Utilizing Artificial Neural Networks to Identify Latent Network Pathways in Psychiatric Comorbidity

Georgia Smith

CSC/Math 4990, The University of Virginia's College at Wise
Advisor: Dr. James Vance

April 5, 2021

Contents

Chapter 1

Introduction

1.1 Overview

1.1.1 Introduction

Mental illness can be a burden for those whose lives are impacted from it, especially those with serious mental illness. This burden is exasperated when multiple mental illnesses co-occur - a phenomenon known as psychiatric comorbidity (further referred to as comorbidity).

Persons with serious mental illness are more likely to have comorbid medical conditions than those in the general population (Sokal et al., 2004). There also have been multiple studies identifying a high comorbidity between drug abuse and mental illness ((Volkow, 2001), (Regier et al., 1990), (Kokkevi and Stefanis, 1995)).

While the identification of illnesses commonly co-occuring is important for diagnosis and treatment, much is unknown about why diseases co-occur and what is causing them. The discovery of these causal factors and predictive features would have vast implications in the diagnosis and treatment of comorbid mental illness.

1.1.2 Latent versus Network Analysis

Recently, the field of psychometric analysis, or the analysis of psychological measurement techniques and theories, has had two major ideas of how two analyze comorbidity: a latent variable perspective and a network perspective. A latent variable perspective, as laid out by Boorsboom (Borsboom, 2008), is a combination of observed variables which allow inference of a latent variable.

The current Diagnostic Statistical Model of Mental Disorders, DSM-V, is considered a latent variable approach to mental illness. For diagnosis of a disorder (the latent variable) certain observable criteria must be met which show as symptoms. The latent variable, or disorder, is not just a label for when symptoms occur in the latent variable perspective. The latent variable is an unknown causal factor of the disorder, such as lack of serotonin and depression.

The network perspective does not discount the DSM-V, but instead does not focus on an unobservable causal factor. Instead, a diagnosis is caused and defined by an interactive *network* of observable symptoms. The network perspective was originally laid out by Cramer et. al in 2010 (Cramer et al., 2010) using Alegria and Kessler's National Comorbidity Survey - Replication (NCS-R) (Kessler et al., 2004).

Using the NCS-R, Kessler et. al. proposed a radical new way to conceptualize metal disorders and naturally leads to more complex network analysis. A main finding of Kessler et. al. was the discovery of ”bridge symptoms,” which are symptoms that occur in multiple disorders connecting them.

1.2 Goal

We hope to extend Cramer et. al.'s (Cramer et al., 2010) initial hypothesis of the usability of network analysis to latent network path analysis, identifying pathways of comorbidity to identify causal symptoms of disorders. We will do this in a multi-stage production.

First, we will develop networks of individuals in the NCS-R building directed graphs based on the onset age of symptoms. Then, we will combine all of the networks for individuals with a certain diagnosis. After generating graphs for all of the diagnoses in Table A.10 we will identify cases of comorbidity and combine the networks of individuals with comorbidity using the Deep Graph Library (DGL) (Wang et al., 2019). Finally, will used GraphSAGE (Hamilton et al., 2017) and the DGL to implement convolutional and long-short-term-memory (LSTM) Neural Networks to identify latent paths in the comorbid dataset.

1.3 Data

We will utilize the data from the NCS-R (Kessler et al., 2004). The NCS-R is comprised of 3,713 variables which represent questions from a questionnaire designed to aid in diagnosis based on the DSM-V. The questionnaire is split into 46 sections with questions specific to specific disorders - such as depression, post-traumatic stress disorder (PTSD), and dementia - as well as demographics and lifetime events.

The NCS-R is a representative survey of adults (age 18+) in the United States, comprised of 9,282 subjects. The NCS-R was completed in the homes of participants and all participants that met criteria for a mental disorder were issued a second, more in-depth interview (Kessler et al., 2004).

The demographics of the NCS-R can be seen in Tables 1.1 - 1.4. All of the DSM-IV Diagnosis can be seen in Table A.10 in the appendix.

Age	**18-29**	**30-44**	**45-59**	**60+**
Percent	22.7	31.7	24.6	21.0

Table 1.1: Age Distribution of the NCS-R (Kessler et al., 2004)

Sex	**M**ale	**F**emale
Percent	44.6	55.4

Table 1.2: Sex Distribution of the NCS-R (Kessler et al., 2004)

Race	**N**on-**H**ispanic **W**hite	**N**on-**H**ispanic **B**lack	**H**ispanic	**O**ther
Percent	72.1	13.3	9.5	5.1

Table 1.3: Race Distribution of the NCS-R (Kessler et al., 2004)

Region	**N**ortheast	**M**idwest	**S**outh	**W**est
Percent	18.4	26.7	34.5	20.5

Table 1.4: Regional Distribution of the NCS-R (Kessler et al., 2004)

1.4 Contribution

While a network analysis approach to comorbidity has been investigated before, minimal work has been done to identify the direction of the edges between symptoms to identify causal symptoms. Main contributions to the field include:

