IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

APPROACH FOR DRUG REPURPOSING USING GRAPH NEUARL NETWORKS

, [1]Dr. Prasantha H S, [2] Kishore U, [3]Deepak U, [4]Ajay

[1]Professor, Dept. of Electronics and Communication Engineering
[2,3,4] Undergraduate Student, Dept. of Electronics and Communication Engineering
Nitte Meenakshi Institute of Technology, Bangalore, India

Abstract: *The objective of drug repurposing is identification of potential drugs and surveilling their in vivo efficacy and effectiveness against a disease. Drug repurposing is associated with several advantages. The risk of failure is low as it uses the drugs which are already been approved. It uses the drugs which has less unknown harmful adverse effects. Since it uses the drugs which have passed all pre-clinical trials and safety standards, the time span of drug development is reduced significantly. Drug repurposing requires less financial investment when compared to the discovery of new drug. It reduces the time frame for drug development as the drugs have passed all the pre-clinical trials and safety norms.*

Index Terms – Covid, Corona Virus, Drug Repurposing, Graph Neural Networks

1. INTRODUCTION

More than 3.9 million people have died as a result of the 2019 novel coronavirus (SARS-CoV-2) pandemic and it has resulted in significant morbidity and economic hardship around the world. According to the June 2020 Global Economic Prospects, they estimated a 5.2% decline in global GDP in 2020 which is the worst economic recession since WWII.COVID-19 is caused by a pathogen that belongs to the Coronaviridae family of enveloped positive-strand RNA viruses that harm mammals, birds, and amphibians[1].

Coronaviruses are classified into four genera which are alpha coronavirus, beta coronavirus, delta coronavirus, and gamma coronavirus. The three beta coronaviruses, middle east respiratory syndrome coronavirus (MERS-CoV), severe acute respiratory syndrome coronavirus (SARS-CoV), and the novel severe acute respiratory syndrome coronavirus (SARS-CoV-2) generate severe symptoms out of the seven known strains of human CoVs (HCoVs).The 2003 SARS-CoV outbreak was suppressed, but it infected 8098 people and killed 774 individuals. Considering SARS-CoV-2 as highly contagious, the World Health Organization (WHO) declared it a public health emergency of global concern on January 30, 2020 and issued a warning to all countries with poor health care systems. As there are no specific known medicines for the 2019 novel Coronavirus, the present treatment for COVID-19 is entirely supportive and symptomatic[1].

Vaccines to prevent SARS-CoV-2 is being developed by several research groups across the world. Given the current unpredictability trajectory of the disease transmission and the virus's life cycle, developing preventive strategies against it is crucial. There is a demanding need to find medicines that can cure and prevent novel diseases such as 2019 coronavirus (COVID-19).Given the tight timeline, drug repurposing or repositioning, which aims to find new medical indications for approved drugs, is a more realistic solution. Drug repurposing is a promising strategy to discover new medical indications of the existing approved drugs due to several advantages in terms of the costs, safety factors, and quick results compared to new drug design and discovery.

2. BACKGROUND WORK

2.1 Limitations of existing system

There are no specific medicines known for COVID-19. Many vaccines have been developed by research groups around the world. But the availability of vaccines and their efficacy are of concern. None of the vaccines are completely effective. Recent mutated variants of virus are unaffected by vaccines. So, the treatment is completely supportive and symptomatic. Given this stringent timeline, the most effective solution is drug repurposing, whose aim is to identify new medical indications of drugs which are already been approved. Drug repurposing is associated with several advantages. The risk of failure is low as it uses the drugs which are already been approved. It uses the drugs which has less unknown harmful adverse effects. Since it uses the drugs which have passed all pre-clinical trials and safety standards, the time span of drug development is reduced significantly. Drug repurposing requires less financial investment when compared to the discovery of new drug.

