the next step, the output vector for the input solute molecule interacting with solvent features is calculated from the TIP3P water model. Calculating water features adds to the time complexity of the molecule. The TIP3P model has calculated water features, which are now interacted with the solute. Here respective features from the solute and water model(solvent) interact. This interaction provides a complete interaction vector with all the feature mappings for the complete molecule as output. This output vector and respective solvation energy are then fed as input to the input layer of our neural network, which helps us train the neural network for solvation energy prediction. This neural network outputs the predicted solvation energy value, which is then used to calculate the loss factor, which in turn helps us in calculating new sets of weights and biases.

4.1.3 Graph Covolutional Network

GCN model implements a neural network based on circular fingerprints. GCN first computes features for each atom in a solute molecule using either Attentive FP or Canonical Featurizer. Now on the basis of circular fingerprint, differentiable network layers are used. Here the information regarding neighboring atoms is also taken with each atom. This is used in extending central atom features by each convolutional layer. This is performed by applying each atom's convolutional functions on not only itself and neighboring atoms as well. In this way, GCN Model uses each atom's initial feature vectors and the neighbor list for better molecule representation.GCN predicts solvation energy based on features like hybridization, valency etc. This predicted value is now used for loss calculation. This loss value then helps us calculate and update the weights and biases if necessary.

4.1.4 GAT

GAT is built on the basis of the attentive layers concept of the deep learning mechanism. First, the features are generated for each atom using Attentive FP or Canonical featurizer. Afterward, these features are combined to build one molecular representation. This combined feature vector is then fed as input to the neural network with attentive layers between the input and output layers. As the input layer feeds the attentive layer with molecular representation along with initial weights and biases, the attentive layer maps a set of node features to a set of new node features using a soft attention mechanism.GAT works on each atom along with its neighbors, so its attentive layer outputs a weighted average for all the nodes and neighbors. Weights for nodes and their neighbors are independent of the input provided. This helps us in carrying the training in non-linear functions contrary to the linear nature of the problem. Graph features play a significant role in determining the new set of weights. The last layer in GAT neural network computes the softmax function for solvation energy prediction, which helps us compute new weights for the model to be retrained.

4.1.5 Attentive FP

The attentive FP model is built up on attentive layers in between input and output layers. Attentive FP model works with both Canonical and its own Attentive FP featurizer.This model uses bond and atomic features extracted using RDKit. The input layer is fed the feature vector is calculated in the first step. This input layer outputs these values with initial weights calculated on the basis of graph features to the input for fully connected layers. State vectors for each atom and its neighbors help in calculating initial weights for attentive layer input. Now initial state vectors are embedded with stacked attentive layers for node embedding. Atomic messages play an important role here and are thus aggregated progressively in embedding. Now each attentive layer provides nodal descriptors for each atom. The complete molecule vector is computed in the next attentive layer using the output of stacked attentive layers working for each atom. The nodal state vectors thus computed help complete molecule attentive layer build molecular representation as a super

virtual node. This layer outputs a single state vector for the whole molecule, which has connected embedding from all the atoms present in the input molecule. The output state vector computed from the last attentive layer for the whole molecule is now encoded, and a molecular representation is prepared to be learned fro solvation energy prediction. In the end, the output layer predicts the solvation energy from which we compute the loss value for updating the weights.

4.1.6 GIN

GIN model first calculates the features for each atom using RDKit features based on Attentive FP or Canonical featurizers. First, the molecule is input in smile format which helps us in building a 2D graph. This 2D graph is broken into a series of motifs. This procedure is carried out on the basis of the BRICS 42 rule.BRICS 42 rule suggests breaking large ring fragments and selecting their minimum rings as generated motifs. Graph representation of molecule reads all atoms as nodes and bonds as edges. The state vectors correspond to the nodes and edges of the graph motif, and information is also added to the edge vectors. Now the graph node vectors are manipulated based on motifs and edge vectors merger. Now based on nodes and edges new graph is built. Now this graph is input to a Graph neural network or GNN for learning the representation against the corresponding solvation energy value. The MSP process is used for learning hierarchical representations based on atomic representations. Atomic representations are used to compute molecular features. Smooth L1 loss and cross-entropy loss are used for optimizing molecule-level learning. A 2 layer MLP model is also implemented for fine-tuning the representations output from graph evaluations. Now the pre-trained model weights are transferred to the mfine tuning model, which gets updated under label-based supervision. In the end, solvation energy is predicted, which helps in updating weights.

