International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Graph-Based Approach for Solubility Prediction of Drugs using SMILES Data

Kalidindi Venkateswara Rao[1], Dr. Kunjam Nageswara Rao[2], Dr. G. Sitaratnam[3], Kandula Vamsi[4]

Submitted:13/03/2024 **Revised**: 28/04/2024 **Accepted**: 05/05/2024

Abstract: Graph Neural Networks (GNN) utilization in the case of molecular property prediction is considered a significant advancement in computational chemistry and drug discovery. Initial approaches to molecular property prediction especially solubility prediction depend on empirical rules or physicochemical descriptors, which lack generalization and predictive accuracy. The proposed model Graph Convolutional Network (GCN) which is a variant of GNN learns representations of molecular graphs, enabling accurate prediction of molecular properties directly from raw molecular structures. The molecular graphs are created from the Simplified Molecular Input Line Entry System (SMILES) data which are molecular sequences of drug target compounds. In the proposed work, GCN uses graph pooling, which effectively reduces the node dimensionality. This work shows how the whole graph can be considered as input and how different pooling techniques can be used to handle large and complex graph data and also the effectiveness of GCN for solubility prediction. The proposed GCN model is hyperparameter tuned by using Grid Hyperparameter optimization on ESoL dataset which is a regressive type dataset achieving a low RMSE value of 0.43 outperforming machine learning and many deep learning models.

Keywords: *GCN, GNN, Grid Hyperparameter Optimization, RMSE, SMILES*

1. Introduction

Drug discovery is the process of finding and creating new medications aimed at treating diseases and enhancing human health. This process encompasses multiple stages, such as identifying and validating targets, discovering lead compounds, conducting preclinical tests, carrying out clinical trials, and obtaining regulatory approval. Target Identification is the first step in drug discovery. It is the process of identifying potential targets that play a key role in the disease process. These targets are often identified through a variety of methods, including genetic studies, molecular biology techniques, analysis of disease mechanisms, and bioinformatics.

SMILES (Simplified Molecular Input Line Entry System) is a compact and human-readable notation for representing chemical structures using ASCII strings. It provides a concise and standardized way to encode molecular structures, facilitating data exchange and manipulation in cheminformatics. SMILES not only represents the connectivity of atoms in a molecule but also captures stereochemical and isotopic information. This representation is extensively used in various fields such as drug discovery, chemical database management, and computational chemistry due to its simplicity and versatility. In QSAR (Quantitative Structure-Activity Relationship) modelling, SMILES representations play a crucial role in encoding molecular structures for predictive modelling. QSAR approaches utilizing SMILES typically involve converting molecular structures into numerical descriptors or fingerprints derived from the SMILES strings, which are then used as input features for machine learning models to predict biological activities or properties of interest, such as drug potency or toxicity. By leveraging SMILES representations, QSAR enables the development of predictive models that correlate molecular structure with biological activity, facilitating the design and optimization of new chemical compounds with desired properties.

SMILES representations can be computed using two primary approaches: sequence-based and graph-based methods. Sequence-based approaches treat SMILES strings as linear sequences of characters, where each character represents an atom, bond, or special symbol (e.g., branching or aromaticity). These approaches involve parsing the SMILES strings character by character and converting them into numerical representations, such as one-hot encoding or embedding vectors, which can be used as input features for machine learning models. While sequence-based methods are straightforward and easy to implement, they may not fully capture the structural relationships and spatial arrangements of atoms within molecules. graph-based approaches represent molecules as graphs, where atoms are

[1] *Research Scholar, Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India.*
ORCID ID: 0009-0008-4240-4237
[2] *Professor, Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India.*
ORCID ID: 0009-0005-2779-0238
[3] *Professor, Department of Computer Science and Engineering, Chaitanya Engineering College, Visakhapatnam, India.*
ORCID ID: 0009-0006-7618-7045
[4] *Graduate, Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India.*
ORCID ID: 0009-0002-1791-5285

** Corresponding Author Email: kvrao.rs@andhrauniversity.edu.in*

represented as nodes and bonds as edges connecting the nodes. In this representation, the molecular structure is captured in terms of its topology, bond connectivity, and atom properties. Graph-based methods leverage graph neural networks (GNNs) or other graph-based models to directly operate on the molecular graph data, enabling the models to learn and exploit the spatial relationships and structural features of molecules more effectively. By considering the entire molecular graph, graph-based approaches can capture complex interactions and dependencies between atoms and bonds, leading to more accurate and informative representations for machine learning tasks.