- Latent Network Path Analysis on Comorbidity Data
- Neural Networks used in on the Psychometric Network Approach
- Identification of Causal Symptoms using Network Analysis
- Use of Graph Neural Networks on Comorbidiy Data

Chapter 2

Literature Review

2.1 Cramer et al. (2010)

Cramer et. al. sought to rethink the idea of causal relationships in psychiatry in their 2010 paper. They advocated for a network perspective that says "disorders are *networks* that consist of *symptoms* and *causal* relations between them." (Cramer et al., 2010)

A cornerstone of Cramer et. al.'s argument is an assumption that the latent variable model cannot include the cyclic networks which support the causal relationships between symptoms (i.e. you're anxious and trying not to be which makes you more anxious) (Cramer et al., 2010). Danks et. al. disagreed with this assumption, arguing instead that we can define the latent variable model to do this and assuming inability to do so limits possibilities (Danks et al., 2010).

In this paper we will propose a latent-network hybrid where a network model is used to identify latent, directed pathways in comorbidity.

Cramer et. al. focused on the comorbid (or co-occuring) relationship between Major Depression Disorder (MDD) and Genral Anxiety Disorder (GAD). The began with the theory of complex networks "without assuming *a priori* that scuh relationships arise from a mental disorder as a common cause" (Cramer et al., 2010). Then, Cramer et. al. put symptoms into nodes and created paths to represent the relation between symptoms.

They used statistical parameterization and the Akaike Information Criteron to find the most accurate model. They found that a bridge model holds when there are no independent variables. A bridge model is a undirected graph where overlapping nodes (symptoms) indicate a comorbid relationship (see Figure 2.1)

(Cramer et al., 2010).

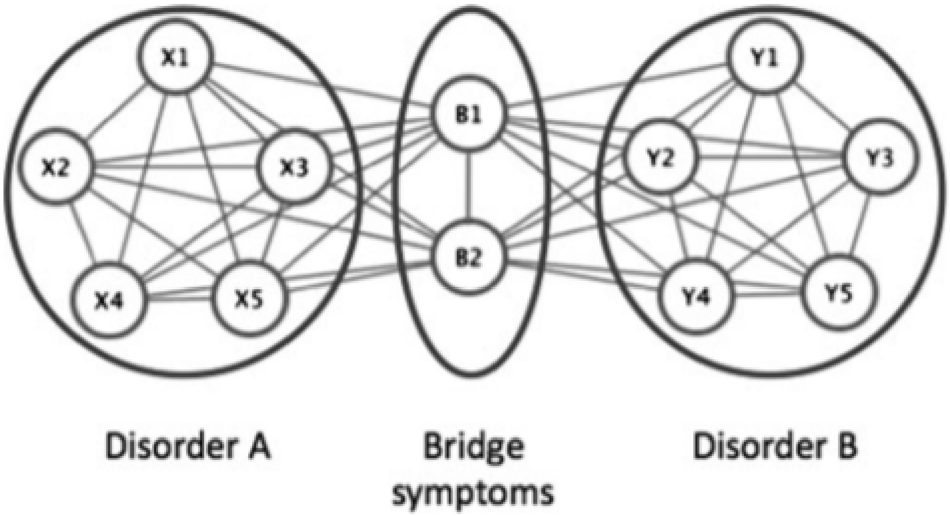


Figure 2.1: Example of a Bridged Network (Cramer et al., 2010)

Using this bridge model, Cramer et. al. used edge thickness and color to further demonstrate relationships between symptoms. The edge thickness represented the co-occurrence of the two symptoms and the edge color represented the strength or association, or log odds ratio, between the two symptoms. The node size was used to exemplify frequency while the color is the node strength, or the sum of the weights of the connected edges (Cramer et al., 2010). The outcome of these additions can be seen in Figure 2.2.