4.1.7 CIGIN

CIGIN is based on message-passing neural networks. Here the features are generated for each atom using a set of descriptors. The CIGIN model calculates

the features for both solute and solvent for higher accuracy.CIGIN works with solvents other than water as well. First, the features calculated from both solute and solvent are combined to prepare molecular representations for both solute and solvent. These values are then output to the next phase for interaction. In this phase, relative solute and solvent features interact with each other, and a combined representation of solute and solvent is prepared, which has a high correlation to solvation energy. Now this combined vector is input to the neural network for solvation energy prediction. Loss value is calculated from output solvation energy, and weights are updated for better predictions.

4.2 Descriptors Based Machine Learning Algorithms

4.2.1 SVR

SVR is used on 1D, 2D, and 3D descriptors calculated from RDKit for FreeSolv and for 25 descriptors from MNSolv Dataset.SVR classifies between types of data on the basis of hyper-plane computation. All the data values are basically read as marked points in a T-dimensional space, where T is a number of features. However, SVR can also be used for regression purposes. Here SVR is used for the regression purpose of predicting solvation energy by learning various sets of descriptors against corresponding solvation energy values.

4.2.2 RFR

RFR uses multiple decision trees, bootstrap, and aggregation. RFR splits the data into different sets of split data pieces. RFR is built for classification but can also be used for regression with slight manipulations. The multiple decision trees are first prepared and then regressed. Afterward, these are combined for better performance than individual trees. Feature and random row sampling are used in RFR to learn better the features and their correlations to the solvation energy prediction. These two types of sampling combined are called bootstrapping.RFR

is trained on different sets of 1D,2D and 3D descriptors separately.

4.2.3 LR

In this ML model, we study the linear relationship between 1D, 2D, and 3D descriptors and solvation energy. Here we run a multivariate regression to find an equation for a straight line for our solvation energy prediction problem based on descriptors. This algorithm keeps updating the linear equation for better prediction accuracy. Metrics used for rating the performance is basically the slope of the straight line, as this slope indicates the change in the dependent variable for a unit change in independent variables.