2.2 Computational Approaches

In computational pharmacology, many common machine learning (ML) and deep learning (DL) techniques have been used. Drug-drug interaction was modelled as a binary classification problem and solved using machine learning (ML) techniques such as random forest, support vector machines (SVM), and naive bayes, as well as deep learning (DL) models such as deep multi-layer perceptrons and recurrent neural networks, to name a few. Standard ML approaches are frequently outperformed by DL techniques. These approaches, on the other hand, are unable to capture the structural information in the data, particularly the links between four biological components (e.g., interactions between drugs and genes or between drugs and diseases). Constructing a graph with nodes representing items such as drugs, genes, diseases, and so on, and edges indicating the intricate interactions between these entities, is a natural and efficient way to describe such structural information. While processing data, graph neural networks (GNNs) collect structural information by accounting for the underlying graph structure. Decagon, a GNN-based model for predicting the side effects of a pair of drugs, has proven its worth by surpassing non-graph-based machine learning algorithms in terms of prediction accuracy [1].

2.3 Drug Repurposing

Drug repurposing procedure comprises of identification of potential drugs and survey their in vivo efficacy and effectiveness against the disease. The most crucial step in this procedure is identification of the right candidate drugs and this considers experimental and computational approaches. The accessibility to large biological data has made computational approaches more attractive field of work. Many opportunities in the field of pharmacology have opened up as a result of more efficient techniques to handle large data.

2.4 Graph Neural Networks

Basic element of Graph Neural Networks is a graph. In Computer Science, a graph is a data structure consisting of two components, vertices and edges. A graph *G* can be well described by the set of vertices *V* and edges *E* it contains. It is given as G = (V, E). Edges can be either directed or undirected, depending on whether there exist directional dependencies between vertices. The vertices are often called nodes.

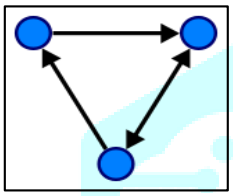

Figure 1: A graph

Graph Neural Network is a type of Neural Network which directly operates on the Graph structure. A typical application of GNN is node classification. GNNs are a kind of data structure which models a set of objects (nodes) and their relationships (edges). Recently, researches on analysing graphs with machine learning have been receiving more and more attention because of the great expressive power of graphs, i.e. graphs can be used as denotation of a large number of systems across various areas including social science, natural science and protein-protein interaction networks, knowledge graphs and many other research areas. As a unique non-Euclidean data structure for machine learning, graph analysis focuses on tasks such as node classification, link prediction, and clustering. Graph neural networks (GNNs) are deep learning based methods that operate on graph domain. Due to its convincing performance, GNN has become a widely applied graph analysis method recently. In the following paragraphs, we will illustrate the fundamental motivations of graph neural networks.

2.5 Challenges and recommendations

Given the high attrition rates, high costs, and slow pace of new drug discovery and development, repurposing of "old" drugs to treat both common and rare diseases is becoming an increasingly appealing proposition because it involves the use of derisked compounds, which could result in lower overall development costs and shorter development timelines. For the identification of re-purposely drug candidates, a variety of data-driven and experimental approaches have been proposed. However, there are significant technological and regulatory hurdles to overcome. This review discusses drug repurposing approaches, as well as the challenges faced by the repurposing community and innovative ways to address these challenges in order to help realize the full potential of drug repurposing. Drug repurposing (also known as drug repositioning, re-profiling, or re-tasking) is a strategy for discovering new uses for approved or investigational drugs that go beyond the original medical indication. This approach has several advantages over developing a completely new drug for a specific indication. With the opportunities and challenges of drug repurposing in mind, the conclusion is reached by presenting six recommendations to help realize drug repurposing's full potential.

First and foremost, better data analysis platforms are required. The advantages of big data and how it can assist with identifying repurposing opportunities are clear.

Second, there is a need for improved access to industry-developed preclinical and clinical compounds

Third, greater access to data from industry-sponsored phase II–IV clinical trials is required. External scientists may be able to mine the data for new findings that could lead to repurposing opportunities, particularly for discontinued programs.