Molecular fingerprints are representations of molecular structures used in cheminformatics and computational chemistry. They encode structural information about molecules into a fixed-length binary or numerical vector, which can be used for tasks such as similarity search, virtual screening, and quantitative structure-activity relationship (QSAR) modelling. Molecular descriptors, which capture different aspects of molecules, are categorized into one-dimensional (1-D), two-dimensional (2-D), and three-dimensional (3-D) descriptors. Structural keys encode molecule structures into binary strings based on predefined structural features like substructures or fragments, exemplified by MACCS keys and PubChem fingerprints. Path-based fingerprints follow linear paths within molecules, with examples like Daylight fingerprints, while circular fingerprints consider circular atom environments up to a specified radius or diameter. These approaches offer versatile means of representing molecular structures, crucial for diverse cheminformatics tasks and machine learning applications.

Understanding the solubility of drug targets is paramount in drug discovery and development processes. Solubility refers to the ability of a substance to dissolve in a given solvent, typically water in pharmaceutical contexts. In the context of drug targets, solubility data provide crucial insights into the potential bioavailability and pharmacokinetic behaviour of candidate compounds. Poorly soluble compounds may face challenges in formulation development, leading to issues such as low bioavailability and inconsistent drug delivery. Hence, predicting the solubility of target compounds early in the drug discovery process is essential for selecting promising candidates for further development. ESOL (Estimated SOLubility) predicted log solubility in mols per litre is a widely used tool for estimating the solubility of organic compounds in water. It provides a quantitative measure of solubility, aiding researchers in prioritizing compounds with favourable solubility profiles for further experimentation and optimization. Integrating solubility predictions into drug design workflows allows researchers to expedite the drug discovery process, thereby speeding up the development of safe and effective therapies.

2. Literature Survey

"Semi-Supervised Classification with Graph Convolutional Networks"[1] by Thomas N. Kipf and Max Welling (2017): This seminal paper introduces graph convolutional networks (GCNs), which extend the convolutional operation to graph-structured data. They propose a spectral-based approach and demonstrate its effectiveness on semi-supervised node classification tasks.

"GraphSAGE: Inductive Representation Learning on Large Graphs"[2] by William L. Hamilton et al. (2017): This work introduces GraphSAGE, a framework for inductive representation learning on large graphs. GraphSAGE performs neighbourhood aggregation to generate node embeddings, enabling scalable and efficient learning on graphs of varying sizes.

"Graph Attention Networks"[3] by Petar Velickovic et al. (2018): This paper presents graph attention networks (GATs), which leverage self-attention mechanisms to weigh the importance of neighbour nodes during message passing. Graph Attention Networks (GATs) achieve cutting-edge performance on a variety of graph-related tasks, such as node classification and link prediction.

"How Powerful Are Graph Neural Networks?"[4] by Keyulu Xu et al. (2019): This work investigates the expressive power of graph neural networks (GNNs), including GCNs, in terms of their ability to approximate graph functions. They analyze the limitations of existing GNN architectures and propose techniques to enhance their expressive capacity.

"Hierarchical Graph Representation Learning with Differentiable Pooling" [5] by Rex Ying et al. (2018): This paper introduces differentiable pooling techniques for hierarchical graph representation learning. They propose a framework that learns to coarsen graphs hierarchically while preserving important structural information, enabling scalable graph classification.

"DiffPool: Graph Pooling via Learning Differentiable Graph Structures" [6] by Ying et al. (2018): This work presents DiffPool, a differentiable pooling mechanism that learns to pool nodes based on their representations and the underlying graph structure. DiffPool enables end-to-end learning of graph representations and outperforms traditional pooling methods on graph classification tasks.