CHAPTER 5

Statistics and Graphs

5.1 Statistics

5.1.1 ML Results

For SVR on 1D,2D and 3D descriptors

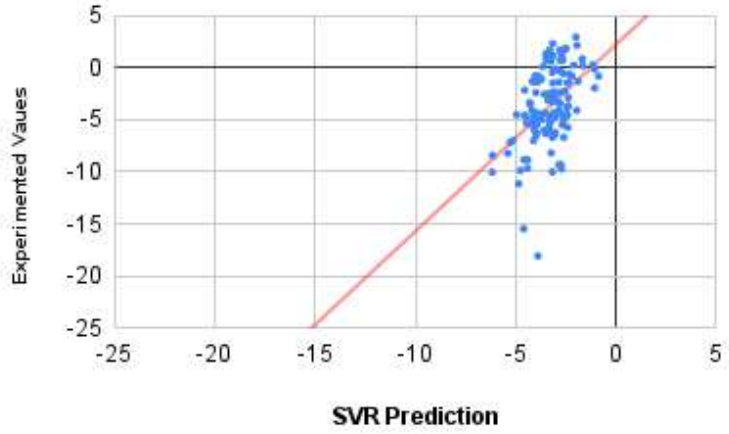

Figure 5.1: SVR with 1D 2D 3D Descriptors

For RFR on 1D,2D and 3D descriptors

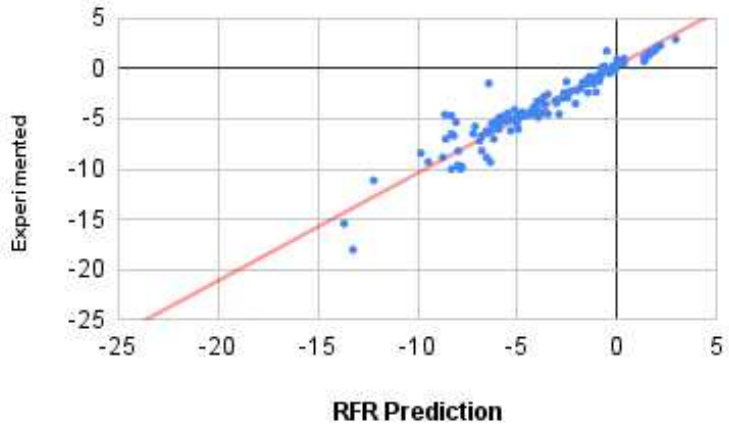

Figure 5.2: RFR with 1D 2D 3D Descriptors

For LR on 1D,2D and 3D descriptors

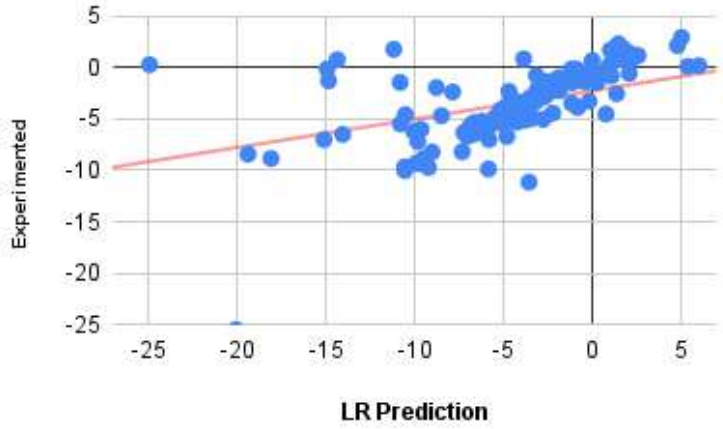

Figure 5.3: LR with 1D 2D 3D Descriptors

For SVR on 2D and 3D descriptors

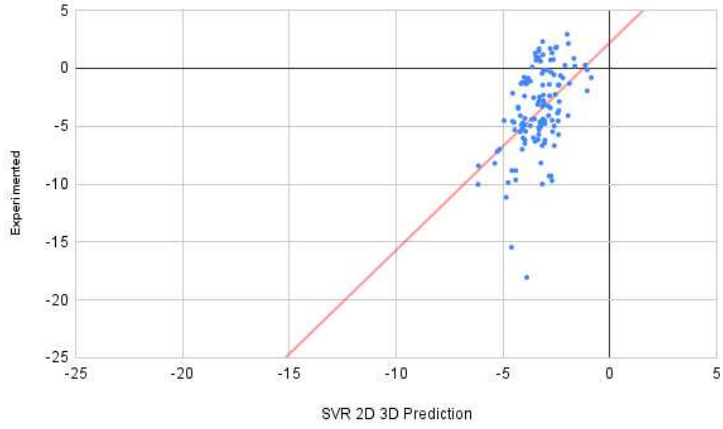

Figure 5.4: SVR with 2D 3D Descriptors

For RFR on 2D and 3D descriptors