Fourth, newer risks associated with repurposed drugs should be investigated. Any new safety issues associated with repurposed medications must be investigated on a regular basis.

Fifth, more funding opportunities for drug repurposing initiatives are needed, such as funding for appropriate technology, facilitating compound access, and sharing drug repurposing libraries.

Finally, incentives for drug repurposing are needed, particularly to address the patent and regulatory barriers mentioned earlier [2].

2.6 Proposed Methodology

Drug repurposing using graph neural networks involves following steps:
Investigation of data-driven computational approaches for drug repurposing and develop a specialised GNN-based drug repurposing model.
Generation of a four-layered heterogeneous graph, representing the intricate connections between drugs, diseases, genes, and anatomies.
Representing this graph with the help of an adjacency matrix.
Training the machine with this matrix representation of the graph by performing mathematical operations (activation function) on this matrix.
Prediction of drugs for any novel diseases using this model with their symptoms as the input.

3. MODEL

3.1 Data Sets

Deep learning has attracted substantial attention from a range of scientific disciplines in recent years as a result of its amazing performance in numerous difficult problems such as data cleansing, mining, and classification, mostly for images, speech and text datasets. Deep learning for graph-structured data, also known as geometric deep learning (GDL), has recently gained a lot of attention for drug repurposing applications. GDL intends to create graph neural networks (GNNs), which are neural network architectures that can learn from graph-structured data.

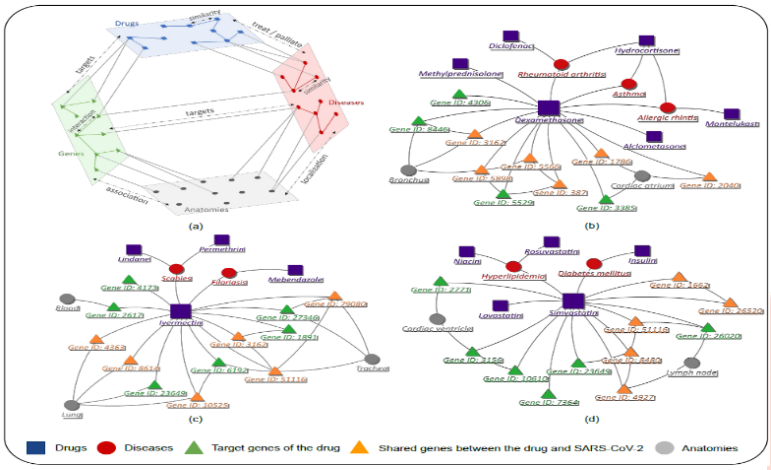

Figure 3.1: Four Layered Graph

As seen in Figure 3.1, this model produces a four-layered heterogeneous graph with these four items in each layer. The required data is derived from a comprehensive biological knowledge graph that connects genes, drugs, diseases, biological processes, side effects, and eight other entities for computational pharmacological tasks like drug repurposing, drug discovery, and drug adverse effect prediction, called DRKG (drug repurposing knowledge graph). DRKG gathers all this information from six databases, namely, Drugbank, Hetionet, GNBR, STRING, IntAct, and DGIdb.

This model, based on DRKG, takes into account four entities that are relevant to the drug repurposing task. The four entities are drugs, diseases, anatomies, and genes. As shown in Figure 3.1, it produces a four-layered heterogeneous graph with these four items in each layer. There are 8070 drugs, 4166 diseases, 29848 genes, 400 anatomies, and a total of 1,417,624 links in the four-layered graph, which includes all inter-layer and intra-layer relationships.

Between the four layers, there are inter-layered connections, and certain layers include intra-layered connections. The inter-layered connections represent connection of different entities. The drug-disease linkages denote treatment or palliation, which means that a drug treats or relieves an illness. One of the reasons for a drug to be a potential repurposing candidate against a disease is that the drug and the disease share gene targets.