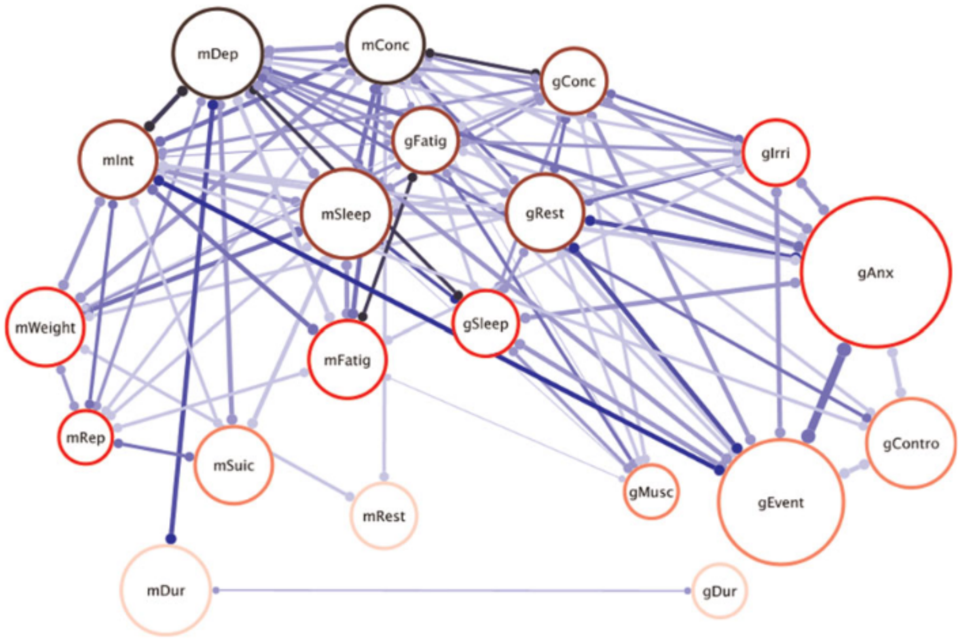


Figure 2.2: Example of a Fully Realized Network using GAD and MDD (Cramer et al., 2010)

Cramer et. al. found that a network approach can be very useful. First, it can aid in hypothesis development on the cause of psychiatric disorders and specific symptoms. Secondly, the can identify ”pathways of comorbidity” that can help lead to identifying some root cause of symptoms (i.e. MDD turning into GAD or *vice versa*). They also found that symptoms were more strongly connected when there was at least one pair of overlapping symptoms (Cramer et al., 2010).

Overall, Cramer et. al. found that a network approach to psychometric analysis is a potentially groundbreaking approach to the analysis of comorbidity and can lead to discovery about the causal relationship between symptoms (Cramer et al., 2010).

2.2 Gomez-Rodriguez et al. (2012)

Network diffusion and latent networks are most commonly talked about today with social media and is a burgeoning field. Gomez-Rodriguez proposed a novel approach in 2012 to identifying latent networks where events are only influenced by the most recent prior events (Gomez-Rodriguez et al., 2012).

2.3 Qian et al. (2020)

Qian et. al. published a paper in 2020 identifying latent pathways in comorbidity using a dataset comprised of colorectal cancer patients. Qian et. al. built on current point process models to develop a new process - Deep Diffusion Process (DDP) (Qian et al., 2020).

DDP was realized using a continuous-time recurrent neural network (RNN) using sigmoid normalization and long short term memory. These methods allow for not only the identification of network pathways, but also judge the probability of future event given a different event has happened (Qian et al., 2020).

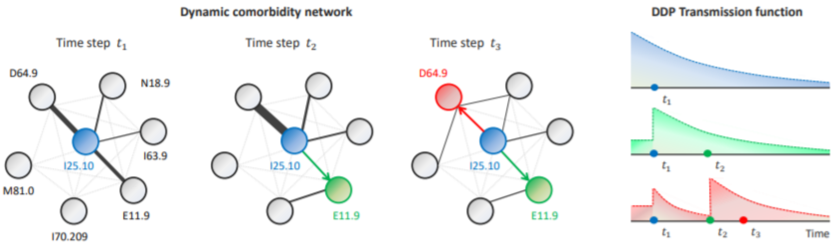


Figure 2.3: Qian et. al.'s realization of DDP. Nodes are identified by their ICD-10 code (Qian et al., 2020)

Figure 2.3 shows the impact DDP can have. The onset of heart disease is at t_1, which increases the possibility of anemia, diabetes, renal failure, and cerebral infarction. When diabetes onset begins at timestep t_2 that increases the probability of the illnesses from t_1, and creates a connection to atherosclerosis.

While we do not utilize DDP due to it's minimal documentation, it proves that the use of ANNs with comorbidity data is a viable approach.

2.4 Graph Neural Networks

Graph Neural Networks (GNNs) were originally proposed by Scarselli et. al. in 2008. Their goal, as stated in their paper, was to ”encode the underlying graph structured data using the topological relationships among the nodes of thegraph, in order to incorporate graph structured information in the data processing step”

(Scarselli et al., 2009). They do this by utilizing recursive neural networks (RNNs) and Markov chains.

Wang et. al., in 2019, compiled recent literature and applications of GNNs and developed a Python library called Deep Graph Library (Wang et al., 2019). This library is centered on the unification of multiple GNN variants using Gilmer and Battaglia's message passing paradigm (Sanchez-Gonzalez et al., 2018), (Battaglia et al., 2018)). We outline this in Equation 2.1 where $G(V,E)$ is a graph with nodes V and edges E. The node v has the features $x_v \in \mathbb{R}^{n_1}$ and edge (u,v) has features $w_{(u,v)} \in \mathbb{R}^{n_2}$.

$$\begin{aligned} \text{Edge-wise} : \mathbf{m}^{(t+1)}_{(u,v)} &= \phi(\mathrm{x}_v^{(t)}, \mathrm{x}_v^{(t)}, \mathrm{x}_{(u,v)}^{(t)}), (u,v) \in E \\ \text{Node-wise} : x^{(t+1)}_{(u,v)} &= \psi(\mathrm{x}_v^{(t)}, \rho(\{\mathbf{m}^{(t+1)}_{(u,v)} : (u,v) \in E\}) \end{aligned} \tag{2.1}$$

We will utilize a specific module in the DGL which is an implementation of GraphSAGE, which applies low-dimensional vector embedding of nodes in a graph for use with GNNs. This was originially developed by Hamilton et. al. and applies the Weisfeiler-Lehman Isomorphism Test to graphs to find isomorphic subgraphs (Hamilton et al., 2017).