"Understanding Graph Convolutional Networks for Node Classification"[7] by Jie Zhou et al. (2018): This paper provides a comprehensive analysis of graph convolutional networks (GCNs) for node classification tasks. It investigates the behaviour of different GCN architectures and explores the impact of various factors, such as graph structure and initialization schemes, on their performance.

"Deep Graph Convolutional Encoder-Decoder Networks for

Representation Learning of Chemical Molecules"[8] by Y. Li et al. (2018): This study focuses on applying graph convolutional networks (GCNs) to the task of molecular representation learning. They propose a deep graph convolutional encoder-decoder network for generating meaningful molecular embeddings, which can be used for molecular property prediction and chemical reaction prediction.

"Mean Field Graph Convolutional Neural Networks"[9] by Stefan W. R. Selsam et al. (2019): This paper introduces mean field graph convolutional neural networks (MFGCNs), which extend traditional GCNs by incorporating mean field theory from statistical physics. MFGCNs capture long-range dependencies in graphs more effectively and achieve competitive performance on various graph-based tasks.

"Graph U-Nets"[10] by Hongyang Gao et al. (2019): This work proposes Graph U-Nets, a novel architecture that combines the strengths of graph convolutional networks (GCNs) and traditional U-Net architectures for various graph-level tasks such as graph classification and segmentation. Graph U-Nets leverage skip connections and hierarchical pooling to capture both local and global graph features effectively.

"Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks"[11] by Bastian Rieck, Christian Bock, and Heiko Strathmann (2020): This paper introduces higher-order graph neural networks (HOGNs), which extend traditional graph neural networks (GNNs) to capture higher-order interactions between nodes in a graph. The motivation behind HOGNs is inspired by the Weisfeiler-Lehman (WL) graph isomorphism test, a powerful graph theoretic method for distinguishing non-isomorphic graphs. HOGNs, particularly the proposed Weisfeiler-Lehman Neural Networks (WLNNs), offer efficient and scalable learning of higher-order graph representations, as demonstrated through empirical evaluations showcasing their effectiveness on various graph-related tasks.

“TRANSFORMER-CNN: FAST AND RELIABLE TOOL FOR QSAR” [12] by Pavel Karpov et al. (2020): This work demonstrates how a Convolutional Neural Network (CNN) can be constructed using transformers, employing a 10-block self-attention mechanism in the encoders, and evaluates the reliability of transformers in the QSAR approach.

“malC: A novel deep learning architecture for malware classification” [13] by V Harinadh et al. (2024): This work shows that accuracy increases for unbalanced datasets by using the deep neural network.

3. About Dataset

The Delaney dataset, also known as the "Delaney's solubility dataset," is a widely used benchmark dataset in cheminformatics and computational chemistry. It was compiled by John Delaney and originally published in 2004. The dataset consists of experimentally measured aqueous solubility data for a diverse set of chemical compounds. Aqueous solubility refers to the ability of a compound to dissolve in water. The dataset includes molecular structures represented in SMILES notation along with corresponding experimental solubility values. It has been extensively utilized for the development and validation of quantitative structure-activity relationship (QSAR) models and other predictive models in the field of computational chemistry and cheminformatics. The dataset consists of 10 attributes (columns) and multiple instances (rows), where each row represents a unique chemical compound. The attributes include both numerical and categorical data related to molecular properties and solubility. Feature extraction involves transforming raw data into a format that is suitable for modelling. In this dataset, features can be extracted from attributes like molecular weight, the number of hydrogen bond donors, the number of rings, and more. These features are useful for constructing predictive models for solubility.