3.2 Algorithm

Consider an undirected graph G = (V, E) with a set of vertices V = $\{v_1, v_2, \ldots, v_N\}$ and edges $e_{ij} \in E$ denoting a connection between nodes v_i and v_j. The graph G is represented using the adjacency matrix $A \in R^{N \times N}$, where the (i , j)th entry of A denoted by a_{ij}. If there is an edge between nodes v_i and v_j, the value of a_{ij} is 1, otherwise it is zero. Using the normalized adjacency matrix, the non-uniformity in the degrees of the nodes can be accounted. The normalized adjacency matrix is given by $\widetilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $D \in R^{N \times N}$ is the diagonal degree matrix. In the graph, each node will be having its own feature vector (input feature). The input feature of node i is denoted by $\mathbf{x_i^{(0)}} \in R^d$, which holds the node's important information or attributes. Let $X^{(0)} \in R^{N \times d}$ denote the input feature matrix for the N nodes in the graph G, which is created by stacking the input characteristics of all nodes in G. To account for local interactions, new embeddings for a node are generated by merging information from its nearby nodes. A single GNN block handles the process of integrating input and creating new representations for a node. Information from its K-hop neighbors can be incorporated when K such blocks are stacked. This operation is mathematically represented as $X^{(k+1)} = g_k(\widetilde{\mathbf{A}}\, X^{(k)} W_k)$; where $X^{(k)} \in R^{N \times dk}$denotes the embedding matrix in k-th layer and dk is the embedding dimension of the k-th layer. Here, $\widetilde{\mathbf{A}} = I + \widetilde{\boldsymbol{A}}$, where $I \in R^{N \times N}$ is the identity matrix, added in order to account for the self-node embedding's, $W_k \in \mathbf{R^{d_k \times d_{k+1}}}$ is the learnable transformation matrix, and $g_k(.)$ is the activation Function used in the kth layer.

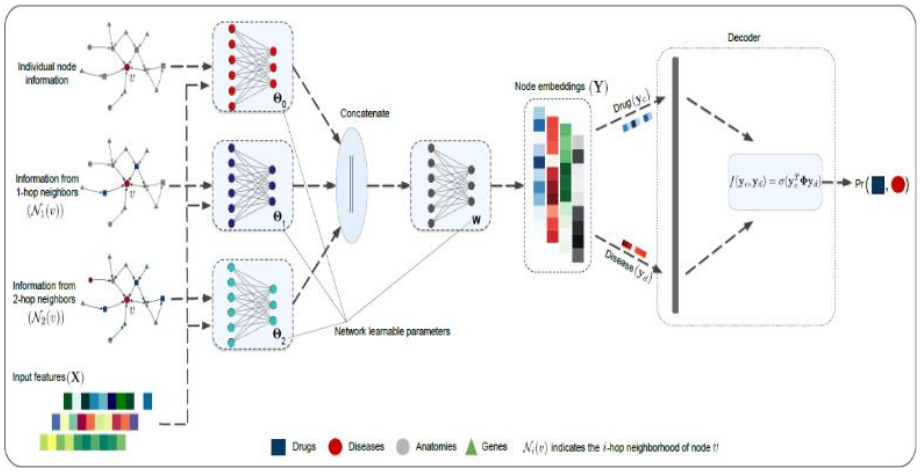

Figure 3.2 Architecture

3.3 Components

The encoder and decoder are the two primary components of the proposed GNN architecture for SARS-CoV-2 drug repurposing. All nodes in the four-layer graph have their node embeddings generated by the encoder which is based on the SIGN architecture. The drug-disease pair is then given a score by the decoder based on embeddings. End-to-end training is done on the encoder and decoder networks.

