GNN Model for Classification of SARS-CoV-2 severity in Molecules

by

H M LAYES DELOWER
18201059
KHANDAKAR MAISHA TANZIM
23141070
FAISAL SHAHRIAR
20301020
SHARIKA FAIROOZ
21101110

A thesis submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
May, 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

H M LAYES DELOWER
18201059

KHANDAKAR MAISHA TANZIM
23141070

FAISAL SHAHRIAR
20301020

SHARIKA FAIROOZ
21101110

Approval

The thesis/project titled "GNN Model for Classification of SARS-CoV-2 severity in Molecules" submitted by

1. H M LAYES DELOWER (18201059)
2. KHANDAKAR MAISHA TANZIM (23141070)
3. FAISAL SHAHRIAR (20301020)
4. SHARIKA FAIROOZ (21101110)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May, 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
BRAC University

Co - Supervisor:
(Member)

Mohammad Sayeem Sadat Hossain
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

In a time when healthcare issues and diseases get more complex every day, it becomes evident that efficient and precise disease detection and classification are invaluable. Through the quick development of machine learning methods and artificial intelligence, there are now new and revolutionary techniques for disease detection and diagnosis. Conventional methods of detecting a disease may oversimplify this complex relationship of dependence and reflection inside a bundle of dataset comprising extremely heterogeneous symptoms and pathologies. Therefore, conventional methods may fail to provide enough feedback and inputs to the medical unit. The main topic of this thesis is the usage of Graph Neural Networks (GNNs) to spot and diagnose diseases. Particularly, this analysis focuses on the ability of GNNs to assess COVID-19 severity based on the SMILES dataset. Particularly, this analysis focuses on the ability of GNNs to assess COVID-19 severity based on the SMILES dataset. This study proves that by exploiting the capacity of GNNs, GNNs can deliver the precision required for prompt interventions, and this results in improved patients' outcomes and an effective healthcare system. The experimental results are highly promising, with GNNs achieving an accuracy of 87.16%, an F1 score of 82.63%, a precision of 84.27%, and a recall of 81.06% for Version 1 (not considering inactive cases), and an accuracy of 69.52%, an F1 score of 71.28%, a precision of 65.42%, and a recall of 78.30% for Version 2 (considering all cases — active, intermediate, and inactive). These data show that the GNNs approach is a successful method of classifying the level of severity of COVID-19 correctly by the way they depict the complicated connections of the dataset. This marks an ideal balance between the two metrics of precision and recall, suggesting that the model can correctly identify the cases and also minimize false negatives. This becomes even more important in a healthcare setting where the cost of misdiagnosis is extremely high. The article in general illustrates the capabilities of GNNs in transforming the process of disease diagnosis into a more efficient, effective, and accurate one, which can have a profound meaning for doctors, patients, and other healthcare providers.

Keywords: GNN, SMILES, Graph data, COVID-19.

Table of Contents

List of Figures

Chapter 1

Introduction

The face of health care has been greatly reshaped off late by the fast-changing technology via artificial intelligence (AI) and machine learning (ML). Introduced by these advancements, the era of research and practice has been full of innovative methods for diagnosing, categorizing, and the cure for diseases. Among the techniques that have emerged, graph neural networks (GNNs) are being used because of their capability to capture the network of connections and dependencies within the data. GNNs introduce a way to improve disease detection and diagnosis which is crucial in healthcare where the impact of machine learning can be much felt.

Disease detection is the process of bringing to bear whether a certain medical condition is present or is likely to occur in a patient or a group in health care, and it is one of the major things in medicine. Early intervention and treatment became better because disease identification became fast and accurate which is beneficial to health care resources too. However, traditional disease detection methods now and then experienced difficulty because they could not handle many intricate interconnections seen in medical data. It is really very difficult to create accurate diagnostic models since diseases exhibit themselves with a wide diversity, and patients tend to show different symptoms.

In prior research, GNNs have been employed to predict protein reactivity and identify potential antimalarial drugs by modeling graph-structured data, achieving high performance across various metrics. For instance, in a paper, Gil and Rowley demonstrated GNNs' superiority in distinguishing between covalent and non-covalent inhibitors [1]. While Mswahili et al. showcased GNNs' efficacy in predicting drug-target interactions using a combination of BERT and Relational Graph Convolutional Network (RGCN) [2]. These applications underscore GNNs' potential in handling intricate biomedical data relationships. The objective of this research is to explore how Graph Neural Networks can optimize disease detection and overcome the challenges of traditional approaches. The GNNs are based on a graph theory and they are computationally designed to resemble real relationships and dependencies that occur in graph-represented data. They present the ability to throw light on the complex patterns and connections that perhaps are not visible to the usual techniques in the area of healthcare, in which patients, symptoms and diseases con-

stitute linked nodes in a graph. Inherent network structures in healthcare data and the ability of GNNs to understand as well as utilize these patterns are two different things that this research seeks to bridge. In doing so, this study aims to improve the accuracy, dependability, and efficiency of illness detection processes, which would eventually improve patient care and the efficacy of the healthcare system.

1.1 Problem Statement

The limitations of traditional disease detection methods in healthcare are primarily due to their inability to effectively capture and utilize the interconnections and dependencies found in medical data. Diseases exhibit varying levels of complexity and patients with a range of symptoms making it challenging to establish definitive diagnostic criteria. Additionally both researchers and medical professionals face interpretive obstacles due to the vast volume and variability of medical data.

The main issue at hand is the need for disease detection methods that can enhance efficiency and accuracy by leveraging the inherent network structures within medical data. Medical data is inherently interconnected, resembling a network where patients, ailments, symptoms and diagnostic procedures are interconnected nodes. Traditional diagnostic techniques often struggle to record and analyze these relationships. They may overlook connections and patterns that are crucial for precise disease diagnosis. This limitation stems from the rule based nature of algorithms, which fail to capture the nonlinear networked nature of medical data.

Graph Neural Networks (GNNs) have emerged as a solution to this problem because they can effectively represent relationships and dependencies present in graph structured data. This characteristic uniquely equips them for tasks related to disease detection within the healthcare field.

The potential of GNNs has been highlighted in various studies, including their application in learning molecule representations for tasks like Graph Edit Distance predictions and embedding visualization [3], and predicting antiviral drug interactions with coronaviruses [4]. These examples illustrate GNNs' capability to overcome limitations of traditional fingerprint-based models and handle heterogeneous biomedical data effectively.

This research seeks to address the following key aspects of the problem:

- **How can GNNs be effectively applied to disease detection tasks, specifically in determining the severity of COVID in molecules using the sarscov2 SMILES dataset?**
- **What are the benefits and limitations of using GNNs for disease detection, particularly in the context of COVID severity assessment?**
- **How do GNNs compare to traditional methods in terms of accuracy, interpretability, and efficiency?**

By addressing these questions, this thesis aims to contribute to the development of robust disease detection solutions that can have a significant impact on healthcare outcomes. The successful integration of GNNs promises to enhance diagnostic accuracy, reduce misdiagnosis, enable earlier interventions, and ultimately improve patient outcomes. Moreover, it has the potential to optimize healthcare resource allocation, making healthcare systems more efficient and conceivably, more cost-effective.

1.2 Research Objective

The primary objective of this paper is to investigate and establish the efficacy, accuracy and efficiency of integrating GNN into the process of COVID severity assessment within SMILES molecules. Specifically, the research aims to:

- **Evaluate the Applicability of GNN in Simplified Molecular Input Line Entry System(SMILES) dataset:** Investigate how GNN can effectively capture, identify and link the intricate relationships and dependencies within molecular data, and assess the accuracy of the results.
- **Advance molecular classification Practices:** Contribute insights that can lead to the systematic integration of GNN into various aspects of medical diagnosis and healthcare management, ultimately advancing the quality of healthcare delivery.

By achieving these objectives, this paper seeks to drive the development of GNN as a transformative tool in the field of molecular classifications, with the overall goal to enhance healthcare quality and accessibility, ultimately contributing to the well-being of patients and other healthcare stakeholders.

1.3 Thesis Structure

This paper is organized into several sections, each outlining the steps and methods undertaken to reach our conclusions on the application of Graph Neural Networks (GNNs) for disease detection, particularly for assessing the severity of COVID-19 using the sarscov2 dataset.

Chapter 1 – Introduction: In this chapter, the paper dives into the importance of disease detection in healthcare and how GNNs can boost diagnostic accuracy. It states the details on the Problem Statement (Section 1.1) addressing current method limitations and the need for fresh solutions. As well as, the Research Objective (Section 1.2) lays out the study's goals.

Chapter 2 – Review: Here, the paper dive into research on using machine learning and GNNs in diagnoses. It explores past studies, their methods, and where they might have missed, shaping our thesis.

Chapter 3 – The paper outlines the research plan, including data collection and evaluation details to help reach the goals.

Chapter 4 – Methodology delves into research methods. It includes Data Description, Data Preprocessing, and Model sections.

Chapter 5 – The research results cover accuracy, precision, recall, and F1 scores of the GNN model.

Chapter 6 – Section 6.1 suggests future research directions and ways to improve GNNs for better disease detection. Section 6.2 Summarizes the key findings, discusses using GNNs in COVID severity assessment, and their impact on healthcare. It also points out study limitations and offers recommendations for future research.

Chapter 2

Background and Related works

2.1 Background

Graph Neural Network:
A Graph Neural Network (GNN) is a part of artificial neural networks for processing graph-structured data, consisting of nodes and edges. The main design element of GNNs are pairwise message passing. In this process the nodes iteratively update their representation by exchanging information with their neighbours. The neighbourhood aggregation process allows GNNs to effectively collect local graph structures and features. GNNs are used for many different applications like social network analysis, molecular property prediction and recommendation systems as they are highly flexible and can handle irregular data structures. GNNs have the ability to model complex relationships and heterogeneous graphs by incorporating edge features and different types of nodes.

MoIR:
MolR uses GNN-based technology to encode a molecule structure and incorporate chemical reaction information to enhance learning. It stands on the solid chemical graph architecture and combines templates of reactions, which adds versatility and reflects the spirit of organic transformations. Here, MolR's goal is to develop a molecular representation that not only is expressive but can be applied to problems like Graph-Edit-Distance and molecule property visualisation. The embedding models like Word2vec and TransE of natural language processing (NLP) have shown meaningful and interpretable results. This inspired the development of MoIR. MoIR archives a high accuracy and interpretability in the representation and analysis of molecular structures by applying the same principles to molecular graphs.