4. Methodology

Graph Neural Networks (GNNs) are a type of neural network model specifically designed to work with graph-structured data. Unlike traditional neural networks, which are intended for fixed-dimensional data such as images or sequences, GNNs can process data with arbitrary graph structures. This capability makes them ideal for tasks that involve relational data or data with intricate dependencies. In molecular datasets, where molecules are naturally represented as graphs with atoms mapped to nodes and bonds as edges, GNNs are used for various cheminformatics tasks. By leveraging the structural information encoded in molecular graphs, GNNs can effectively capture complex interactions between atoms and bonds, enabling tasks such as molecular property prediction, molecular similarity assessment, and molecular generation. GNN-based models can learn meaningful representations of molecules directly from their graph structures, allowing for more accurate and interpretable predictions compared to traditional methods. Additionally, GNNs can be combined with other deep learning techniques, such as attention mechanisms and reinforcement learning, to further enhance their performance and versatility in analyzing molecular datasets.

In order to model convolutional network on the graphs it requires a convolutional layer followed by graph pooling layer. At the step of convolutional layer among different approaches Graph Convolutional ha been used. And for the next layer graph pooling, which is a crucial component in graph neural networks (GNNs) for effectively aggregating information from multiple nodes in a graph while reducing dimensionality. It plays a vital role in enhancing the

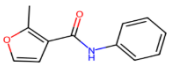
discriminative power of GNNs by capturing hierarchical features and global structural patterns in the graph. Various pooling techniques have been developed to address different aspects of graph data. Top-K pooling selects the top K nodes based on certain criteria, prioritizing the most informative nodes in the graph. SAGPooling (Self-Attention Graph Pooling) employs self-attention mechanisms to dynamically assign importance scores to nodes and aggregates information accordingly, enabling adaptive pooling based on node relevance. EdgePooling focuses on preserving important structural information by selectively removing less relevant edges, thereby reducing the graph size while maintaining its connectivity. MaxPooling aggregates information from neighboring nodes by taking the maximum value over node features, providing a simple yet effective way to capture local features in the graph. These pooling techniques offer diverse strategies for summarizing graph data, catering to different tasks and modeling requirements in molecular datasets.

In the context of ESOL dataset GraphConv paired with Maxpooling yields the lowest Root Mean Square Error(RMSE) score compared to others ,This outcome is due to Maxpooling simplicity and efficiency which makes it less prone to overfitting and computational complexity issues compared to other pooling techniques. By prioritizing the extraction of local features and maintaining computational efficiency, MaxPooling in conjunction with the GraphConv layer can provide a robust and effective framework for modelling molecular datasets like ESOL, ultimately resulting in superior predictive performance as indicated by the lower RMSE score. Different approaches have been used by using the transformer convolutional and graph convolutional at convolutional layer point and also different pooling such as Topk pooling and Graph pooling has been used.

4.1 Architecture

The ESOL dataset contains SMILES representation of the compound and the ESOL predicted log solubility in mols per litre. We use the SMILES representation of the compound and predict the ESOL log solubility using our Model architecture.

Cc1occc1C(=O)Nc2ccccc2



Fenfuram

After converting the dataset to molecular objects, the next step involved extracting molecular descriptors using the DeepChem library. Specifically, the ConvMolFeaturizer from DeepChem was employed to generate default node and edge representations for each molecule. The node representation was constructed by concatenating various features, resulting in a feature length of 30. These features included atom type, formal charge, hybridization, hydrogen bonding properties, aromaticity, degree, number of hydrogens, and chirality (optional). Additionally, partial charge (optional) was also considered. Similarly, the edge representation was constructed by concatenating features such as bond type, same ring indicator, conjugation status, and stereo configuration, resulting in a feature length of 11. This comprehensive feature extraction process ensured that each molecule was represented by a rich set of descriptors capturing its structural and chemical properties, thereby enabling downstream modelling tasks such as predictive modelling and property prediction in cheminformatics.