Chapter 3

System Analysis and Design

3.1 Implementation Platform

We used multiple platforms in the development of this project. They code is written in Python version 3.8.5. Multiple IDEs were used including Visual Studio Code, VIM, and Jupyter Notebook. We also utilized University of Virginia's Rivanna High Performance Computing (HPC) cluster and acknowledge Research Computing at The University of Virginia for providing computational resources and technical support that have contributed to the results reported within this publication (https://rc.virginia.edu).

3.1.1 Programming Languages

Python was chosen as the primary programming language of this project due to its touted versatility (Prendergast, 2019), our prior expertise in the language, and because it Python is the language used in multiple prior works on latent networks providing us libraries and modules to utilize.

3.1.2 Description of Programming Environment

We first developed code in Jupyter Notebook for testing and debugging. Once the code was working satisfactorily we merged it into a Python file using Visual Studio Code. This code was then either run on a local machine or exported to the Rivanna HPC cluster, depending on computational intensity of the code. We then took code output, in the form of a CSV or serialized with Pickle, and imported it into Jupyter Notebook for further analysis and visualization develpment.

Visualizations were developed using the NetworkX (Hagberg et al., 2008) and MatPlotLib (Hunter, 2007) Libraries.

3.2 Design Overview

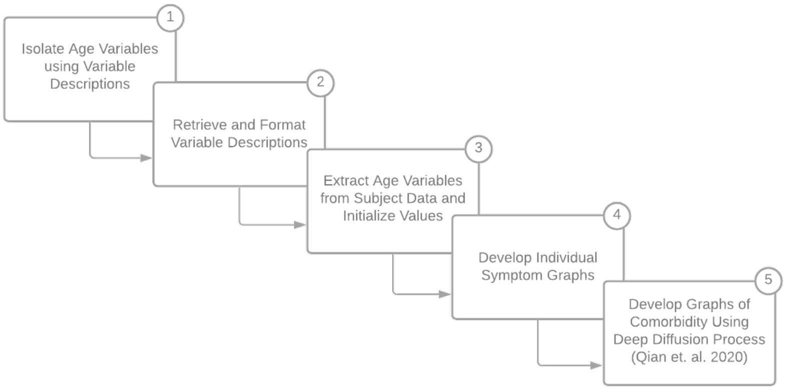

1
Isolate Age Variables using Variable Descriptions
2
Retrieve and Format Variable Descriptions
3
Extract Age Variables from Subject Data and Initialize Values
4
Develop Individual Symptom Graphs
5
Develop Graphs of Comorbidity Using Deep Diffusion Process (Qian et. al. 2020)

Figure 3.1: System Design Overview

We use a multi-staged process for our system outlined in Figure 3.1. We begin by processing the data to initialize the values and extract relevant information. Then, we develop graphs to describe individual subjects. We finish by implementing Deep Diffusion Process (DDP) (Qian et al., 2020).