Convolutional Neural Networks:
Convolutional neural networks (CNNs) are feed-forward neural networks that use filters or kernel optimisation to teach themselves feature engineering. CNNs mainly work with grid-like data structures, especially images. In the first layer named convolutional layers, it applies learnable filters known as kernels to the input data which performs convolutions to extract local patterns and features. Then downsample is done by pooling layers to the feature maps produced by the first layers. This process

reduces their spatial dimensions, keeping important information. Finally, the last layer combines the extracted features from previous layers for prediction or classifications. CNNs excel at tasks including object detection, picture segmentation, and classification of images., due to their hierarchical feature learning and parameter sharing properties. CNNs have completely revolutionised several fields mainly in computer vision, achieving remarkable performance in various applications.

Graph Convolutional Neural Networks:
Graph Convolutional Neural Networks (GCNs) are a variety of neural networks that use graph structure as an input. Different from traditional Convolutional Neural Networks (CNNs) which work with gridded data such as images, GCNs are able to analyse the graph data which are usually represented as edges connecting nodes just like the ones you see a lot in social networks, recommendation systems and molecular biology. GCNs extend the idea of convolutional neural network to graph by applying in each node's neighbours to update its feature representation. This is usually carried out through a propagation rule, where each node's feature gets updated by a sum of its neighbour's features weighted with an activation function. GCNs reveal both the structural and feature information in a graph In this way which makes the system better suited for node classification, link prediction and graph classification.

Relational Graph Convolutional Networks:
Relational Graph Convolutional Networks (R-GCNs) are modified and improved versions of Graph Convolutional Networks(GCN) which mainly focus on graphs with lots of relational information. Traditional GCNs usually focus on aggregating information from neighbouring nodes without considering the types of relationships between them. But in the case of R-GCNs, they explicitly model the different types of relations present in the graph. During the convolutional operation of R-GCNs, each edge of the graph has a different relation type and the network learns the distinct weights for each type. Here, the R-GCNs is able to identify the degree of importance in each relationship when it comes to aggregating from the neighbouring nodes. By using relation-specific weights, R-GCNs can effectively capture the semantics encoded in the graph's relational structure, which makes them suitable for tasks like knowledge graph completion, entity classification, and link prediction in relational datasets.

2.2 Related works

In this chapter we have gathered knowledge from various recent published papers in order to understand the current domain of methods and models under which diseases are precisely detected. At the same time, we gained a comprehensive idea on how Graph Neural Network(GNN) is currently being utilised and how it has been funtional throughout different fields of work.

In this paper, Gil and Rowley use machine learning(ML) techniques to predict protein reactivity of molecules [1]. The authors mainly focused on distinguishing between covalent and non-covalent inhibitors by using various ML techniques, including conventional methods like Morgan fingerprints and advanced approaches like Graph Neural Networks (GNNs). According to the paper, the GNN models outperform conventional methods, showing improved accuracy in detecting protein-reactive compounds. However, the authors mentioned challenges in accurately predicting reactivity due to factors like protein environment and limited representation of novel chemical motifs. In spite of these challenges, this paper shows the potential of ML models in drug discovery by aiding in compound screening and identifying potentially reactive molecules.

Li et al. introduced MoIR, a Graph Neural Network (GNN) based model for learning molecule representations [3]. This model captures structural and chemical properties of molecules in order to enable tasks like Graph Edit Distance (GED) predictions and embedding visualization. The authors evaluated on the QM9 dataset which comprises molecule pairs. The results in the paper shows the effectiveness of MoIR's in approximating GED, preserving structural similarity and lastly, organizing molecule embedding based on properties like permeability and size. The authors also mentioned that existing methods for molecule representation are divided into SMILES based and structure base approaches. The MoIR belongs to structure based models which utilize GNNs to overcome limitations of other traditional fingerprint based models. Overall, MoIR shows a promising method for learning molecule representations capturing both structural and chemical properties effectively.

In their paper for predicting antiviral drugs and their interactions with coronaviruses, Mswahili et al. used Graph Neural Networks (GNNs) [2]. The author leverages a heterogeneous graph using various types of features like topology, sequence and location to represent interactions between drugs(DCC) and corona viruses (CvT) where the nodes represent samples(DCC,CvT) and edges represent their interactions. The authors explored different GNNs variants, such as Graph Convolutional Networks(GCNs) and Relational Graph Convolutional Networks(R-GCNs). The models were runned on a dataset with various features, split into training and test sets. Among the models, the single-layer Relational Graph Convolutional Network (R-GCN) performed best with high accuracy, sensitivity, and specificity, particularly when combining multiple features. The authors also mentioned that other models lacked behind due to issues like vanishing gradients and overfitting in deeper models. In spite of the challenges and data limitations, the paper suggests that GNNs are highly effective for antiviral drug prediction with potential future improvements.

Hirohara et al. introduces a groundbreaking deep learning model for compound classification and motif detection in chemical analysis utilizing Convolutional Neural Networks (CNNs) on a dataset with SMILES notation [4]. The paper highlights the importance of machine learning in chemical analysis, especially in predicting the interactions of compound-protein interactions for drug discovery. The authors used SMILES notation as this offers a linear representation which enables CNN applications for virtual screening and motif detection. SMILES strings can be utilized

to represent chemical compounds which allows the CNN to automatically extract low-dimensional representations and effectively classify compounds. The proposed model achieved superior performance on the TOX21 dataset compared to other conventional fingerprint methods. Furthermore, the authors highlighted CNN models interpretability through its ability to detect chemical motifs which helps to understand the prediction outcomes by identifying crucial substructures within compounds. Lastly, the proposed model shows promise for compound classification and chemical motif detection, providing valuable insights for drug discovery.

Ahmed and Kashmola, in their paper incorporate several variants of CNN models in order to classify different types of skin lesions. A much more robust outcome was achieved for image classification due to the enhancement of the models and training methods [5]. These performance of models were achieved by intermixing them with other algorithms. Classification accuracy was the highest when the following models were hybridized with each other:

- VGG16 architecturegave with Artificial Bee Algorithm
- AlexNet architecture with Particle Swarm Optimization algorithm
- VGG19 architecturegave with the Particle Swarm Optimization algorithm
- ZfNetarchitecture with Bat Algorithm

Hossain et al. classified kidney images by incorporating 3 types of convolutional neural network classificatoin methods [6]. With the help of deep neural networks, they were able to classify which images of kidney had a cyst, stone , tumor or if it was normal. With the halp of an algorithm known as watershed, the images were segmented based on the affected area. Models such as ResNet50, EANet and an altered version of convolutional neural network were used. The altered version attained the most accuracy of 98.66%, where the other two, EANet and ResNet50, attained 83.65% and 87.92% accuracy respectively.

Nandy et al., in their paper explores an intelligent heart disease prediction system which is based on the Swarm Artificial Neural Network (Swarm-ANN) strategy [7]. This strategy is a hybrid technique which combines Artificial Neural Networks with swarm optimization methods. In this technique, they generate a population of ANNs with random weights. Then the swarm optimization algorithm is used to guide the ANNs to the optimal result. They also showed the input variables for their model training. The output was a probability score which indicates a patient's risk of heart disease. After evaluating the Swarm-ANN system on a benchmark dataset , the result showed that it performs better than other methods for predicting heart disease. They got 95.78% accuracy in their suggested model.

In their paper for Brain Age prediction, Gao et al. proposed Graph Neural Network (GNN) model using rs-fMRI in patients with Alzheimer disease(AD) [8]. According to the authors, brain age prediction is crucial as it helps to detect Alzheimer early and improves the diagnosis and treatment of Alzheimer's disease. The authors obtained the dataset from Alzheuner's Disease Neuroimaging initiative (ADNI). A

substantial number of PET, MRI, and other medical image data were included in the dataset. The data went through several preprocessing steps. Then the authors constructed Graph data structure where Brain regions were represented as vertices and functional connectivity matrix values were used as vertex features. To determine edges a threshold was applied. After applying the GNN model, the model performance was evaluated using mean absolute error (MAE), root mean squared error (RMSE) and pearson correlation coefficient (PCC). The authors also compared their model to six other regression methods which are SVR, RFR, LASSO, LR, AlexNet and AE. The goal of this thesis was to determine how the brain age gap (BAG) and an AD diagnosis relate to one another. The result showed a significant increase in BAG in the Alzheimer patients compared to Healthy people suggesting BAG serves as a valuable biomarker for early diagnosis of Alzheimer disease.

Tiwari et al. discusses convolutional neural networks (CNNs) architecture for early detection and diagnosis of lung cancer [9]. The authors suggested a new profound learning based paradigm to examine dangerous knobs. They used CMixNet for lung nodule detection and faster R-CNN for nodule recognition. Gradient Boosting Machine (GMB) is used on the outlines of the intended 3D CMixNet layout for nodule characterization. The authors integrated clinical symptoms and pathogenesis to reduce false positives. In their paper, they used the Kaggle Data Science Bowl 2017 dataset which consists of CT scan images of lung cancer patients. Several preprocessing steps such as standardization, downsampling and thresholding were used. Moreover, the authors used the technique of watershed segmentation for improving the accuracy. Lastly, after evaluating the performance the authors achieved notable increase, scoring 94% affectability and 91% explicitness on The LIDC-IDRI datasets.

In their paper Zhou et al. Provide an overview of how Graph Neural Networks (GNNs) have been applied in the field of finance [10]. GNNs have become increasingly popular because they excel at modeling the relationships and dependencies found in data. The paper explores approaches, datasets and real world applications where GNNs have shown potential. These applications include portfolio optimization, fraud detection, risk assessment and market prediction. The authors also discuss the challenges and future directions, for incorporating GNNs into contexts. Furthermore the survey examines how GNNs are used in domains such as physics systems modeling fingerprint learning, protein interface prediction, disease classification and analyzing structured data from non structural sources like texts and images. GNNs utilize message passing between nodes to capture graph dependencies effectively. Variants like Graph Convolutional Networks (GCNs) Graph Attention Networks (GATs) and Graph Recurrent Networks (GRNs) have demonstrated performance, across deep learning tasks. This paper introduces a comprehensive design pipeline for GNN models, categorizes their components, highlights applications, and outlines four key open problems for future research.

Valls et al. investigated a crucial aspect of Graph Neural Networks (GNNs) in the context of Knowledge Graphs (KGs) and their application in clinical triage, how the flow of embedding information within GNNs impacts the prediction of links in Knowledge Graphs [11]. This suggests that understanding the information propagation mechanism within GNNs is crucial for achieving optimal performance. A

proposal of decoupling GNN connectivity regarding a mathematical model that separates the GNN connectivity from the connectivity of the graph data. This indicates that there may be benefits to customizing the GNN's connectivity based on domain-specific knowledge. That results demonstrate that integrating domain knowledge into the GNN connectivity leads to improved performance compared to using the same connectivity as the Knowledge Graph or allowing unconstrained embedding propagation. This emphasizes the importance of domain-specific information in designing effective models.The significance of negative edges in achieving accurate predictions are also being highlighted. This suggests that considering both positive and negative edges in the training process is crucial for obtaining good results.Also it has been observed that using too many GNN layers can lead to degraded performance. This indicates that there might be an optimal depth for GNN architectures in this specific context.Overall, research addresses important challenges in training GNNs for healthcare applications, providing valuable insights for the community.