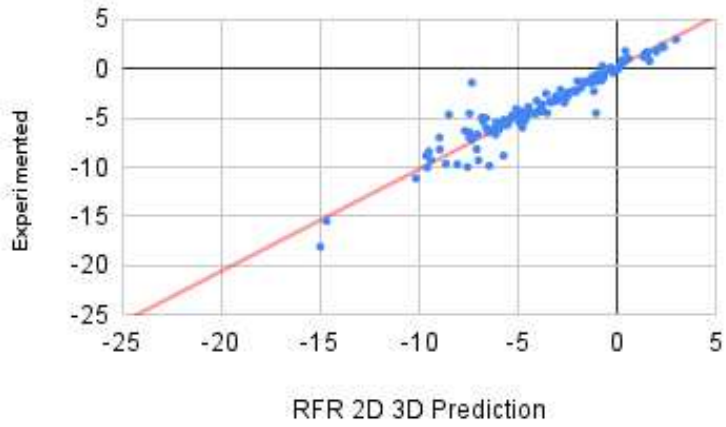

Figure 5.5: RFR with 2D 3D Descriptors

For LR on 2D and 3D descriptors

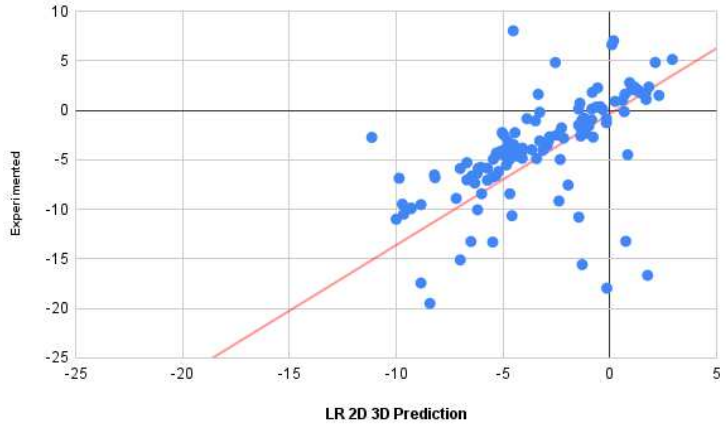

Figure 5.6: LR with 2D 3D Descriptors

For SVR on MNSol

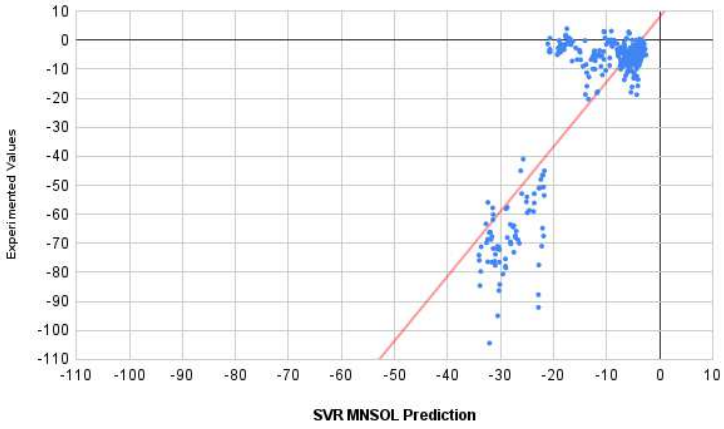

Figure 5.7: SVR with MNSol Descriptors

For RFR on MNSol

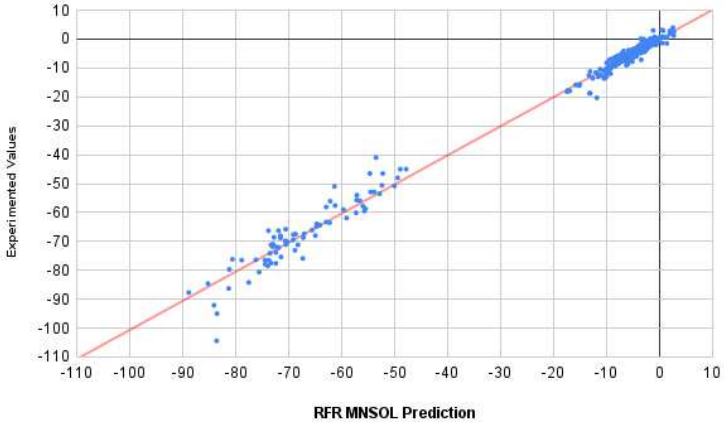

Figure 5.8: RFR with MNSol Descriptors

For LR on MNSolv

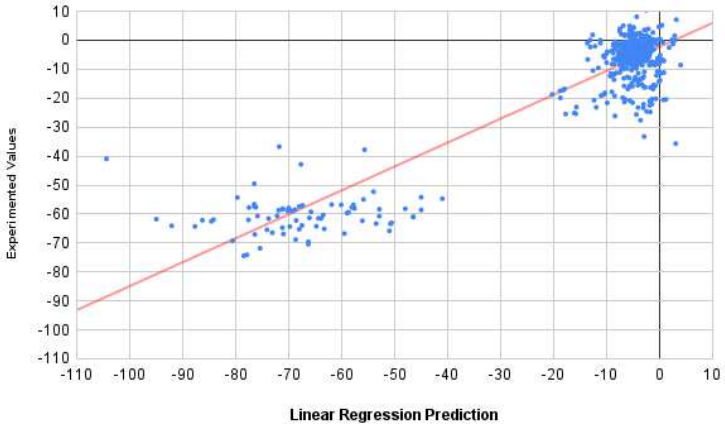

Figure 5.9: LR with MNSol Descriptors

5.1.2 Best Graph based learning models

Best Test PCC in MoleculeNet:Weave Model with AttentiveFP featurizer