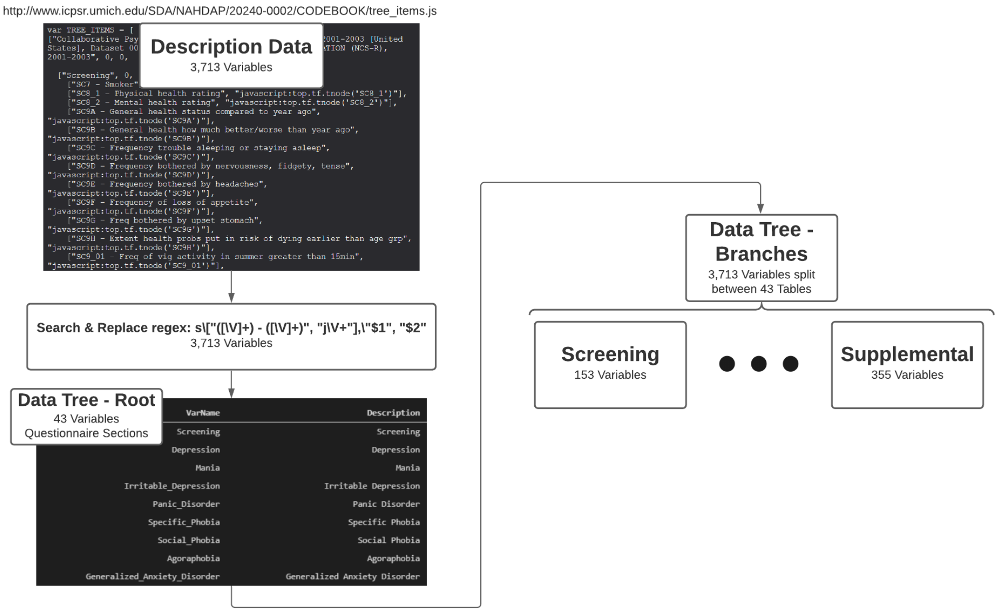
3.3 Input

The code we developed only requires Kessler's NCS-R dataset (Kessler et al., 2004). For comorbidity analysis two DSM-IV diagnosis listed in Table A.10 need to be selected. Analyses can also be run on single diagnosis instead of comorbid by selecting on diagnosis from Table A.10.

3.4 Data Preprocessing

3.4.1 Data Descriptions

The first step in our software design is to get the descriptions for the variables in the NCS-R dataset. As described in section 1.3 there are 3,714 variables in the NCS-R. The NCS-R is hosted on the Inter-university Consortium for Political and Social Research (ICPSR) at University of Michigan. ICPSR includes a section using the Survey Documentation and Analysis (SDA) package which allows users greater insight into the data including descriptions for each variable. This information can also be accessed as a JavaScript array containing all variables and their description or queried by variable to access JSON objects.



http://www.icpsr.umich.edu/SDA/NAHDAP/20240-0002/CODEBOOK/tree_items.js
Description Data
3,713 Variables
Search & Replace regex: s\["([\V]+) - ([\V]+)", "j\V+"],\"$1", "$2"
3,713 Variables
Data Tree - Root
43 Variables
Questionnaire Sections
Data Tree - Branches
3,713 Variables split between 43 Tables
Screening
153 Variables
● ● ●
Supplemental
355 Variables

```
var TREE_ITEMS = [
["Collaborative Psy                              2001-2003 [United
States], Dataset 00                              ATION (NCS-R),
2001-2003", 0, 0,

  ["Screening", 0,
    ["SC7 - Smoker"
    ["SC8_1 - Physical health rating", "javascript:top.tf.tnode('SC8_1')"],
    ["SC8_2 - Mental health rating", "javascript:top.tf.tnode('SC8_2')"],
    ["SC9A - General health status compared to year ago",
"javascript:top.tf.tnode('SC9A')"],
    ["SC9B - General health how much better/worse than year ago",
"javascript:top.tf.tnode('SC9B')"],
    ["SC9C - Frequency trouble sleeping or staying asleep",
"javascript:top.tf.tnode('SC9C')"],
    ["SC9D - Frequency bothered by nervousness, fidgety, tense",
"javascript:top.tf.tnode('SC9D')"],
    ["SC9E - Frequency bothered by headaches",
"javascript:top.tf.tnode('SC9E')"],
    ["SC9F - Frequency of loss of appetite",
"javascript:top.tf.tnode('SC9F')"],
    ["SC9G - Freq bothered by upset stomach",
"javascript:top.tf.tnode('SC9G')"],
    ["SC9H - Extent health probs put in risk of dying earlier than age grp",
"javascript:top.tf.tnode('SC9H')"],
    ["SC9_01 - Freq of vig activity in summer greater than 15min",
"javascript:top.tf.tnode('SC9_01')"],
```

VarName	Description
Screening	Screening
Depression	Depression
Mania	Mania
Irritable_Depression	Irritable Depression
Panic_Disorder	Panic Disorder
Specific_Phobia	Specific Phobia
Social_Phobia	Social Phobia
Agoraphobia	Agoraphobia
Generalized_Anxiety_Disorder	Generalized Anxiety Disorder

Figure 3.2: Process Used to Build the NCS-R Data Description Tree

The JavaScript array describing variables in the NCS-R dataset is split into 43 sections mirroring the questionnaire given to subjects. This array is accessible at https://www.icpsr.umich.edu/SDA/NAHDAP/20240-0002/CODEBOOK/tree_items.js when logged in with an ICPSR account.

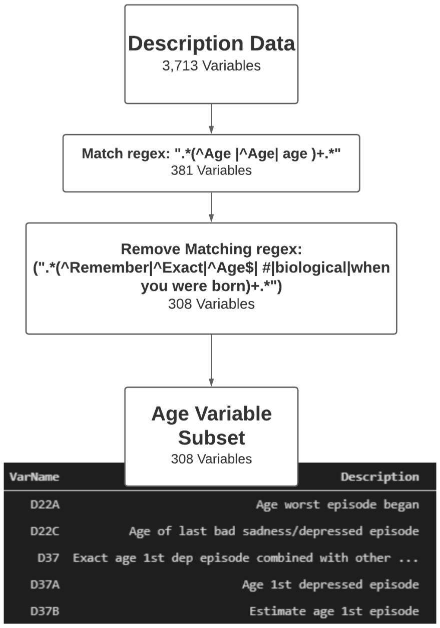
We used the JavaScript array to build a data tree so that we could easily access data descriptions. The process used to build the tree is outlined in Figure 3.2.