The research paper by Zipfl et al. focused on improving the validation process for autonomous vehicle driving models [12]. The research paper presents a method that integrates image based representation of factors using Convolutional Neural Networks (CNNs) and a graph based representation of traffic participant relations using Graph Neural Networks (GNNs). The goal of this combination is to enhance the precision of predicting the trajectory, for traffic participants. By employing imitation learning the system can identify traffic scenarios, for testing vehicles. This topic holds significance as it has the potential to enhance both safety and efficiency in vehicles. The authors of this paper proposed to use the activity of these networks as a measure of interactivity of a traffic scene. Some additional points of this paper are:

- The approach taken by the authors is quite fascinating because it combines two types of networks; CNNs and GNNs. CNNs are great at understanding features in images while GNNs excel at learning the relationships between elements in a graph. This unique combination allows the model to grasp both the social aspects of traffic scenes.

- I find the author's evaluation to be comprehensive and convincing. They have utilized a large motion dataset to assess their models' performance. Have successfully demonstrated its ability to accurately predict the trajectories of traffic participants. Additionally they have shown that incorporating relationship information plays a role in identifying traffic scenarios.

- It is worth noting that this work by the authors holds potential for influencing the advancement of vehicles. By employing trajectory prediction based on imitation learning they can effectively detect traffic scenarios thereby enhancing safety and efficiency during vehicle testing.

In another paper authored by Hu et al. they introduce a graph network (GNN) model capable of predicting a patients risk of developing ADRD (Alzheimer's Disease and Related Dementias) [13]. Furthermore this model provides insights into the factors contributing to that risk.When it comes to predicting the risk of Alzheimer's disease and related dementias (ADRD) the nodes in the graph can represent things

like patients, medical codes or other factors that contribute to the risk. The edges in the graph show connections between patients, like when certain medical codes occur together or if there is a family history of ADRD. The Graph Neural Network (GNN) is trained using data from patients whose ADRD status is already known. The graph neural network(GNN) collects knowledge on the relationships among the data points that exhibit the predictability of ADRD risks. After training the model, it can be used to predict the ADRD risk of new patients. Each patient will be given a score by the model which represents the likelihood of developing ADRD for a patient. Moreover, this model is also capable of providing explanations for its given predictions. This is because the GNN can detect the connections between the data points that had the influence on the prediction. For instance the GNN could identify that a patient's chances of developing ADRD are high due to both having a family history of ADRD and recently being diagnosed with diabetes. Compared to baseline models the VGNN model showed a 10% improvement in the area under the operating characteristic curve (AUC ROC). Alzheimer's disease and related dementias pose public health challenges ranking as the leading cause of death in the US. This emphasizes the need for prediction of Alzheimer's Disease and Related Dementias (ADRD) risk. This project suggests integrating machine learning with claims data to uncover risk factors and understand how various medical codes relate to each other. This approach has potential to provide an understanding of ADRD risk factors. The project tackles the problem of lacking explanations that humans can easily interpret for predictions. It introduces a technique to evaluate how significant relationships are and their impact on predicting ADRD risk ensuring interpretations. The study utilizes a Variationally Regularized Encoder Decoder Graph Neural Network (VGNN) to estimate the likelihood of ADRD occurrence. The study establishes three scenarios to assess how effective the VGNN model is using Random Forest and Light Gradient Boost Machine as baseline models. This analysis demonstrates the excellence of the suggested method. This approach does not improve the modeling of Alzheimer's disease and related dementias (ADRD) risk. Also it presents possibilities for expanding into other types of predictions such, as image analysis and claims data. This suggests broader applications beyond ADRD risk prediction. Overall, the study demonstrates a promising approach in leveraging GNNs with claims data to improve ADRD risk prediction, while also providing valuable insights into the impact of interconnected medical code relationships. This methodology has the potential to make significant contributions to both ADRD research and broader applications in healthcare prediction.

Sunil et al. explores GNN architectures in order to understand schizophrenia by analyzing rs-fMRI data [14]. The authors mentioned the challenges of detecting schizophrenia and highlighted the importance of early detection of the disease. This paper further shows the potential of GNN for understanding . In this paper, the authors used a dataset from OpenNeuro repository which has both structural and functional MRI scans. The authors also addressed that the dataset has class imbalance. In this proposed approach, the authors preprocessed the dataset to clean and prepare it by applying the following techniques: slice timing correction, outlier detection, segmentation, normalization, functional realignment and smoothing. Then they developed a deep graph convolutional neural network for feature extraction and classification tasks. After running the GNN model the authors achieved high

accuracy in classifying schizophrenia. The authors then compared the results with other traditional machine learning models stating the strength and weakness of each method.

Jie et al. proposed the Pyramid GNN model for classifying COVID-19 cases from chest X-rays [15]. The proposed model uses a Convolutional Neural Network(CNN) which extracts features from chest X-rays images by dividing them into patches and process each patch to derive feature vectors. Then the Pyramid GNN interprets those extracted patch features as nodes in a graph structure. Through the GNN model the nodes transfer information among its neighboring nodes using graph convolution operations which makes the features even better. Lastly, a Multilayer Perceptron classifies COVID-19 cases by receiving the features from the previous GNN layers. The authors evaluated their proposed model and compared its performance with other existing deep learning models on three different CXR image datasets. The pyramid GNN demonstrated superior accuracy in COVID-19 classification compared to the other models.

This research work by Zhang et al. constructed a novel model GNN-DOL for automatic mitosis detection from video in cell proliferation analysis [16]. According to the authors, previous existing methods utilizes object detection algorithms mixed with link prediction but they fail to consider the biological constraint that a cell can divide into two in the next frame which results in accuracy drops. Considering this the authors proposed a GNN-DOL model which integrates a graph neural network(GNN) with a differentiable optimization layer(DOL). In this process the authors preprocessed using U-net for cell position prediction. It is followed by GNN-DOL processing which utilizes message passing and quadratic programming to enforce the mitosis constraint, improving the accuracy in identifying parent cells. The result shows that GNN-DOL significantly gives better accuracy to previous methodologies in mitosis detection especially with multiple division events. The authors also mentioned that the use of quadratic programming increases the computational demand.

Li et al. introduces the multiphysical graph neural network(MP-GNN) model for covid-19 drug design [17]. The proposed methodology has two main parts. Firstly, the head part is responsible for converting node vector information to hidden feature vectors from the graphs. And the second part called tail is a fully connected neural network which learns the binding affinity from those feature vectors. The authors compared the performance of MP-GNN against existing other traditional machine learning models across three PDBbind dataset. The results show that MP-GNN outperforms other models in both accuracy and efficiency which shows its potential in drug design and ability to handle complex molecular data. Additionally, the MP-GNN model shows better results in predicting binding affinities than MathDL which is the leading method for this application.

By encoding molecules into numerical vectors, Guo et al. review on graph-based molecular representation learning (MRL) emphasizes the significance of maintaining molecular structures for tasks like property and response prediction [18]. The review classifies MRL techniques into two categories: 3D molecular graphs, which

include geometric information critical for particular physical qualities, and 2D molecular graphs, which are basic but lack spatial features. It covers several approaches, such as the 2D-based MRL with Message Passing Neural Networks (MPNN) and its variants (GCN, GIN, and GAT), the 3D-based MRL with techniques like SphereNet and DimeNet that concentrate on spatial data, and the Knowledge Graph-based MRL that integrates external chemical knowledge. The main uses of this technology in drug discovery are described, including drug-to-drug interactions, property prediction, molecular creation, and reaction prediction. Along with outlining future research paths that emphasize spatial learning, model explainability, and controlling data scarcity using self-supervised and meta- learning techniques, the paper also offers a complete list of datasets and benchmarks utilized in MRL research.

Krenn et al. looks at how SELF-referencing embedded strings (SELFIES) have developed and how they affect molecular string representations in chemistry and materials science applications of artificial intelligence (AI) and machine learning (ML) [19]. Throughout history, molecular representations have progressed from ideas from the early eighteenth century to contemporary string representations such as INCHI, DEEPSMILES, and SMILES. Nevertheless, these conventional techniques frequently produce inaccurate molecular graphs. SELFIES is a formal grammar system that follows chemical and physical rules to ensure 100% correctness of created structures. It was introduced in 2020 and addresses these restrictions. SELFIES have demonstrated exceptional performance in a range of AI and ML activities, such as combinatorial chemistry, generative models for drug development, and molecular property prediction. Future work will focus on expanding the use of SELFIES to more intricate chemical domains, enhancing their interpretability and readability, and better integrating them into computational workflows. All things considered, SELFIES mark a major breakthrough in molecular string representations, propelling advances in chemistry and materials research.

Reiser et al. examined the function of graph neural networks (GNNs) in materials science and chemistry [20]. It emphasizes how important it is for GNNs to handle graph or structural representations of molecules and materials in order to predict material properties, speed up simulations, create new structures, and forecast synthesis routes. By learning from graph structures, GNNs expand on standard machine learning. Well-known varieties, such as Message Passing Neural Networks (MPNN), produce graph-level embeddings for tasks including regression and classification. Reviewing state-of-the-art GNN designs using benchmark datasets like QM9 and the Materials Project, it covers spectral convolution, spatial convolution, attention mechanisms, and equivariant GNNs. GNNs are used in molecular properties prediction (e.g., ADMET prediction), molecular dynamics simulation acceleration, chemical reaction and retrosynthesis prediction, crystalline and solid-state system analysis, and disordered system and defect modeling. In order to improve the accuracy and usefulness of GNNs in the discovery of new materials, future efforts will focus on increasing dataset expansion, enhancing data efficiency, and improving model interpretability.

Chapter 3

Overall Procedure

The primary motive of our proposed model is to detect the severity of COVID in the molecules as accurately as possible. We would be implementing Graph Neural Network(GNN) on our dataset. Figure 3.1 shows a flowchart based on our GNN detection model.