Best Test RMSE in MoleculeNet:Weave Model with AttentiveFP featurizer

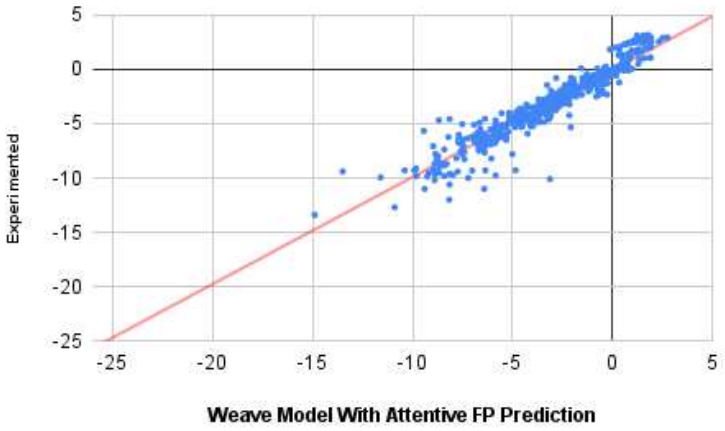

Figure 5.10: Weave Model with AttentiveFP featurizer

CIGIN Test PCC:0.966

CIGIN Test RMSE:0.99

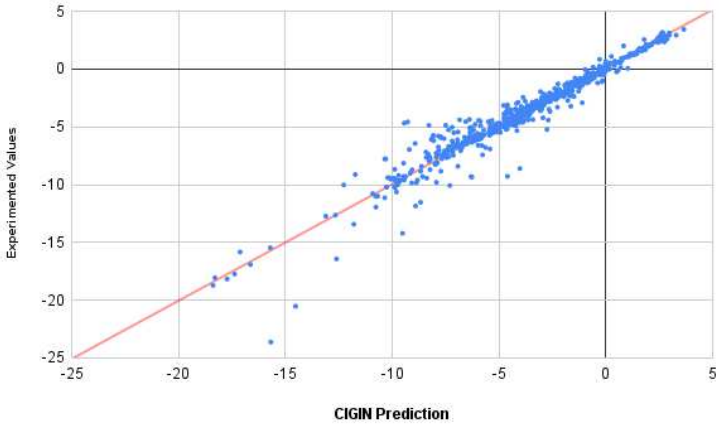

Figure 5.11: CIGIN Model Correlation Graph

5.1.3 Best Descriptors based Machine Learning Models

FreeSolv Test Best PCC:SVR with 2D 3D (0.959)

FreeSolv Test Best RMSE:SVR with 2D 3D (1.15)

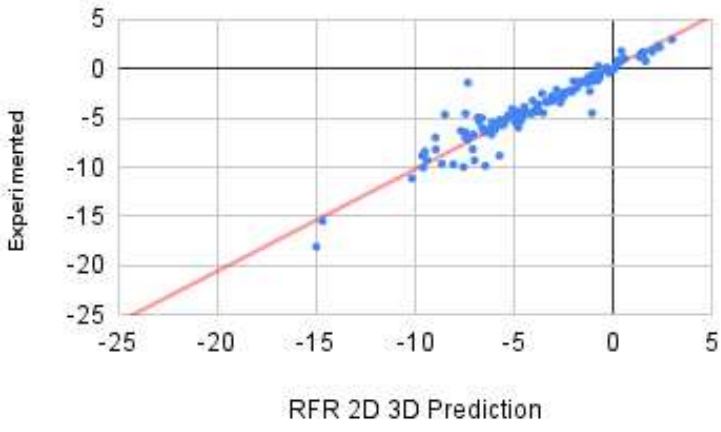

Figure 5.12: Best ML Model with FreeSolv Dataset

MNSol Test Best PCC:RFR(0.996)