Encoder: This encoder is based on the SIGN architecture (Scalable Inceptive Graph Neural Network), which generates low-dimensional node embeddings based on input attributes and nodal connection information. Linear diffusion operators are used by SIGN to perform message passing and accumulate local information in the graph, which are represented using matrices F_r, r = 1, 2, ... The information from r-hop neighbors for a node can be incorporated by choosing $F_r = \widetilde{\boldsymbol{A}}^{\boldsymbol{r}}$. Here, $\widetilde{\boldsymbol{A}}^{\boldsymbol{r}}$ denotes the r^{th} power of the matrix $\widetilde{\boldsymbol{A}}$. The encoder can therefore be written as $Z = \sigma_1 \{[X\Theta_0 \| F_1X\Theta_1 \| . . . \| F_rX\Theta_r]\}$ and $Y = \sigma_2\{ZW\}$; where Y is the final node embedding matrix for the nodes in the graph G , and $\{\Theta_0, . . . , \Theta_r, W\}$ are the learnable parameters. Here, || is concatenation operation and $\sigma_1\{.\}$ and $\sigma_2\{.\}$ are activation functions, which are, nonlinear tanh and leaky rectified linear unit (leaky ReLU) respectively. It is better to choose r = 2, i.e., the low-dimensional node embeddings have information from 2-hop neighbors. It was observed that choosing $\mathbf{r \geq 3}$ was not useful for drug repurposing, because the goal is to capture local information about drug targets in such a way that a drug node embedding should keep track of its target genes as well as shared genes in its neighbourhood.

Decoder: This model presents a score function for drug repurposing that takes the embeddings of drugs and diseases as input and returns a score that determines if a drug treats the disease. The embeddings of all nodes in the four-layer network, including the disease and drug nodes, are contained in the columns of the embedding matrix Y. The embeddings of the i^{th} drug is denoted as $\mathbf{y_{c_i}} \in \mathbf{R}^l$ and the embeddings of the j^{th} disease as $\mathbf{y_{d_j}} \in \mathbf{R}^l$. $\mathbf{s_{ij}}$ is the proposed scoring function f (.) for determining whether drug c_i is a promising treatment for disease d_j, and it is defined as $\mathbf{s_{ij}} = \mathbf{f}(\mathbf{y_{c_i}}, \mathbf{y_{dj}}) = \boldsymbol{\sigma}\left\{\mathbf{y_{c_i}^T \Phi y_{d_j}}\right\}$; where $\boldsymbol{\sigma}\{.\}$ is the nonlinear sigmoid activation function and $\boldsymbol{\Phi} \in R^{l \times l}$ is a learnable co-efficient matrix. $\mathbf{s_{ij}}$can be interpreted as the likelihood of a link between drug c_i and sickness d_j. For the sample corresponding to the drug-disease pair, the loss function is given as $l(s_{ij}, z_{ij}) = \mathbf{wz_{ij}}\left(\log\left(\frac{1}{\sigma(\mathbf{s_{ij}})}\right)\right) + (1 - z_{ij}) \log\left(\frac{1}{1-\sigma(\mathbf{s_{ij}})}\right)$; where z_{ij} is the known training label associated with score s_{ij} for the drug-disease pair. When $z_{ij} = 1$ it means that drug i treats disease j, and when $z_{ij} = 0$ indicates that drug i does not treat disease j. To account for the class imbalance, weight on the positive samples, w, is chosen

4. SOFTWARE DETAILS

Anaconda 3: Anaconda is an open-source software that includes Jupyter, spyder, and other tools for big data processing, data analytics, heavy scientific computing, and machine learning, etc. Anaconda platform works for R and Python programming language. Anaconda Inc., created by Peter Wang and Travis Oliphant in 2012, is responsible for its development and maintenance. Data-science packages for Windows, Linux, and MacOS are included in the release. Python is run using Spyder (sub-application of Anaconda). Opencv is used for python in spyder. Conda, the package management system, manages the package versions.