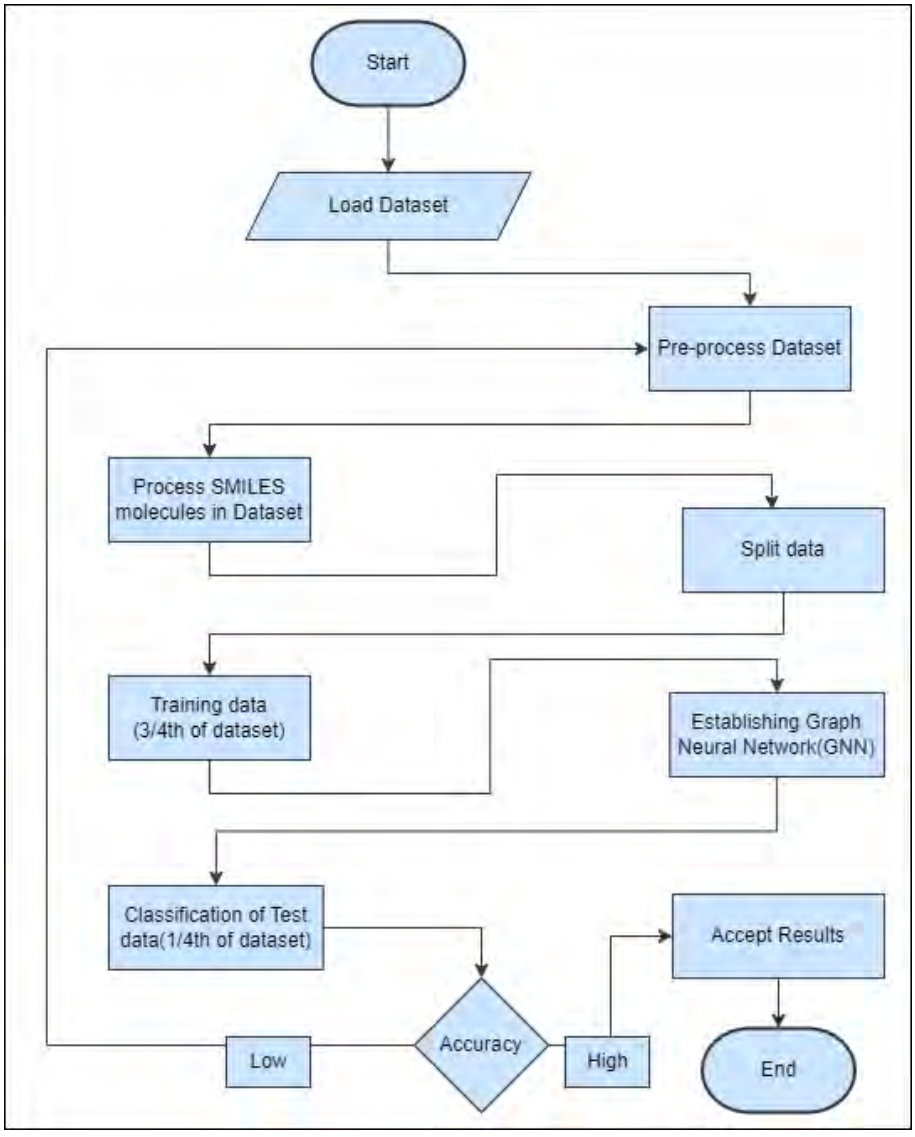

Figure 3.1: The Flowchart Of GNN Model

In order to implement GNN successfully we would need to follow some rudimentary steps. These steps are as Follows:

- **Data Preprocessing:** During this stage we would have to makes changes to our data that would help our data perform better than it would if we did not pre-process it, some examples would include, fixing null values, removing columns of data with weak correlation and oversampling or undersampling our data based on amount of unique results present.
- **Molecule Processing:** With the help of DeepChem featurizer, we've extracted several node and edge features from the molecules in the dataset. After that with PyTorch geometric graph function, we convert those features into a data object which we later on feed to our model.
- **Data Splitting:** We split our data into two parts, Train data, which consists three-fourths of the entire dataset, and test data which has the remaining one-fourth of the dataset.
- **Training and Testing:** We will trained our GNN model with the train dataset initially, and then moved on to test our model using the test dataset. This helped us obtain our results such as accuracy, precision, recall and F-1 score.

If we get below-satifactory results, we will redo these steps in order to get better results, and keep on repeating these steps until we get acceptable outcomes.

Chapter 4

Methodology

4.1 Data Description

We have implemented our model on a dataset known as "sarscov2", which consists of over eight thousand molecular data in Simplified Molecular Input Line Entry System(SMILES) representation. The dataset is collected from ChEMBL Database, which has a wide range of datasets of handpicked bio active molecules with drug like properties. Fig 4.1 shows their representation in Simplified Molecular Input Line Entry System(SMILES) format, and fig 4.2 shows their representation in molecular format.

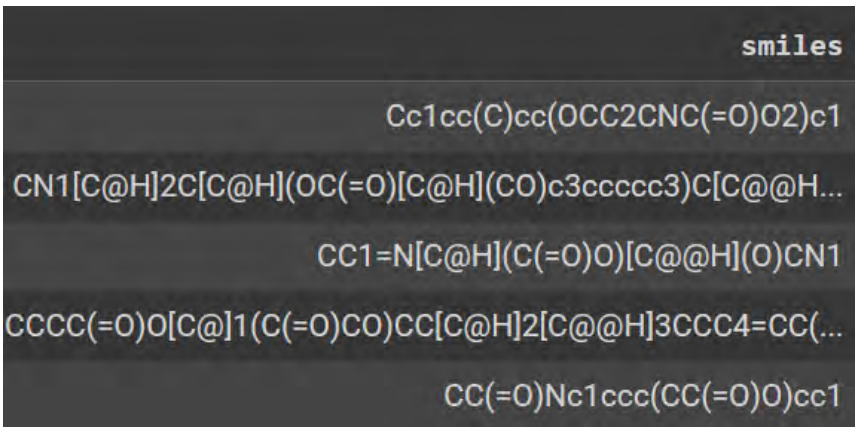

Figure 4.1: SMILES representation

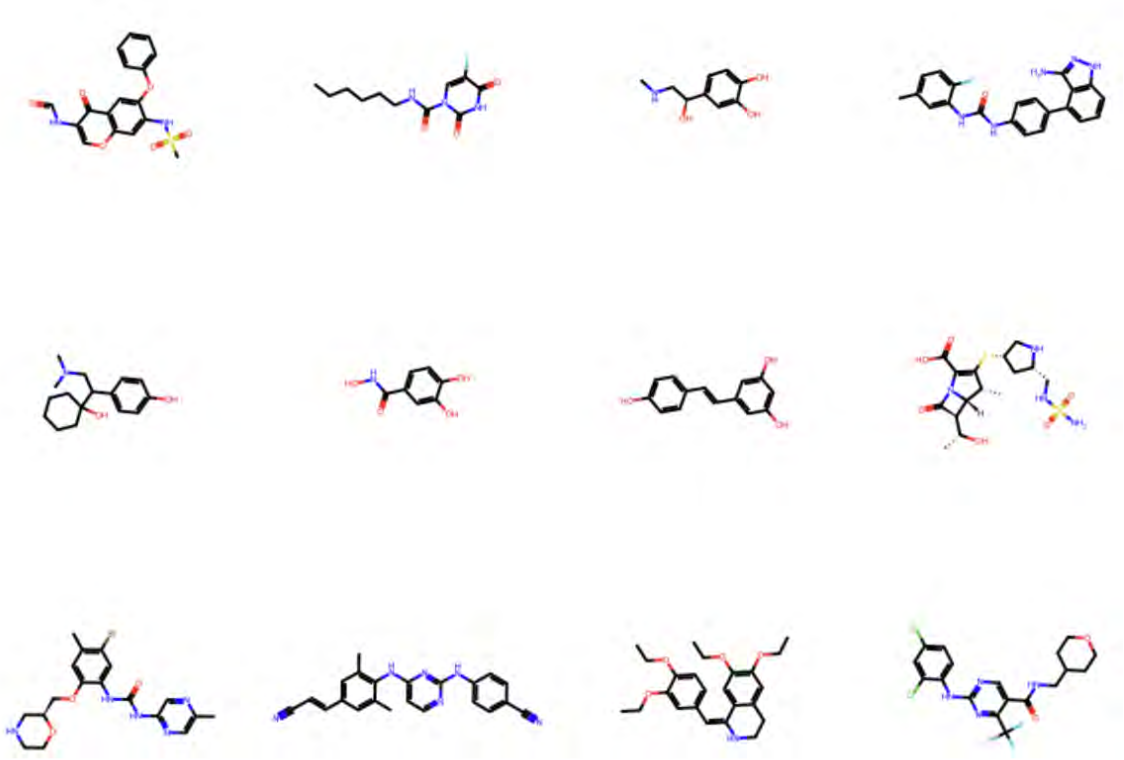

Figure 4.2: Molecular representation

4.2 Data Preprocessing

Data Preprocessing was implemented in three ways:

- **Handling of null values**
- **Handling minority class**
- **Removal of irrelevant columns**

We will now explore how we preprocessed our data based on the aforementioned methods:

4.2.1 Handling of null values

While initially checking the dataset for null values, we found out that all the columns had values in them for all the rows, except for only one column which was labeled "canonical_smiles". Fig 4.3 shows the number of values missing from the column "canonical_smiles". This specific column had missing data on a very few number of rows. And since it was data in SMILE representation and not just numerical value, we could not just find the mean and fill the null values with the mean values. So, our only other option was to remove the rows containing null values.

```
molecule_chembl_id     0
canonical_smiles      56
standard_value         0
activity_class         0
```

Figure 4.3: Null value count

4.2.2 Handling minority class

Moving forward, we were presented with another problem. Our target or output variable, known as "activity_class" in the dataset we have used, had 3 values in it. The three values were "intermediate", "active" and "inactive". while the first two values were balanced in respective to each other, the value "inactive" had very low count. Fig 4.4 gives us the initial visual representation of the column "activity_class". While Fig 4.5 shows the actual counts of the values.