The original JavaScript array includes extraneous JavaScript data and combines the variable name and description into a single string. We applied the regular expression *[”([\V]+) - ([\V]+)”, ”j[\V+”],* which was used to search and replace with *[”$1”, ”$2”]*, resulting in the format *[”Variable Name”, ”Variable Description”].*

We also scanned through the array multiple times and checked for errors due to patterns not matching the regular expression above. Once the array was formatted we iterated over the JavaScript array recursively to build the tree.

3.4.2 Age Variable Subset

The NCS-R questionnaire is very thorough and includes a large amount of boolean questions. We needed to filter the variables representing those questions out so we could isolate variables having to do with age (further referred to age variables) that we could use as time series data. We analyzed the description data outlined above and found patterns in the descriptions for age variables.



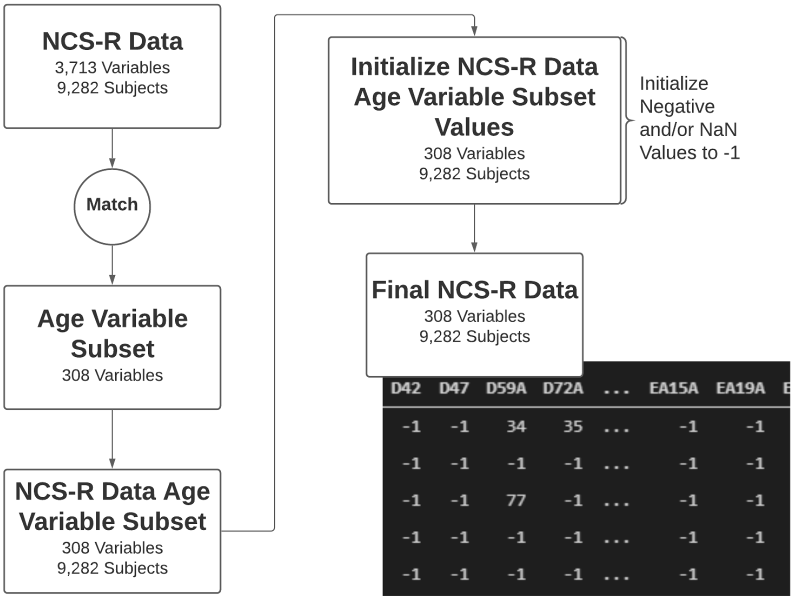
VarName	Description
D22A	Age worst episode began
D22C	Age of last bad sadness/depressed episode
D37	Exact age 1st dep episode combined with other ...
D37A	Age 1st depressed episode
D37B	Estimate age 1st episode

Figure 3.3: Process Used to Extract Age-Based Variables from the NCS-R Data Descriptions

With the patterns isolated we developed regular expressions to extract the age variable subset of the NCS-R. We found that doing this in a two-tiered approach worked best. This approach is outlined in Figure 3.3.

3.4.3 Data Initialization

With the age variables subset we could then isolate them in the NCS-R Survey Response Dataset (further referred to as just NCS-R or NCS-R Data).



```
D42  D47  D59A  D72A  ...  EA15A  EA19A  E
 -1   -1    34    35  ...     -1     -1
 -1   -1    -1    -1  ...     -1     -1
 -1   -1    77    -1  ...     -1     -1
 -1   -1    -1    -1  ...     -1     -1
 -1   -1    -1    -1  ...     -1     -1
```

Figure 3.4: Process Used to Initializes Values in the NCS-R Survey Response Dataset

Figure 3.4 outlines this process. In the NCS-R subject have options to not respond and some questions are not applicable to all subjects. These values are blank, input as *Not a Number* (NaN), or set as negative values. We initialized all of those values to -1 for ease of analysis.

Chapter 4

System Implementation & Documentation

4.1 Introduction

With the data extracted and semi-formatted we then began developing individual graphs based on Gomez-Rodriguez's algorithm for identifying latent networks (Gomez-Rodriguez et al., 2012). Then we fed that into GraphSAGE (Hamilton et al., 2017) and a DGL (Wang et al., 2019) network to infer edges and node connections. We then analyze the output values to identify potential causal events.

4.2 Neural Network Proof

4.2.1 Outline of Neural Networks

An Artificial Neural Network (ANN) is a layered system that takes input values and applies weights to determine output. It does this in a node based system, where in each layer a group of nodes, or layer values, are given weights $\mathbf{w}$ before being summed and applied to an activation layer which determines the nodes value of ”on” versus ”off.”

This process of node activation for some node y_j in the output layer for input nodes $\mathbf{x} = [x_1, x_2, ...x_n]$ and weights $\mathbf{w} = [w_1, w_2, ...w_n]$ can be described by the following formula:

$$y_j = \phi(\sum_{1}^{i=n} x_i w_i) = \phi(\mathbf{x} \cdot \mathbf{w}), \tag{4.1}$$

where $\mathbf{x} \cdot \mathbf{w}$ is the dot product of the vectors $\mathbf{x}$ and $\mathbf{w}$.