4.1 System Requirements

Hardware Requirements: CPU: 2 x 64-bit 2.8 GHz, RAM: 8 GB DDR4, Minimum 3 GB disk space to download and install.
Software Requiremenys: Windows 8 or newer, 64-bit MacOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 6+ operating system. Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bit aarch64 (AWS Graviton2 / arm64), 64-bit Power8/Power9, s390x (Linux on IBM Z &LinuxONE) system architecture.

4.2 Packages and libraries used

PyTorch: PyTorch is an open-source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing. primarily developed by Facebook's AI Research lab (FAIR). It is an open-source software distributed under the Modified BSD licence. PyTorch also provides a C++ interface, although the Python interface is more refined and the primary focus of development.

Command to install PyTorch: *"condainsta llpytorchtorchvisiontorchaudiocudatoolkit=10.2 -c pytorch"*

Numpy: NumPy is a Python library that adds support for huge, multi-dimensional arrays and matrices, as well as a large number of high-level mathematical functions to operate on these arrays. Numeric, NumPy's predecessor, was built by Jim Hugunin with contributions from several other developers. Travis Oliphant built NumPy in 2005 by heavily modifying Numeric and combining features from the competitor Numarray. NumPy is open-source software and has many contributors.
Command to install Numpy: *"conda install -c anaconda numpy"*.

Pickle: The pickle module implements binary serialisation and de-serialization protocols for Python object structures. *"Pickling"* is the process whereby a Python object hierarchy is converted into a byte stream, and *"unpickling"* is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy.
Command to install Pickle: *"conda install -c conda-forge pickle5"*.

Openpyxl: Openpyxl is a Python library for reading and writing Excel (with extension xlsx/xlsm/xltx/xltm) files. The openpyxl module allows Python program to read and modify Excel files. For example, users might have to go through thousands of rows and pick out a few handful of information to make small changes based on some criteria. Using Openpyxl module, these tasks can be done very efficiently and easily.
Command to install Openpyxl: *"conda install -c anaconda openpyxl"*.

RE: This module allows you to match regular expressions. Patterns and strings to be searched can be both Unicode and 8-bit strings (str) (bytes). A regular expression (also known as a RE) describes a set of strings that match it; the functions in this module can be used to see if a string matches a regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).
Command to install RE: *"conda install -c conda-forge regex"*.

5. RESUTLS

The model requires MeSH ID as input to predict drugs for the treatment of the disease. MeSH stands for Medical Subject Headings and is a regulated vocabulary for indexing journal articles and books in the life sciences. It's a thesaurus that makes searching easier. It is used by the MEDLINE/PubMed article database and the NLM's catalogue of book holdings. The United States National Library of Medicine (NLM) has created this thesaurus and it is responsible for updating the same.

The MeSH ID of COVID-19 is SARS-CoV-2. The top 30 predicted drugs for SARS-CoV-2 by the model are: Cidofovir, Yellow Fever Vaccine, Rasburicase, Rubella virus vaccine, Ataluren, Repaglinide, Diazoxide, Chlormerodrin, Hydralazine, Sirolimus, Glyburide, Onasemnogeneabeparvovec, Deflazacort, Fluvastatin, Bromocriptine, Adefovir, Tetrachlorodecaoxide, Chlorothiazide, Furosemide, Hydrochlorothiazide, Adefovirdipivoxil, Rosuvastatin, Phenformin, Simvastatin, Saquinavir, Ganciclovir, Acyclovir, Spironolactone and Etacrynic acid.

The model also predicts the rank of a particular drug. The drug "Bromocriptine" was given 16th rank by the model.

The MeSH ID of Malaria is Disease::MESH:D008288. The top 30 drugs predicted for Malaria by the model are: Tetracycline,Clindamycin, Doxycycline, Metronidazole, Minocycline, Ivermectin, Chloroquine, Rifapentine, Erythromycin, Proguanil, Sulfadiazine, Dapsone, Clarithromycin, Rifabutin, Trimethoprim, Primaquine, Praziquantel, Demeclocycline, Atovaquone, Sulfamethoxazole, Terbinafine, Rifaximin, Rifampicin, Loperamide, Hydroxychloroquine, Telithromycin, Ketoconazole, Esomeprazole, Cyclosporine, Roxithromycin.