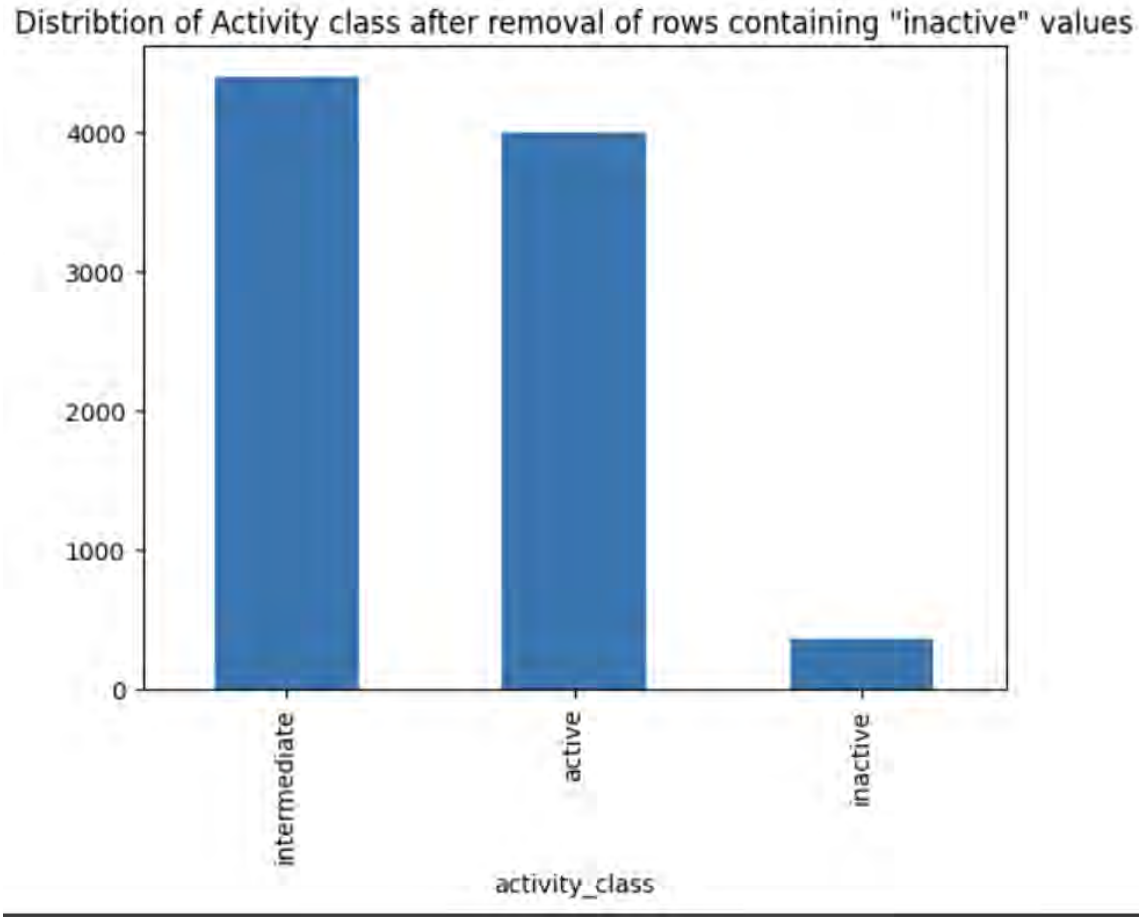

Figure 4.4: Initial bar chart for activity_class

activity_class	
intermediate	4397
active	3998
inactive	368

Figure 4.5: Count for activity_class

As we can clearly see, "inactive" has a very low count compared to the other two values. We chose not to undersample as it would mean we would lose most of our data. We also chose not to oversample as oversampling a minority class with severely low count could lead to overfitting, which would give us unreasonable results. So our only option left was to remove the minority class entirely. But as the value "inactive" was vital to the classification of our dataset, we needed to keep that part of the data as well. So we decided to work on three versions of our dataset.

In the first version, version 1, we will be removing the inactive class as a whole, and just work with the other two classes. Fig 4.6 provides the visual representation of version 1 after the removal of minority class:

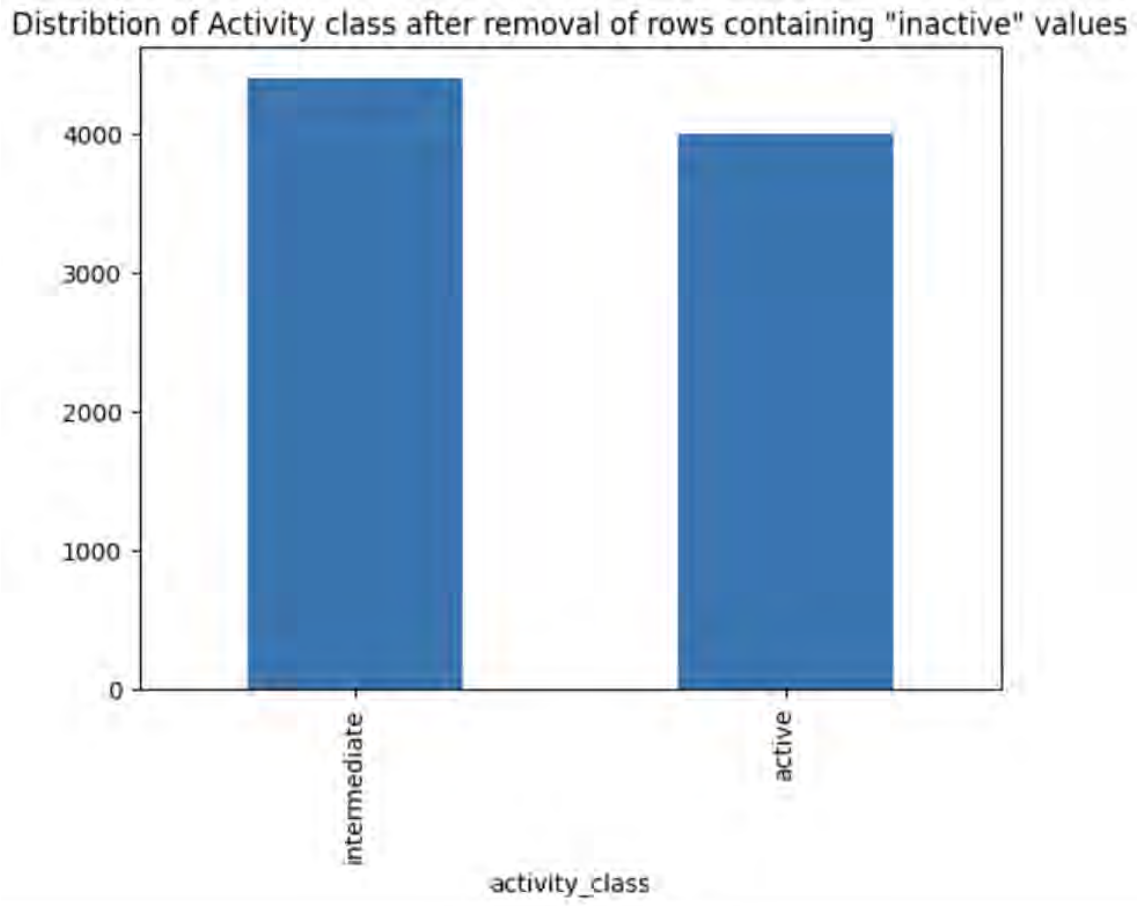

Figure 4.6: Dataset(version 1) bar chart for activity_class

As for the second version of the dataset, version 2, we have kept the "inactive" class.

And we have also realized that we could merge the “active” and “intermediate” class together into one class as they both have molecules affected with COVID-19 in them. Fig 4.7 shows us how it looks after concatenating these pair of classes.

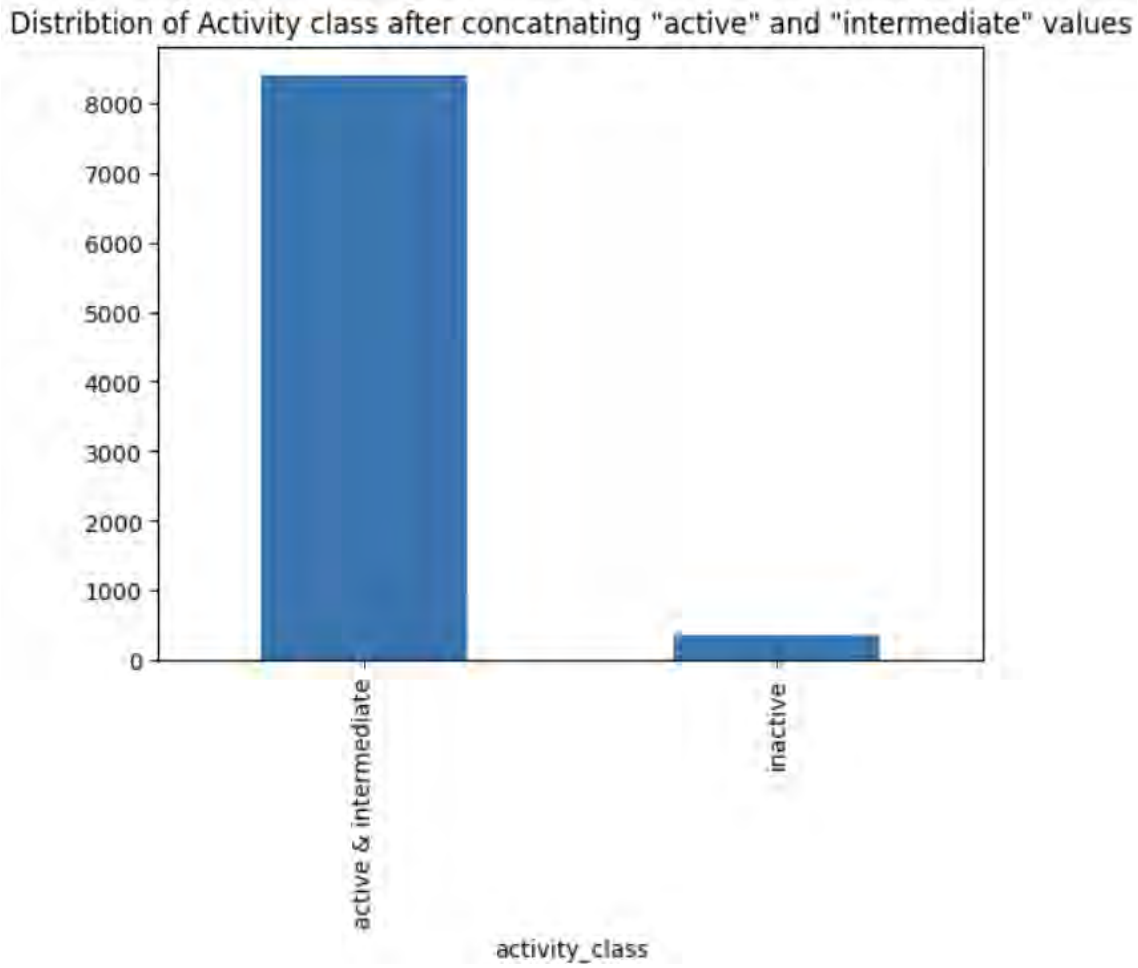

Figure 4.7: Initial Dataset(version 2) bar chart for activity_class

After the concatenation of data, we obviously had to address the class imbalance, which we did through oversampling our dataset. Note that, before oversampling we have already split our dataset to test and train data, in order to not get biased results. So after splitting our data we were left with 75% of our initial data for our train set. Fig 4.8 shows the value counts for the two classes, and Fig 4.9 gives us a proper visualisation of the class imbalance.

```
activity_class
active & intermediate    6307
inactive                  265
```