The activation function ϕ is a bounded function that normalizes values to be able to classify nodes based on a cutoff between 0 and 1. Common activation functions used are outlined in Table 4.1.

Function Name	$\phi(x)$
Sigmoid	$\frac{1}{1+e^{-x}}$
Threshold	$\begin{cases} 0 & x \leq c \\ 1 & c < x \end{cases}$
Piecewise Linear	$\begin{cases} 0 & x < c \\ 0.5 & x = c \\ 1 & c < x \end{cases}$

Table 4.1: Common Activation Functions

.

So, for a one layer neural network with n input values mapping $\mathbb{R}^n \to \mathbb{R}^m$ we can generalize across out put nodes $\mathbf{y} = [y_1, y_2, ...y_m]$ with input denoted as the vector $\mathbf{x} = [x_1, x_2, ...x_n]$. Since to find y_j m weights w_k must be applied to $x_i \in \mathbf{x}$ which can be described by the vector $\mathbf{w}_j = [w_1, w_2, ...w_m]$. We can then describe the weights over all m output nodes as

$$W = \begin{bmatrix} [\mathbf{w}_1] & \dots & [\mathbf{w}_m] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} w_{1,1} \\ \vdots \\ w_{1,n} \end{bmatrix} & \dots & \begin{bmatrix} w_{m,1} \\ \vdots \\ w_{m,n} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} w_{1,1} & \dots & w_{m,1} \\ \vdots & \ddots & \vdots \\ w_{1,n} & \dots & w_{m,n} \end{bmatrix}. \tag{4.2}$$

Then, using the sigmoid activation function $\phi(x) = \frac{1}{1+e^{-x}}$,

$$y_k = \phi(\mathbf{x} \cdot \mathbf{w}_k), \tag{4.3}$$

where $\mathbf{x} \in \mathbb{R}^n$, can be generalized for an entire layer output $\mathbf{y}$ as

$$\mathbf{y} = \phi(W^T\mathbf{x}). \tag{4.4}$$

In reality ANNs are not single layers. Because of that we will further generalize. Consider a network with depth j, $\mathbb{W}$ to be an array of matrices $[W_1, W_2, ...W_j]$ such that

$$\begin{aligned} \mathbb{W} &= \big[[W_1,] \quad \dots \quad [W_j]\big] \\ &= \big[\big[[\mathbf{w}_{1,1}] \quad \dots \quad [\mathbf{w}_{1,m}],\big] \quad \dots \quad \big[[\mathbf{w}_{j,1}] \quad \dots \quad [\mathbf{w}_{j,1}]\big]\big]. \end{aligned} \tag{4.5}$$

Then, for $Y = [[\mathbf{y}_1]\dots[\mathbf{y}_j]]$, the entire ANN can be generalized as where y_j is the final output of the ANN.

$$A(\mathbf{x}, \mathbb{W}) = \mathbf{y}_j = \phi(W_j^T(\phi(W_{j-1}^T(...\phi(W_1^T\mathbf{x})))). \tag{4.6}$$

Thus, Artificial Neural Networks are trivial given $\mathbb{W}$. That is not the reality though. The core of Artificial Neural Networks are in finding $\mathbb{W}$ using gradient descent.

4.2.2 Gradient Descent

In order to find the optimal values of $\mathbb{W}$ we will start with analyzing the performance of a $many - to - 1$ neural network with m input values, one output value, and one layer using Mean Squared Error outlined in Equation 4.7. Note that $\hat{\mathbf{y}}$ is the expected output and $y_j = \phi(\mathbf{x}_j \cdot \mathbf{w})$ is the found output for $X = [\mathbf{x}_1, ...\mathbf{x}_k]$ for k input examples where dim $\mathbf{x}_j = m$ and $\mathbf{w} = [w_1, ...w_m]$ as their weights.

$$E(\mathbf{w}) = \frac{1}{2k}\sum_{i=0}^{k}(\hat{y}_i - y_i)^2 \tag{4.7}$$

Note that as $E(\mathbf{w})$ approaches zero our output $\mathbf{y}$ grows closer to our expected values. Consider the problem

$$\mathbf{w}_{min} = \text{arg min } E(\mathbf{w}), \tag{4.8}$$

where $\mathbf{w}_{min}$ is a vector of the weights $\mathbf{w}$ such that E is minimized, which can be solved using

$$M(\mathbf{w}_t) = \mathbf{w}_{t+1} = \mathbf{w}_t - \alpha\nabla E(\mathbf{w}_t), \tag{4.9}$$

where

$$\nabla E(\mathbf{w}_t) = [\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots \frac{\partial E}{\partial w_k}]. \tag{4.10}$$

Thus, M modifies $\mathbf{w}_t$ to be closer to $\mathbf{w}_{min}$ by incrementing it by $\alpha\nabla E(\mathbf{w}_t)$, where α is a positive constant also known as the learning rate.