MNSol Test Best RMSE:RFR (1.88)

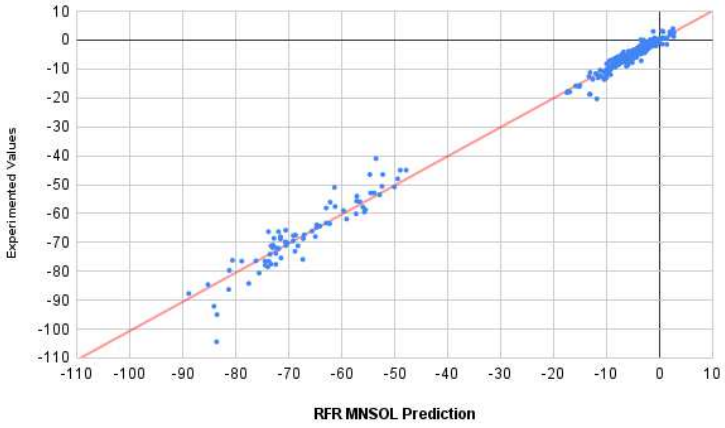

Figure 5.13: Best ML Model with MNSol Dataset

5.1.4 Tables

Graph Based Deep Learning PCC Tables

Models	Featurizer	PCC Values		
		Train	Test	Val
MPNN	Canonical	0.833631	0.885083	0.534303
	Attentive FP	0.962413	0.936518	0.815327
Attentive FP	Canonical	0.926332	0.867609	0.745614
	Attentive FP	0.949709	0.954456	0.840403
GAT	Canonical	0.900063	0.739939	0.707561
	Attentive FP	0.918125	0.944659	0.818147
GCN	Canonical	0.950873	0.748973	0.79698
	Attentive FP	0.934125	0.961473	0.766291
gin edge pred	NA	0.701807	0.862823	0.69781
gin context pred	NA	0.900568	0.943296	0.796976
gin infomax	NA	0.955976	0.865783	0.706608
gin masking	NA	0.892656	0.9358	0.780151
Weave	Canonical	0.951731	0.916734	0.750294
	Attentive FP	0.952443	0.959692	0.756501
CIGIN	Canonical	NA	0.966266	NA

Table 5.1: Graph Based Deep Learning PCC Table

Descriptor Based Machine learning PCC Table

Models	Dataset	PCC Values
SVR	FS 1D 2D 3D	0.9492828
RFR	FS 1D 2D 3D	0.6100712
LR	FS 1D 2D 3D	0.4123368
SVR	FS 2D 3D	0.580132
RFR	**FS 2D 3D**	**0.959431**
LR	FS 2D 3D	0.4123186
SVR	MNSol	0.8581934
RFR	**MNSol**	**0.9960639**
LR	MNSol	0.9229986

Table 5.2: Descriptor Based Machine learning PCC Table

Graph Based Deep Learning RMSE Table

Models	Featurizer	RMSE Values		
		Train	Test	Val
MPNN	Canonical	1.811062	3.8793	3.232
	Attentive FP	0.904992	2.6746	2.2459
Attentive FP	Canonical	1.238914	3.1007	2.4649
	Attentive FP	1.04045	2.0638	2.1904
GAT	Canonical	1.456715	4.4979	2.7233
	Attentive FP	1.31174	2.1126	2.7497
GCN	Canonical	1.019357	4.4218	2.6907
	Attentive FP	1.181536	3.6775	2.9634
gin edge pred	NA	3.198042	3.1159	2.803
gin context pred	NA	2.000627	2.133	3.3994
gin infomax	NA	0.961965	3.8109	2.7832
gin masking	NA	1.542382	2.2649	3.4064
Weave	Canonical	0.2881	2.4436	3.2298
	Attentive FP	**1.000279**	**1.7332**	**3.3879**
CIGIN	**Canonical**	**NA**	**0.992913**	**NA**