Figure 4.8: Count for activity_class for version 2 dataset

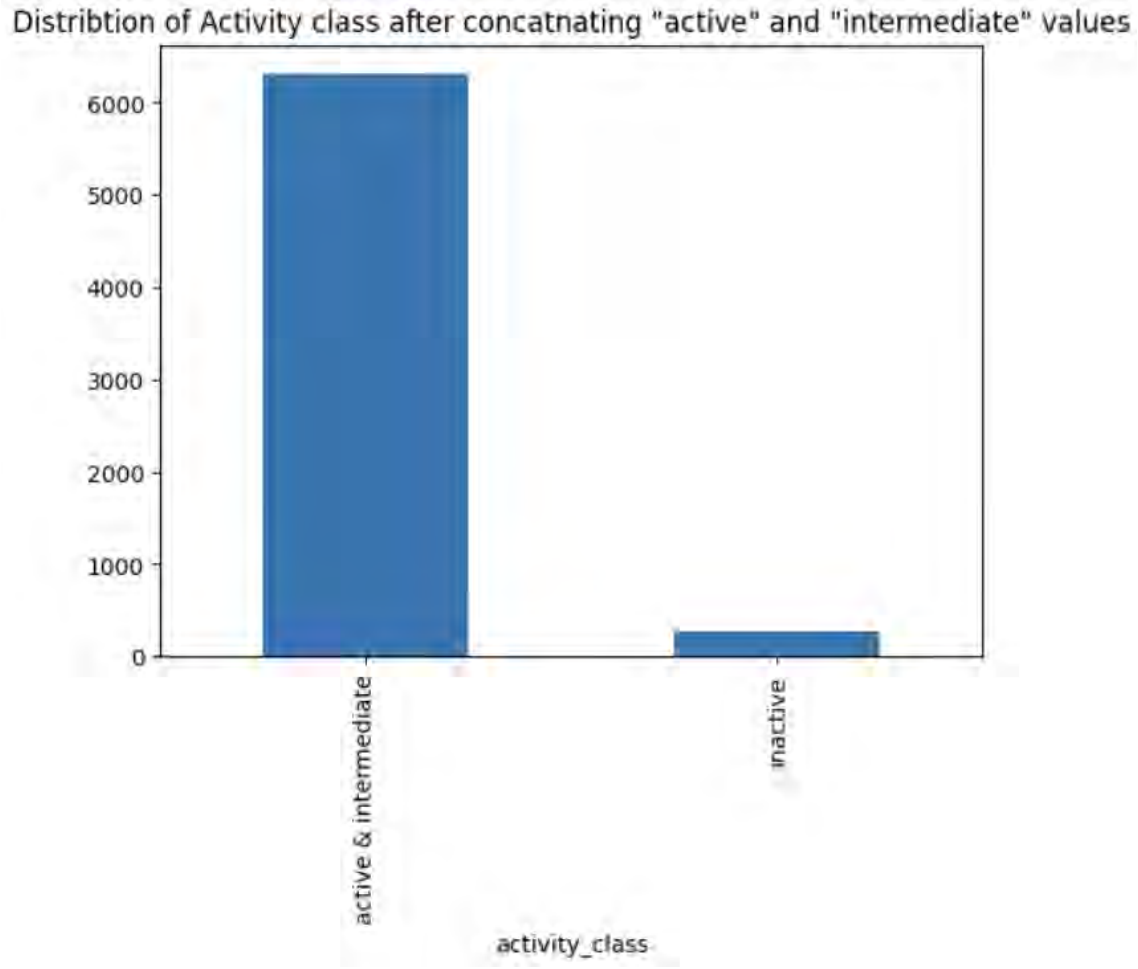

Figure 4.9: Initial Train Dataset(version 2) bar chart for activity_class

Finally, Fig 4.10 and 4.11 shows us our final value count and bar chart representation of the second version of our dataset, after being Oversampled.

```
activity_class
active & intermediate    6307
inactive                 6095
```

Figure 4.10: Count for activity_class for version 2 dataset after oversampling

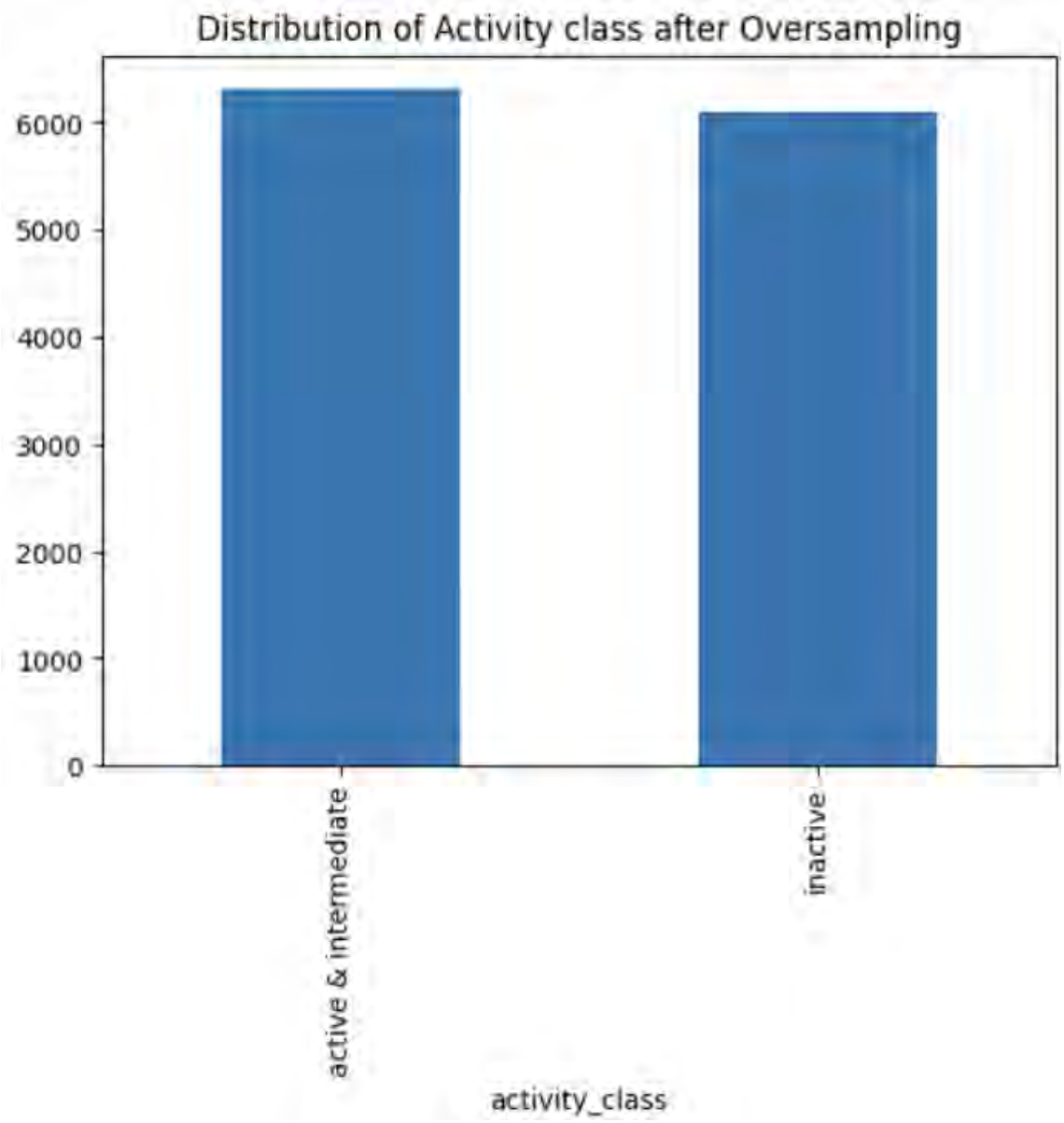

Figure 4.11: Oversampled Dataset(version 2) bar chart for activity_class

Lastly, in the third and final version of the dataset, version 3, we kept everything as it is, split the data into test and train sets, and finally oversampled the "inactive" class in the train set. Fig. 4.12 and 4.13 shows us our final value count and bar chart representation of the third version of our dataset, after being oversampled.

activity_class	
intermediate	3313
active	2981
inactive	2780

Figure 4.12: Count for activity_class for version 3 dataset after oversampling

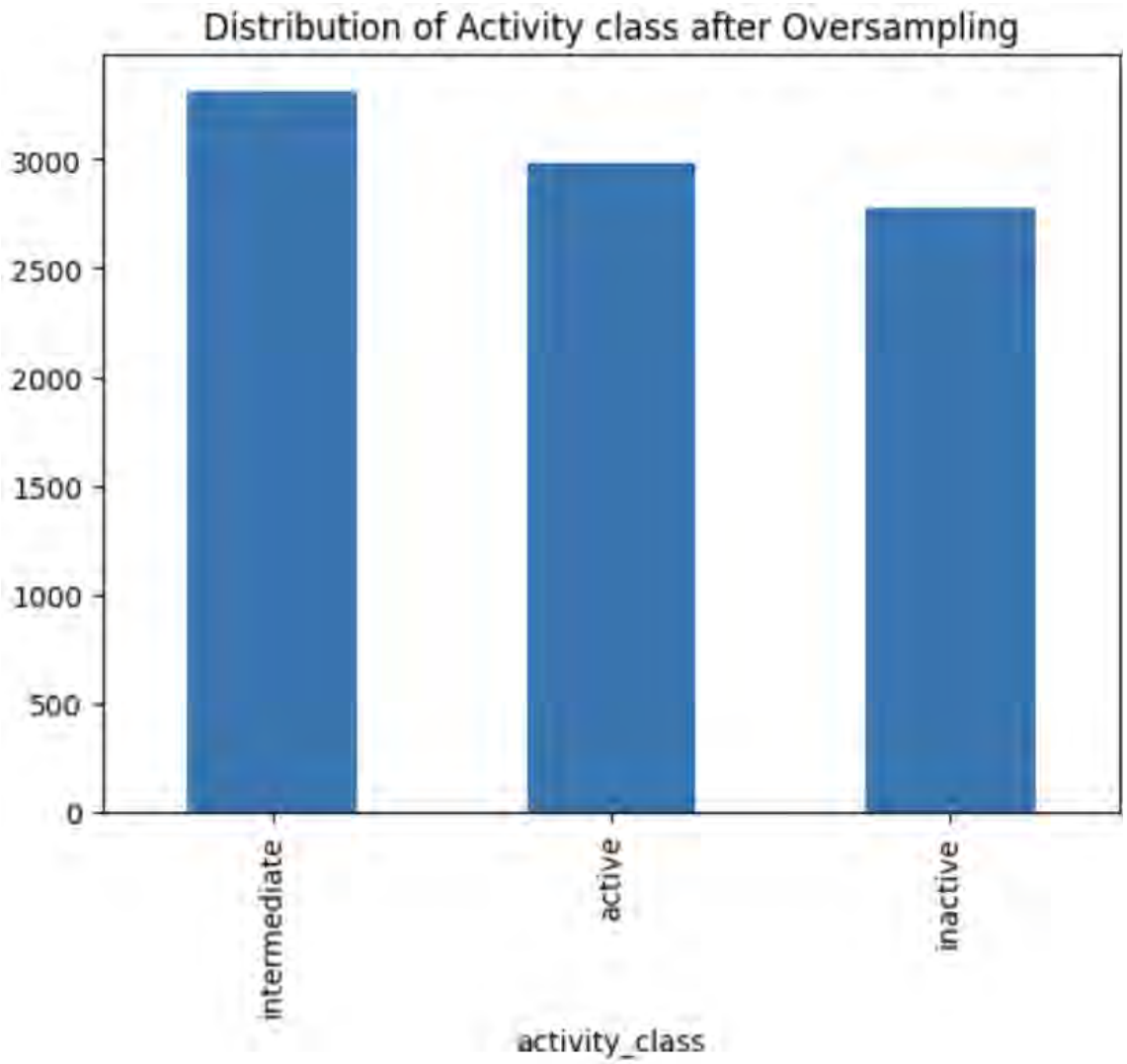

Figure 4.13: Oversampled Dataset(version 3) bar chart for activity_class

4.2.3 Removal of irrelevant columns

Lastly, a columns was found to be unnecessary for our work, to be specific the column named "molecule_chembl_id". This column just provided ids for the molecules, therefore, as they pose no other significance rather than numbering the data, it was of no use. So we went ahead and removed that column.