SMILES representations converted into molecular graphs and

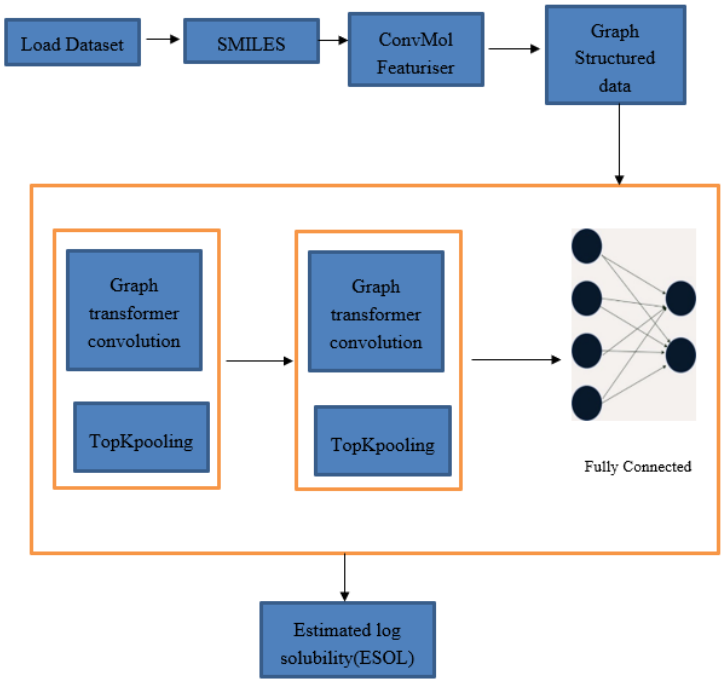


Fig.1 Architecture of GCN with Transformer Convolution and TopK Pooling.

enriched with molecular descriptors, the dataset is primed for utilization in deep learning models. Based on the convolutional layer and pooling layer two different architectures have been proposed which are evaluated based on RMSE value.

4.1.1 TransformerConv

The TransformerConv integrates the self-attention mechanism from transformers with traditional graph convolutional operations. This layer is designed to dynamically weigh the importance of neighboring nodes for each node in the graph, allowing it to capture both local and global contextual information effectively. By leveraging self-attention, TransformerConv can learn to prioritize the most relevant features and relationships within the molecular graph, enhancing the model's ability to understand complex molecular structures. This makes it particularly suitable for tasks that require interpretation of the spatial and chemical properties inherent in molecular data.

We employ Transformer Convolutional (TransformerConv) layers to process the input SMILES strings as shown in Fig. 1. TransformerConv layers are adapted to learning intricate patterns by combining the strengths of convolutional neural networks and transformer models. These layers enable the model to focus on relevant features of the molecular structure, enhancing its ability to understand spatial and contextual information.

To further refine the features extracted by the TransformerConv layers, we utilize Top-k Pooling twice. The TopKPooling layer is a graph pooling technique used to reduce the size of the graph while preserving its most informative parts. It works by selecting the top k of nodes based on a learnable scoring function, which determines the importance of each node. By retaining only, the most significant nodes, TopKPooling effectively compresses the graph, focusing the model's attention on the most crucial elements of the molecular structure. This selective pooling not only reduces computational complexity but also enhances the model's ability to generalize by removing less relevant information. Applied after convolutional layers, TopKPooling ensures that the model retains critical structural and functional details, contributing to more accurate predictions of molecular properties such as log solubility.

Following the pooling stages, the reduced and refined feature set is passed to a fully connected neural network. This network serves as the final prediction layer, where the intricate patterns and relationships identified by the previous layers are synthesized to produce the estimated log solubility. By employing a series of dense layers, the model learns to map the abstracted features to the target solubility value accurately.