Table 5.3: Graph Based Deep Learning RMSE Table

Descriptor Based Machine learning RMSE Table

Models	Dataset	RMSE Values
SVR	FS 1D 2D 3D	1.305640705
RFR	FS 1D 2D 3D	7.459777091
LR	FS 1D 2D 3D	3.798916512
SVR	FS 2D 3D	7.831478405
RFR	**FS 2D 3D**	**1.153717934**
LR	FS 2D 3D	3.798940992
SVR	MNSol	15.17085731
RFR	**MNSol**	**1.881867836**
LR	MNSol	8.166756787

Table 5.4: Descriptor Based Machine learning RMSE Table

CHAPTER 6

Summary

1. Weave Model works best in MoleculeNet as it does not only keep the atom's information but also stores its bonded atoms' information as vectors as well.
2. CIGIN works better than all the models in MoleculeNet as MoleculeNet models only convert Solute SMILES to graph features, whereas CIGIN also featurizes solvents and solutes.
3. ML models work better than graph models as there are 1000+ descriptors to train against solvation energy.
4. The MNSolv dataset provides better results than Freesolv as it is only trained on experimented 25 descriptors and has 3038 solute-solvent combinations. This also shows that 2D descriptors work best with solvation energy prediction.
5. I also ran Quantum multiple regression using Quantum Estimator regressor, VQR, QSVR, and hybrid libraries from qiskit. Multiple linear regression is the only option available for multivariate regression as Quantum ML requires one qubit for each variable, and calculating multiple qubits for algorithms like Random Forest Regressor is not possible with Quantum resources available till now however, this model failed for predicting solvation energy against MNSolv descriptors with RMSE 23.84 and PCC 0.04
The quantum-classical hybrid deep learning model was also trained and tested on 1D,2D and 3D descriptors. However, this model is time-consuming as it also works on a single qubit per variable and hence a single variable at a time. Also, the time taken between sharing the classical data to and fro with quantum computers increases overall time complexity by a considerable margin. Training on a single variable takes approximately two hrs and still gives similar results. Hence, for Quantum ML Algorithms to run efficiently on multivariable models first, Quantum resources must be increased so that multivariate quantum algorithms can be run successfully.

CHAPTER 7

Future Scope

Here in this thesis, graph-based deep learning and ml algorithms have been run with high accuracy, but time and accuracy have been found to be low in most algorithms. Quantum multiple linear algorithms have been used here for multivariate analysis and prediction for solvation energy and 2D descriptors. Quantum Multiple linear algorithms are the only multivariate algorithm possible now because of the single qubit requirement. However, this algorithm did not perform comparably to classical algorithms due to embedding problems with regression and a lack of qubits available for computing. Quantum Machine learning algorithms are now on the rise, and with increased qubit computational capacity, more quantum multivariate algorithms will be possible. With current algorithms, quantum graph networks can understand molecules with high accuracy, but only with molecules of a maximum of three atoms. Afterwards, there is an exponential rise in time complexity with each increasing atom. However, with increasing quantum computational resources, Quantum Graph networks can also be possible candidates for understanding molecular representation. Quantum graph networks can also be used to provide quantum mechanical properties for more giant molecules for training against solvation energy. GPU-based algorithms have performed with comparable speeds with CPU-based algorithms in this thesis as the data transfer time between GPU, and CPU overcomes any time benefit provided by GPU computing. From future algorithms, it can be expected with faster memory transfer speeds, GPU will definitely provide faster computing power for training and predicting purposes in virtual screening. Hence better results can be expected from future algorithms for in silico methods used in virtual screening with an increase in GPU-CPU data transfer speeds and quantum computing resources.