4.3 Model

4.3.1 Model Overview

In GNN, data needs to be in graph form. The node and edge features along with structural properties of the graph data is fed through message passing layers. These layers construct the node embeddings, that contain the knowledge about other nodes and edges in a compressed format. Figure 4.14 gives us a visual representation of how it occurs(here, x1, x2 and x3 are node embeddings):

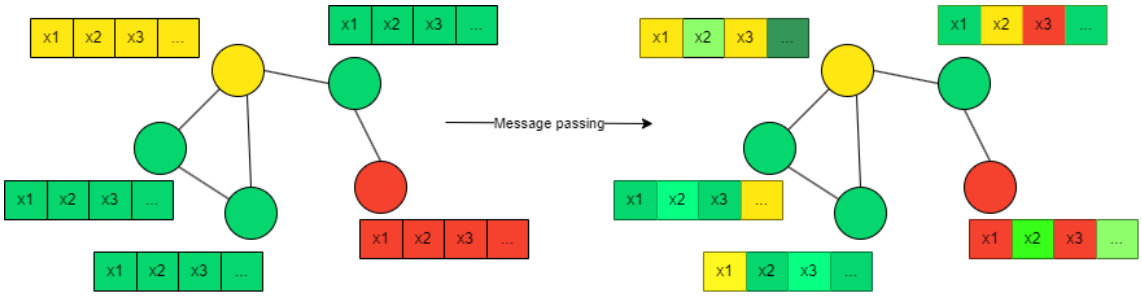

Figure 4.14: GNN working mechanism

Let us assume on the left we have our input graph. Let us focus on the yellow node. To update the node state, we collect the information of the direct neighbors, which means we perform the message passing. What we end up with, is the information about our current node state, and the information about its neighbors' node states. Then we perform an aggregation in the neighbors' states to combine that information. Finally we put the current node state and the the combine neighbor information together in order to obtain a new state or embedding. We then replace the current node embedding with this new state or embedding through the message passing layer. We can clearly see on the right part of figure 4.7 the result of this combination. Note how, the yellow node has node embedding of different colors, which signifies that it has now information of its neighbouring nodes in its own node embedding. This passing is applied for all nodes, hence you can see the changes in all the node embedding of the nodes across the graph. And with every message passing layer applied, the nodes get to know more about the nodes connected to its neighboring nodes, and thus, each nodes gets to know information about all the nodes in its graph.

4.3.2 Model Algorithm

Algorithm 1 shows the algorithm for our GNN model used. Note that num_classes will be set to 3 during running the model on version 3 of the dataset, for multi-class classification. We have used 3 graph attention convolution layers and three TopKpooling layers as the molecule graphs are not that big 3 layers were sufficient enough. Additionally we have three attention heads("heads = 3") for the attention mechanism. Additionally we use a fully connected network("head_transform") to transform back to their initial node feature size, because three heads generate three times the node embedding(in pytorch geometric, three heads create generate 3 different output vectors), so we need to convert it back to the original embedding size to pass it to the next layer.

In the forward function the data(node embeddings) is in x, so the node embeddings are passed through the first convolutional layer with three attention heads("x = Apply conv1 on x with edge_index"). Then it is transformed back into the initial shape("x = Apply head_transform1 on x"). After that pooling is applied to obtain a new graph to essentially get a new graph like before. And for each graphs global max pooling and global average pooling was applied and the results were stored in x1, x2 and x3 respectively. Following that, these three pooled vectors of intermediate graphs are concatenated and stored in x. And finally, we pass everything

Algorithm 1 GNN algorithm

Set num_classes to 2
Set embedding_size to 1024
Define three GAT convolution layers (conv1, conv2, conv3)
Define three linear transformation layers (head_transform1, head_transform2, head_transform3) with input size embedding_size * 3 and output size embedding_size
Define three TopK pooling layers (pool1, pool2, pool3)
Define two fully connected layers (linear1, linear2)
Apply conv1 on input features x with edge_index
Transform the output using head_transform1
Apply pool1 on the transformed output, updating x, edge_index, edge_attr, and batch_index
Concatenate global mean pooling (gap) and global max pooling (gmp) of x to form x1
Apply conv2 on x with edge_index
Transform the output using head_transform2
Apply pool2 on the transformed output, updating x, edge_index, edge_attr, and batch_index
Concatenate global mean pooling (gap) and global max pooling (gmp) of x to form x2
Apply conv3 on x with edge_index
Transform the output using head_transform3
Apply pool3 on the transformed output, updating x, edge_index, edge_attr, and batch_index
Concatenate global mean pooling (gap) and global max pooling (gmp) of x to form x3
Combine Pooled Vectors: x = x1+x2+x3
Apply linear1 on x and apply ReLU activation
Apply dropout to the result with a probability of 0.5
Apply linear2 on the result to obtain the final class result
Return x

through two linear layers until we have an output linear layer which has two output classes("num_classes = 2").

The current implementation of Graph Attention layers for PyTorch geometric does not support edge attributes to be included, so we worked with only the node embeddings for now, while the edges were unweighted.

Chapter 5

Results

5.1 Results

We obtained some satisfactory results using our model with the version 1("inactive" class removed) of our dataset, with measured accuracy, precision, f-1 score and recall. Table 5.1 shows the results we have obtained:

Table 5.1: Version 1 dataset Results

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
GNN	87.16%	84.27%	81.06%	82.63%

These metrics indicate that GNNs provide a robust approach for accurately classifying the severity of COVID-19, demonstrating superior performance in capturing the complex relationships inherent in the sarscov2 dataset. The high accuracy indicates that GNNs can reliably differentiate between active and intermediate molecules, which is crucial for timely and effective treatment interventions.The high F1 score reflects a balance between precision and recall, showing that the model not only correctly identifies positive cases (active molecules) but also effectively minimizes false negatives. Precision and recall are particularly critical in healthcare settings where the cost of misdiagnosis can be extremely high. Figure 5.1 gives us a visual representation of the results we have obtained:

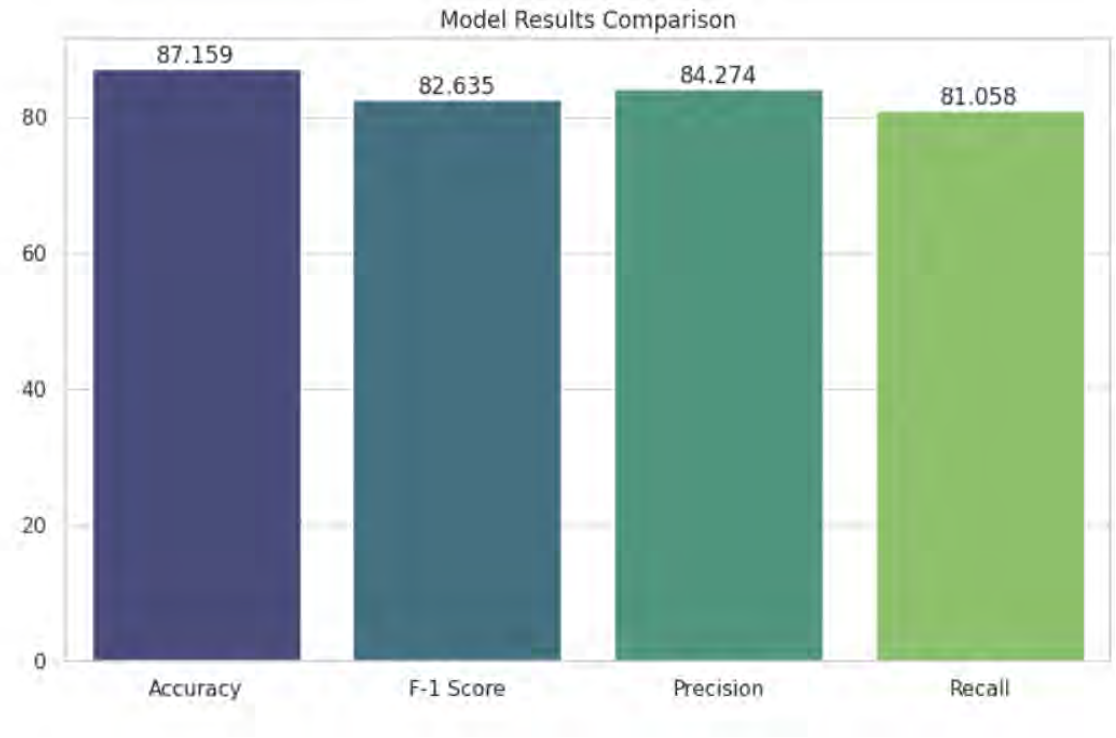

Figure 5.1: Results using GNN on version 1 dataset

As we can see, in terms numbers, our highest result was in terms of accuracy, which was 87.159% which means it was able to determine the number of true positives("active") and true negatives("intermediate") over the prediction set the most. Next comes Precision with 84.275% which means our model was considerably successful at predicting the correct positives("active") over the total positive guesses(true positive and false positive). Furthermore, comes recall, which stands at 81.058%, which means it was also good at predicting the the correct positives("active") over the total positive labels(true positive and false negative). Lastly comes F-1 score, which determined the quality of the prediction, which was at 82.635%.