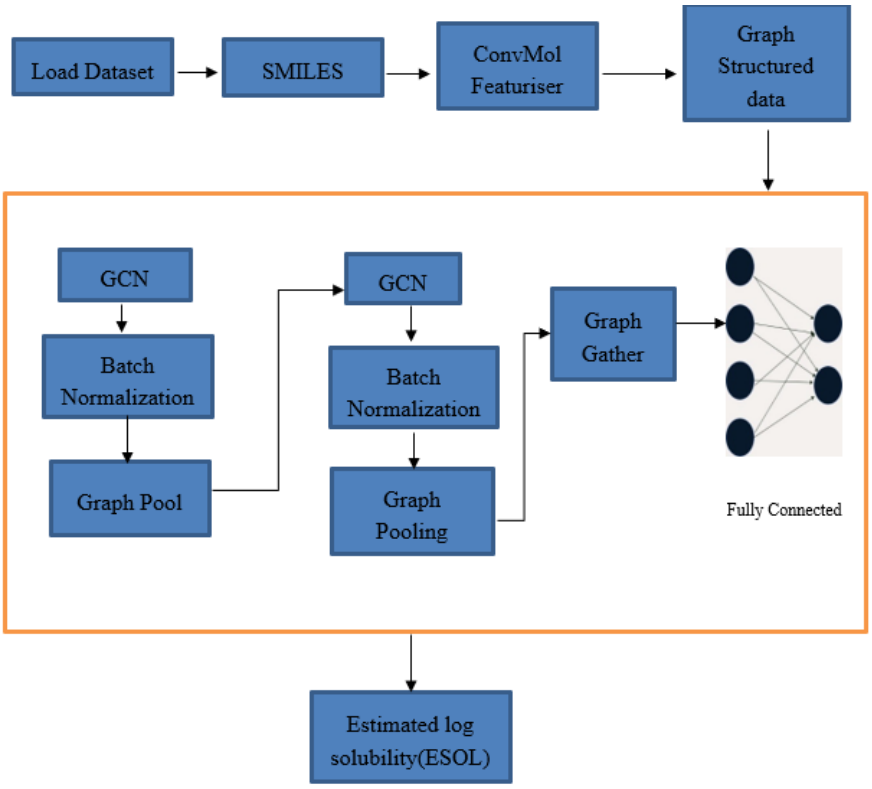


Fig2. The Architecture of GCN with GraphConvolutional and GraphPooling.

We can see that this architecture performs better than the pre-trained models such as D-MPNN The Directed Message Passing Neural Network (D-MPNN) enhances molecular property prediction by utilizing directed edges to capture the directionality

of molecular bonds, providing a more accurate representation of chemical interactions. N-Gram models, commonly used in natural

language processing, predict the next item in a sequence based on

the previous n items. PreTrainGNN refers to the pre-training of graph neural networks on large datasets before fine-tuning them on specific tasks, improving their performance by leveraging learned representations of molecular graphs. GROVER (Graph Representation frOm self-superVised mEssage passing tRansformer) is a self-supervised learning framework for molecular graphs that combines message-passing neural networks with transformer architectures. GraphMVP (Graph-based Multi-View Prediction) employs multiple views of molecular graphs to capture different structural and chemical properties, improving prediction accuracy. MolCLR (Molecular Contrastive Learning of Representations) uses contrastive learning to pre-train graph neural networks on molecular data, distinguishing between similar and dissimilar molecular structures.

As shown in the below Table. 1 GCN with Transformer Conv and TopK pooling has achieved the lowest RMSE value than previous models after the hyperparameter tuned by using Optuna framework.

Table. 1: Comparison of RMSE values for various GNN's

Model	ESOL(RMSE)
D-MPNN	1.050
N-Gram	1.083
PreTrainGNN	1.100
GROVER	0.983
GraphMVP	1.029
MolCLR	1.271
TransformerConv+TopkPooling	**0.944**

4.1.2 GraphConv

Graph convolutional neural networks are particularly well-suited for leveraging the structural information encoded in molecular graphs. These models operate directly on the graph topology, utilizing both node features (representing atoms) and edge features (representing bonds) to convolve and create node embeddings for the subsequent layers. In the Fig.2 proposed architecture of GCN with all the layers is shown. The GraphConv layer is utilized to gather insights from neighbouring atoms, integrating them into the node attributes. This involves leveraging both node and edge attributes to understand the intricate structural connections within the graph.

$$x_\iota = W_1 x_i + W_2 \sum_{j \in N(i)} e_{j,i}.x_j$$

Where xi represents node features and eij represents edge features if edge between i and j nodes.

A pooling layer is utilized to down-sample the graph representation, preserving essential features while reducing computations and enhancing efficiency. A GraphPool layer aggregates data from the local neighbourhoods of a graph, performing max pooling on the feature vectors of atoms within these neighbourhoods. Many Graph Convolutional networks process feature vectors for each graph node. In the context of a molecule, each node may represent an atom, and the network manipulates atomic feature vectors that capture the local chemical environment of the atom.