The drug "Hydroxychloroquine" was given 25th rank by the model.

The MeSH ID of Common Cold is Disease::MESH:D003139. The top 30 drugs predicted for Common cold by the model are: Diphenhydramine, Montelukast, Dimenhydrinate, Hyoscyamine, Promethazine, Carbinoxamine, Loratadine, Levomenthol, Atropine, Camphor, Capsaicin, Glycopyrronium, Chlorpheniramine, Triprolidine, Dexchlorpheniramine maleate, Pheniramine, Ipratropium, Rupatadine, Flavoxate, Doxylamine, Zinc, Chlorcyclizine, Dyclonine, Amitriptyline, Mepyramine, Doxepin, Tiotropium, Fexofenadine, Desloratadine, Cetirizine.

The drug "Zinc" was given 21st rank by the model.

The MeSH ID of Dengue Fever is Disease::MESH:D003715. The top 30 drugs predicted for Dengue Fever by the model are: Rimantadine, Rilpivirine, Enfuvirtide, Bedaquiline, Praziquantel, Rifapentine, Rifaximin, Letermovir, Darunavir, Rifabutin, Famciclovir, Ketoconazole, Oseltamivir, Sebelipasealfa, Bacillus calmette-guerinsubstrain tice live antigen, Clarithromycin, Elvitegravir, Quinupristin, Cobicistat, Fosamprenavir, Terbinafine, Fusidic acid, D-glucose, Clindamycin, Tafenoquine, Aminophylline, Lopinavir, Acyclovir, Glycerol phenylbutyrate, Fluconazole.

The drug "D-glucose" was given 23rd rank by the model.

The MeSH ID of Diabetes is Disease::MESH:D003924. The top 30 drugs predicted for Diabetes by the model are: Vildagliptin, Repaglinide, Glipizide, Glyburide, Canagliflozin, Insulin aspart, Liraglutide, Insulin glargine, Chlorpropamide, Insulin glulisine, Tolazamide, Saxagliptin, Spironolactone, Niacin, Telmisartan, Insulin human, Tolbutamide, Cholestyramine, Exenatide, Linagliptin, Glycyrrhizic acid, Gliclazide, Troglitazone, Semaglutide, Rosuvastatin, Nateglinide, Insulin lispro, Sitagliptin, Atorvastatin, Valsartan.

The drug “Insulin Glargine” was given 8th rank by the model.

The MeSH ID of Tuberculosis is Disease::MESH:D014376. The top 30 drugs predicted for Tuberculosis by the model are: Prednisolone, Prednisone, Gemifloxacin, Moxifloxacin, Methylprednisolone, Triamcinolone, Ciprofloxacin, Dexamethasone, Cortisone acetate, Hydrocortisone, Ofloxacin, Trovafloxacin, Ethambutol, Benzylpenicillin, Sulfadiazine, Betamethasone, Daptomycin, Streptomycin, Besifloxacin, Erythromycin, Norfloxacin, Gentamicin, Ertapenem, Nitrofurantoin, Azithromycin, Clindamycin, Metronidazole, Amphotericin B, Chloramphenicol, Bupivacaine.

The drug “Azithromycin” was given 8th rank by the model.

V. CONCLUSIONS AND FUTURE WORK

This paper presents a broad drug repurposing model for unique human diseases. The drug repurposing job is formulated as a link prediction problem using a biological network of drugs, diseases, genes, and anatomies. A graph neural network model was used to predict medications for novel diseases once it was trained. This model has predicted 150 effective drugs for COVID-19, of which 46 drugs are currently being used for the treatment. The considered GNN model is computationally efficient and outperforms other GNN variations and non-deep methods such as network proximity approaches in ranking known therapeutic drugs for disorders.