CHAPTER 8

BIBLIOGRAPHY

[1] "The drug development process". US Food and Drug Administration. 4 January 2018. Retrieved 18 December 2019.
[2] "The drug development process: Step 1: Discovery and development". US Food and Drug Administration. 4 January 2018. Retrieved 18 December 2019.
[3] Ehrlich S, Göller AH, Grimme S. Towards full Quantum-Mechanics-based Protein-Ligand Binding Affinities. Chemphyschem. 2017 Apr 19;18(8):898-905. doi: 10.1002/cphc.201700082. Epub 2017 Mar 6. PMID: 28133881.
[4] Zgarbova M, et al. (2010). "Large-scale compensation of errors in pairwise-additive empirical force fields: comparison of AMBER intermolecular terms with rigorous DFT-SAPT calculations". Phys. Chem. Chem. Phys. 12 (35): 10476–10493. Bibcode:2010PCCP...1210476Z. doi:10.1039/C002656E. PMID 20603660.
[5]Crampon K, Giorkallos A, Deldossi M, Baud S, Steffenel LA. Machine-learning methods for ligand-protein molecular docking. Drug Discov Today. 2022 Jan;27(1):151-164. doi: 10.1016/j.drudis.2021.09.007. Epub 2021 Sep 21. PMID: 34560276.
[6]M. Andreev; J. de Pable; A. Chremos; J. F. Douglas (2018). "Influence of Ion Solvation on the Properties of Electrolyte Solutions". J. Phys. Chem. B. 122 (14): 4029–4034. doi:10.1021/acs.jpcb.8b00518. PMID 29611710.
[7] Kaycee Low, Michelle L. Coote, and Ekaterina I. Izgorodina Journal of Chemical Information and Modeling 2022 62 (22), 5457-5470 DOI: 10.1021/acs.jcim.2c01013
[8]Weiner P, Kollman P. J Comput Chem. 1981;2:287.
[9]J. Lee, M. Hitzenberger, M. Rieger, N.R. Kern, M. Zacharias, and W. Im (2020)CHARMM-GUI supports the Amber force fields. J. Chem. Phys. 153:035103
[10]Mobley DL, Guthrie JP. FreeSolv: a database of experimental and calculated hydration free energies, with input files. J Comput Aided Mol Des. 2014 Jul;28(7):711-20. doi: 10.1007/s10822-014-9747-x. Epub 2014 Jun 14. PMID: 24928188; PMCID: PMC4113415.
[11] Marenich, A. V.; Kelly, C. P.; Thompson, J. D.; Hawkins, G. D.; Chambers, C. C.; Giesen, D. J.; Winget, P.; Cramer, C. J.; Truhlar, D. G. Minnesota Solvation Database – version 2012, University of Minnesota, Minneapolis, 2012.
[12]RDKit: Open-source cheminformatics. https://www.rdkit.org
[13]Gilmer J, Schoenholz SS, Riley PF, et al (2017) Neural Message Passing for Quantum Chemistry. arXiv:170401212 [cs]

[14]Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving Beyond Fingerprints. J. Comput.-Aided Mol. Des. 2016, 30, 595– 608, DOI: 10.1007/s10822-016-9938-8

[15]Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. Advances in Neural Information Processing Systems 2015, 2224– 2232

[16]arXiv:1710.10903v3

[17]Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng Journal of Medicinal Chemistry 2020 63 (16), 8749-8760 DOI: 10.1021/acs.jmedchem.9b00959

[18]Mufei Li, Jinjing Zhou, Jiajing Hu, Wenxuan Fan, Yangkang Zhang, Yaxin Gu, and George Karypis ACS Omega 2021 6 (41), 27233-27238 DOI: 10.1021/acsomega.1c04017

[19]Pathak,Y., Laghuvarapu,S.,Mehta,S. I& Priyakumar,U.D.(2020). Chemically Interpretable Graph Interaction Network for Prediction of Pharmacokinetic Properties of Drug-Like Molecules. Proceedings of the AAAI Conference on Artificial Intelligence, 34(01), 873-880. https://doi.org/10.1609/aaai.v34i01.5433

[20]Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, I& Zhenqin Wu (2019). Deep Learning for the Life Sciences. O'Reilly Media.

[21]Drucker Harris, Burges Christopher J. C., Kaufman Linda, Smola Alex J., and Vapnik Vladimir. 1997. Support Vector Regression Machines. In Advances in Neural Information Processing Systems. MIT Press, 155–161

[22]Liaw A, Wiener M (2002). "Classification and Regression by randomForest." R News, 2(3), 18-22. https://CRAN.R-project.org/doc/Rnews/.

[23]David A. Freedman (2009). Statistical Models: Theory and Practice. Cambridge University Press. p. 26. A simple regression equation has on the right hand side an intercept and an explanatory variable with a slope coefficient. A multiple regression e right hand side, each with its own slope coefficient [24]Vinyals,Bengio, and Kudlur 2015