As for when the model was run on version 2 and 3 of our dataset, where the "inactive" class was present, we received below satisfactory results, Table 5.2 shows the results we have obtained:

Table 5.2: Version 2 and Version 3 dataset Results

Model	**Dataset version**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
GNN	2	69.52%	65.42%	78.30%	71.28%
GNN	3	73.54%	68.23%	75.02%	71.46%

Furthermore, in Fig 5.2, we can observe the comparison of GNN used in different versions of the dataset:

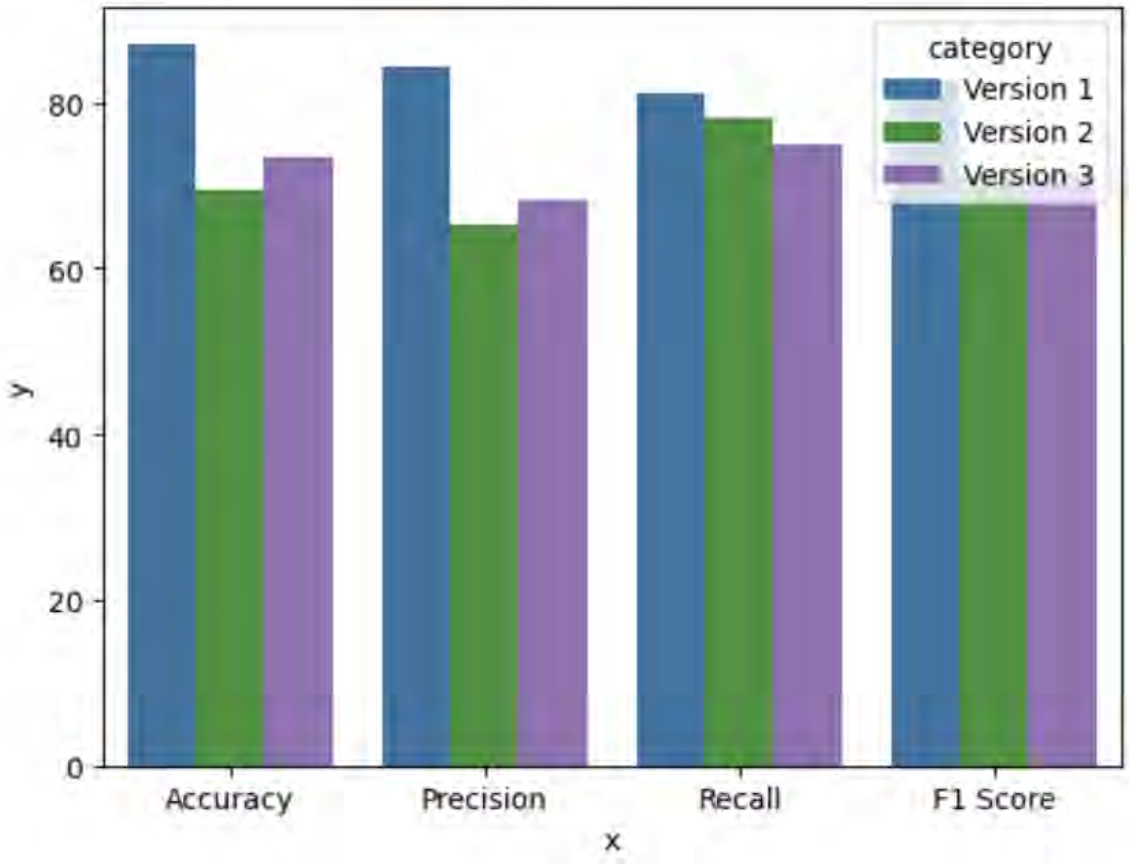

Figure 5.2: Results comparison of GNN implementation on three versions of the dataset

In analyzing the results for Version 2 and 3 of the dataset, which considered all cases including active, intermediate, and inactive, we observed that, even though version 3 performed slightly better than version 2, the performance metrics of both these datasets were notably lower compared to Version 1. Specifically, Version 2 and 3 achieved an accuracy of 69.52% and 73.54%, a precision of 65.42% and 68.23%, a recall of 78.30% and 75.02%, and an F1 score of 71.28% and 71.46% respectively. These results are significantly diminished relative to the performance of Version 1. The primary reason for this performance drop is attributed to the oversampling of the "inactive" class in both Version 2 and Version 3. This oversampling led to overfitting, where the model became too tailored to the specific characteristics of the training data, thus reducing its generalization capability on unseen data. Consequently, the model's ability to accurately classify the diverse range of cases in the dataset was compromised. The inflated presence of the inactive class skewed the model's learning process, causing it to prioritize the inactive class at the expense of accurately identifying active and intermediate cases. This imbalance not only affected the precision and accuracy but also had a significant impact on recall and the overall F1 score, highlighting the challenges and complexities of handling imbalanced datasets in disease detection using GNNs.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

To conclude, this thesis has explored the application of Graph Neural Networks (GNNs) for assessing COVID-19 severity using the SARS-CoV-2 dataset. The research findings demonstrate the significant potential of GNNs in revolutionizing molecular classification, offering high accuracy, efficiency, and effectiveness. Through experimental validation, the study has shown that GNNs can accurately classify the severity of COVID-19 in molecules represented by Simplified Molecular Input Line Entry System (SMILES) strings, achieving high accuracy, precision, recall, and F1 scores. Specifically, Version 1 of the dataset, which excluded inactive cases, outperformed Version 2 in all metrics, highlighting the importance of dataset composition and handling. These results underscore the robustness of GNNs in capturing complex relationships and dependencies within biomedical data, potentially enabling timely and accurate disease detection. Key to the success of GNNs in detection is their ability to model intricate associations in graph-structured data. GNNs have the potential to offer a more holistic approach, leveraging graph representations to uncover subtle patterns and dependencies crucial for accurate diagnosis.

6.2 Future work

While this study has provided valuable insights into the application of GNNs for detecting the severity of COVID in molecules, several avenues for future research remain:

As GNN exclusively works on graph data, in order for a GNN model to function we will have to provide it with graph data. The problem which arises here is that most of the datasets out there are not graph-based. So we have to find unique ways for converting normal datasets to graph data. For example, we can convert a tabular dataset to a graph dataset by assigning different attributes of the dataset as nodes and weighted edges.

Another issue that we faced is converting images to graph data. It has been a bigger challenge as there are very limited number of literature that successfully did this, or even discuss about how they did it. We tried to tackle this by transforming our images into one-dimensional feature vectors. We will use this feature vectors along with other attributes from our dataset to create graph data, which will then be used on executing our model.

References

[1] V. H. C. Gil and C. Rowley, "Graph neural networks for identifying protein-reactive compounds," 2024.

[2] M. E. Mswahili, G. E. Ndomba, K. Jo, and Y.-S. Jeong, "Graph neural network and bert model for antimalarial drug predictions using plasmodium potential targets," *Applied Sciences*, vol. 14, no. 4, p. 1472, 2024.

[3] H. Wang, W. Li, X. Jin, K. Cho, H. Ji, J. Han, and M. D. Burke, "Chemical-reaction-aware molecule representation learning," *arXiv preprint arXiv:2109.09888*, 2021.

[4] M. Hirohara, Y. Saito, Y. Koda, K. Sato, and Y. Sakakibara, "Convolutional neural network based on smiles representation of compounds for detecting chemical motif," *BMC bioinformatics*, vol. 19, pp. 83–94, 2018.

[5] H. M. Ahmed and M. Y. Kashmola, "Performance improvement of convolutional neural network architectures for skin disease detection," *International Journal of Computing and Digital Systems*, pp. 189–201, 2023.

[6] M. S. Hossain, S. N. Hassan, M. Al-Amin, M. N. Rahaman, R. Hossain, and M. I. Hossain, "Kidney disease detection from ct images using a customized cnn model and deep learning," in *2023 International Conference on Advances in Intelligent Computing and Applications (AICAPS)*. IEEE, 2023, pp. 1–6.

[7] S. Nandy, M. Adhikari, V. Balasubramanian, V. G. Menon, X. Li, and M. Zakarya, "An intelligent heart disease prediction system based on swarm-artificial neural network," *Neural Computing and Applications*, vol. 35, no. 20, pp. 14 723–14 737, 2023.

[8] J. Gao, J. Liu, Y. Xu, D. Peng, and Z. Wang, "Brain age prediction using the graph neural network based on resting-state functional mri in alzheimer's disease," *Frontiers in Neuroscience*, vol. 17, 2023.

[9] L. Tiwari, V. Awasthi, R. K. Patra, R. Miri, H. Raja, and N. Bhaskar, "Lung cancer detection using deep convolutional neural networks," in *Data Engineering and Intelligent Computing: Proceedings of 5th ICICC 2021, Volume 1*. Springer, 2022, pp. 373–385.

[10] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[11] V. Valls, M. Zayats, and A. Pascale, "Information flow in graph neural networks: A clinical triage use case," in *2023 IEEE International Conference on Digital Health (ICDH)*. IEEE, 2023, pp. 81–87.

[12] M. Zipfl, S. Spickermann, and J. M. Zöllner, "Utilizing hybrid trajectory prediction models to recognize highly interactive traffic scenarios," *arXiv preprint arXiv:2309.06887*, 2023.

[13] X. Hu, Z. Sun, Y. Nian, Y. Dang, F. Li, J. Feng, E. Yu, and C. Tao, "Explainable graph neural network for alzheimer's disease and related dementias risk prediction," *arXiv preprint arXiv:2309.06584*, 2023.

[14] G. Sunil, S. Gowtham, A. Bose, S. Harish, and G. Srinivasa, "Graph neural network and machine learning analysis of functional neuroimaging for understanding schizophrenia," *BMC neuroscience*, vol. 25, no. 1, p. 2, 2024.

[15] C. Jie, C. Jiming, S. Ying, T. Yanchun, and R. Haodong, "A pyramid gnn model for cxr-based covid-19 classification," *The Journal of Supercomputing*, pp. 1–19, 2023.

[16] H. Zhang, D. H. Nguyen, and K. Tsuda, "Differentiable optimization layers enhance gnn-based mitosis detection," *Scientific Reports*, vol. 13, no. 1, p. 14306, 2023.

[17] X.-S. Li, X. Liu, L. Lu, X.-S. Hua, Y. Chi, and K. Xia, "Multiphysical graph neural network (mp-gnn) for covid-19 drug design," *Briefings in Bioinformatics*, vol. 23, no. 4, p. bbac231, 2022.

[18] Z. Guo, K. Guo, B. Nan, Y. Tian, R. G. Iyer, Y. Ma, O. Wiest, X. Zhang, W. Wang, C. Zhang *et al.*, "Graph-based molecular representation learning," *arXiv preprint arXiv:2207.04869*, 2022.

[19] M. Krenn, Q. Ai, S. Barthel, N. Carson, A. Frei, N. C. Frey, P. Friederich, T. Gaudin, A. A. Gayle, K. M. Jablonka *et al.*, "Selfies and the future of molecular string representations," *Patterns*, vol. 3, no. 10, 2022.

[20] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer *et al.*, "Graph neural networks for materials science and chemistry," *Communications Materials*, vol. 3, no. 1, p. 93, 2022.