This work can be expanded in a number of ways. Given the abundance of biological data, the addition of information such as individual drug side effects, drug chemical structure, and so on, could improve predictions even more. Considering a patient's comorbidities can help us analyse the biological processes and gene interactions in the body that are unique to that person and prescribe line of treatment accordingly. Another area where graph neural networks could be useful is predicting a synergistic combination of medications for a condition.

This model has been designed using Leaky ReLU and Sigmoid activation functions. Another area of interest would be the use of different activation functions and comapring the performance. Also comparing the performance of Graph Neural Network with that of Convolution Neural Network is of great interest. Further, graphical user interface can be incorporated to generate heat map to indicate the rank or score of different drugs for several diseases together.

VI. REFERENCES

1. S. Doshi and S. P. Chepuri, "Dr-COVID: Graph Neural Networks for SARS-CoV-2 Drug Repurposing," *Cornell Universiy*, 2020
2. S. Pushpakom, F. Iorio, P. A. Eyers, K. J. Escott, S. Hopper, A. Wells and e. a. , "Drug repurposing: progress, challenges and recommendations," *Nature Reviews Drug Discovery*, 2019
3. C. I. Paules, H. D. Marston and A. S. Fauci, "Coronavirus Infections—More Than Just the Common Cold," *The Journal of the American Medical Association*, 2020.
4. N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong, Y. Han and e. a. , "Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study," *Lancet*, 2020.
5. A. Zumla, F. W. J. Chan, E. I. Azhar, D. S. C. Hui and K. Y. Yuen, "Coronaviruses — drug discovery and therapeutic options," *Nature Reviews Drug Discovery*, 2016.
6. M. Zitnik, M. Agrawal and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, 2018.
7. Dr. Prasantha H S, Ajay S, Deepak U, Kishore U, "Literature review for SARS-CoV-2 Drug Repurposing", IJCRT, 2021.
8. Dr. H S Prasantha, “NOVEL APPROACH FOR IMAGE COMPRESSION USING MODIFIED SVD”, International Journal of Creative Research Thoughts (IJCRT), Volume 8, Issue 8, Page 2234-2243, Aug 2020
9. Dr. H S Prasantha, “IMPLEMENTATION OF IMAGE COMPRESSION USING FAST COMPUTATION OF SVD ON DM642”, International Journal of Creative Research Thoughts (IJCRT), Volume 8, Issue 8, Page 2364-2368, Aug 2020
10. Prasantha, H, H Shashidhara, K N B Murthy, and M Venkatesh. "Performance Evaluation of H.264 Decoder on Different Processors." International Journal on Computer Science & Engineering. 1.5 (2010): 1768. Web. 7 Apr. 2013.
11. Chirag B., A. Lohith and Prashantha H. S. , “Comparative performance analysis of various digital modulation schemes in AWGN channel,” in Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 2017, pp. 1-5
12. M. J. Raghavendra, H. S. Prasantha and S. Sandya, "Image Compression Using Hybrid Combinations of DCT SVD and RLE", International Journal of Computer Techniques, vol. 2, no. 5, 2015.
13. M.J. Raghavendra, H. S. Prasantha and S. Sandya, "DCT SVD Based Hybrid Transform Coding for Image Compression", International Journal of Recent and Innovative Trends in computing and communication, 2015.
14. H. S. Prasantha, H. L. Shashidhara, and K. N. Balasubramanya Murthy. Image compression using SVD. In Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, pages 143–145. IEEE Computer Society, 2007.
15. Gunasheela K S, H S Prasantha, “Compressive sensing for image compression: survey of algorithms”, Proceedings of Emerging Research in Computing, Information, Communication and Applications, ERCICA, Springer publication, Bengaluru, 2018

16. K N Shruthi, B M Shashank, Y. SaiKrishna Saketh, H.S Prasantha and S. Sandya, "Comparison Analysis Of A Biomedical Image For Compression Using Various Transform Coding Techniques", IEEE, pp. 297-303, 2016