To represent the entire molecule as a single vector, we use GraphGather, which pools the node-level feature vectors to generate a graph-level feature vector.

We utilize the GraphConv, GraphPool, and GraphGather layers to achieve a vector representation of the molecule. By performing two down-sampling operations on the graph and applying the GraphGather layer, we create and aggregate representations of the graphs, resulting in a comprehensive vector representation of the entire molecule. This vector is then fed into a multilayer perceptron for the prediction task. The MLP typically consists of one or more hidden layers, each comprising multiple neurons. These hidden layers enable the model to learn complex nonlinear relationships between the input features and the target labels. The number of hidden layers and neurons per layer is chosen based on the complexity of the regression task and the available computational resources. Each neuron in the hidden layers of the MLP applies an activation function to its input to introduce nonlinearity into the model. We use the ReLU activation function in our Model .

The training of a Multilayer Perceptron (MLP) involves using an appropriate loss function to measure the difference between predicted class probabilities and the true labels. For binary classification, binary cross-entropy loss is typically used, while categorical cross-entropy loss is used for multi-class classification. This loss function guides the optimization process during training. In our case, we employ L1 Loss, which is commonly used in classification tasks. To improve the training efficiency by dynamically adjusting the learning rates and incorporating bias correction Adam optimisation technique is used.

5. Results & Analysis

This study analyzes the performance of various state-of-the-art machine learning (ML) and deep learning (DL) techniques in predicting aqueous solubility. The proposed model is compared with several ML and DL algorithms that rely on molecular graph

structures, using a dataset of over 1,128 compounds. Throughout this study, we calculated and evaluated the Root Mean Squared

Error (RMSE) of different existing ML and DL models against the

proposed Graph Convolutional Neural Network model. From the Table. 2 RMSE value of 0.43 for GCN model which is less RMSE value when compared to the existing ML and DL models. Some of the machine learning algorithms which included are MLR,1D-CNN and among the machine learning models we observed Random Forest has low RMSE value and among Deep learning algorithms included GAT, AGNN, SGC, GCN from which GCN has the least RSME value.

Table. 2: Comparing ML and Dl models to GCN

Algorithms	RMSE
1D-CNN	0.971
TransformerConv+TopkPooling	**0.944**
MLR	0.82
GAT	0.71
SGC	0.87
AGNN	0.76
RF	0.64
GCN	**0.43**

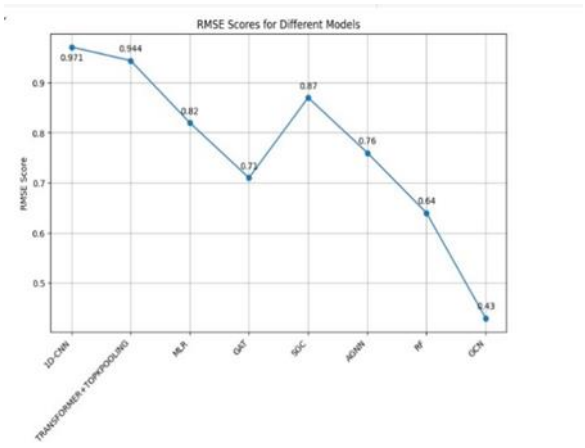


Fig. 3: Comparison graph of existing models and proposed GCN's.

The proposed GCN with Transformer Convolutional and TopK pooling has been examined on FreeSolv dataset which is also a regression dataset for solubility prediction. The model acheived RMSE value of 1.52 on a few trials. GCN with GraphConvolutional and GraphPooling achieved RMSE value of 2.34. On analysis, it was found that GCN with GraphConvolutional and GraphPooling works efficiently if there are efficient training samples, whereas

GCN with TransformerConvolutional and TopKpooling shows a good performance irrespective of the number of training samples.

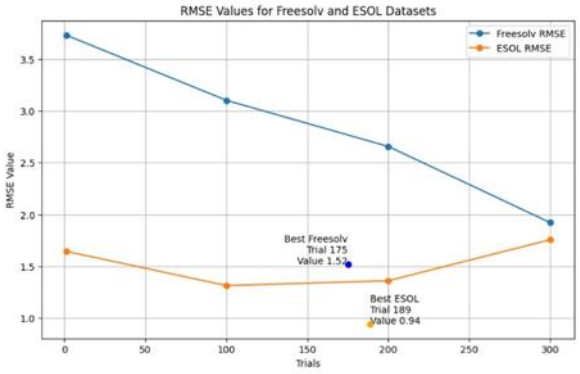


Fig. 4 RMSE value comparison for FreeSolv and Esol datasets by using GCN with Transformer Convolutional and TopK pooling.

At trail 175 and 189 the proposed GCN has achieved the lowest RMSE value for FreeSolv dataset and Esol dataset respectievely.

6. CONCLUSION

This work aims to investigate the problem of predicting aqueous solubility of drug compounds. To achieve this, Graph Convolutional Networks (GCNs) were chosen and implemented using the Quantitative Structure-Activity Relationship (QSAR) approach. The implemented GCN achieved an RMSE value of 0.43, which is lower compared to existing graph neural network models. Here we have used the grid-based hyperparameter to fine-tune the model. Grid Search evaluates different combinations of specified hyperparameters and their values, calculates the performance for each combination, and selects the optimal hyperparameters. Hence using the grid-based hyperparameter tuning with the QSAR approach made the GCN model achieve a RMSE value of 0.43 which is less when compared to some other graph neural networks. Depending upon the variation in datasets sizes among the two different variants of GCN, the GCN with Transformer Convolutional and TopK pooling works efficiently. Many new technologies are coming up to solve the aqueous solubility prediction problem. Where, Sequence-based learning and Natural Language Processing techniques can also be used to check the drugs' solubility.

REFERENCES

[1] Kipf, Thomas N., and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks." *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. arXiv:1609.02907.

[2] Hamilton, William L., Rex Ying, and Jure Leskovec. "GraphSAGE: Inductive Representation Learning on Large Graphs." *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017. arXiv:1706.02216.

[3] Veličković, Petar, et al. "Graph Attention Networks." *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. arXiv:1710.10903.

[4] Xu, Keyulu, et al. "How Powerful Are Graph Neural Networks?" *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. arXiv:1810.00826.

[5] Ying, Rex, et al. "Hierarchical Graph Representation Learning with Differentiable Pooling." *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018. arXiv:1806.08804.

[6] Ying, Rex, et al. "DiffPool: Graph Pooling via Learning Differentiable Graph Structures." *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018. arXiv:1806.08804.

[7] Zhou, Jie, et al. "Understanding Graph Convolutional Networks for Node Classification." 2018. arXiv:1806.09978.

[8] Li, Y., Zhang, L., and Liu, Z. "Deep Graph Convolutional Encoder-Decoder Networks for Representation Learning of Chemical Molecules." 2018. arXiv:1711.07553.

[9] Selsam, Stefan W. R., Nikolaj Bjørner, and William Cook. "Mean Field Graph Convolutional Neural Networks." 2019. arXiv:1810.05997.

[10] Gao, Hongyang, and Shuiwang Ji. "Graph U-Nets." *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. arXiv:1905.05178.

[11] Rieck, Bastian, Christian Bock, and Heiko Strathmann. "Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks." *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, 2020. arXiv:1906.01277.

[12] Karpov, Pavel, Grigory Godin, and Igor V. Tetko. "TRANSFORMER-CNN: Fast and Reliable Tool for QSAR." 2020. arXiv:2002.06038.

[13] Varikuti, H., Vatsavayi, V. (2024). malC: A Novel Deep Learning Architecture for Malware Classification. In: Roy, N.R., Tanwar, S., Batra, U. (eds) Cyber Security and Digital Forensics. REDCYSEC 2023. Lecture Notes in Networks and Systems, vol 896. Springer, Singapore, pp 435-448 .