Consider w_j, a single weight in our network. $\frac{\partial E}{\partial w_j}$ then represents the impact of w_j on E and we can then derive $\nabla E(\mathbf{w}_t)$ for $\frac{\partial E}{\partial w_j}$ where $w_j \in \mathbf{w}_t$, outlined below.

$$
\begin{align*}
\frac{\partial E}{\partial w_j} &= \frac{\partial}{\partial w_j}(\frac{1}{2k}\sum_{i=0}^{k}(\hat{y}_i - y_i)^2) \\
&= \frac{1}{2k}\sum_{i=0}^{k}\frac{\partial}{\partial w_j}(\hat{y}_i - y_i)^2 \\
&= \frac{1}{2k}\sum_{i=0}^{k}2(\hat{y}_i - y_i)\frac{\partial}{\partial w_j}(\hat{y}_i - y_i) \\
&= \frac{1}{k}\sum_{i=0}^{k}(\hat{y}_i - y_i)\frac{\partial}{\partial w_j}(\hat{y}_i - \phi(y_i)) \\
&= \frac{1}{k}\sum_{i=0}^{k}(\hat{y}_i - y_i)(\frac{\partial}{\partial w_j}(\hat{y}_i) - \frac{\partial}{\partial w_j}\phi(\mathbf{x}_i \cdot \mathbf{w})) \\
&= \frac{1}{k}\sum_{i=0}^{k}(\hat{y}_i - y_i)(-\frac{\partial \phi(\mathbf{x}_i \cdot \mathbf{w})}{\partial w_j}) \tag{4.11} \\
&= \frac{1}{k}\sum_{i=0}^{k}(\hat{y}_i - y_i)(-\frac{\partial \phi(\mathbf{x}_i \cdot \mathbf{w})}{\partial(\mathbf{x}_i \cdot \mathbf{w})}\frac{\partial(\mathbf{x}_i \cdot \mathbf{w})}{\partial w_j}) \\
&= \frac{1}{k}\sum_{i=0}^{k}(y_i - \hat{y}_i)(\phi(\mathbf{x}_i \cdot \mathbf{w})(1 - \phi(\mathbf{x}_i \cdot \mathbf{w}))\frac{\partial(\mathbf{x}_i \cdot \mathbf{w})}{\partial w_j}) \\
&= \frac{1}{k}\sum_{i=0}^{k}(y_i - \hat{y}_i)y_i(1 - y_i)\frac{\partial}{\partial w_j}(w_1x_{i,1} + w_2x_{i,2} - \ldots w_jx_{i,j} + \ldots w_nx_{i,n}) \\
&= \frac{1}{k}\sum_{i=0}^{k}(y_i - \hat{y}_i)y_i(1 - y_i)(x_{i,j}) \\
&= \frac{1}{k}\sum_{i=0}^{k}y_i(1 - y_i)(y_i - \hat{y}_i)(x_{i,j})
\end{align*}
$$

Then, using Equation 4.11,

$$\nabla E(\mathbf{w}_t) = [\sum_{i=0}^{k}y_i(1 - y_i)(y_i - \hat{y}_i)(x_{i,1}), \ldots \sum_{i=0}^{k}y_i(1 - y_i)(y_i - \hat{y}_i)(x_{i,m})]. \tag{4.12}$$

And,

$$M(\mathbf{w}_t) = \mathbf{w}_{t+1} = \begin{bmatrix} w_{t,1} - \frac{\alpha}{k}\sum_{i=0}^{k} y_i(1-y_i)(y_i-\hat{y}_i)(x_{i,1}) \\ w_{t,2} - \frac{\alpha}{k}\sum_{i=0}^{k} y_i(1-y_i)(y_i-\hat{y}_i)(x_{i,2}) \\ \vdots \\ w_{t,m} - \frac{\alpha}{k}\sum_{i=0}^{k} y_i(1-y_i)(y_i-\hat{y}_i)(x_{i,m}) \end{bmatrix}, \tag{4.13}$$

where $w_{t,v}$ is the v^{th} weight in $\mathbf{w}_t$.

If our network were to have l output variables instead of 1 such that $Y = [\mathbf{y}_1, \ldots \mathbf{y}_l]$, our weights would be com $W = [\mathbf{w}_1, \ldots \mathbf{w}_l]$, and E would become E^*, below.

$$E^*(W) = \frac{1}{2km}\sum_{i=0}^{l}\sum_{n=0}^{k}(\hat{y}_{i,n} - y_{i,n})^2 = \frac{1}{m}\sum_{n=0}^{m} E(\mathbf{w}_n) \tag{4.14}$$

And M^* as

$$M^*(W_t) = W_t - \alpha\nabla E^*(W_t) \tag{4.15}$$

and,

$$\nabla E^*(W_t) = [\frac{\partial E^*}{\partial \mathbf{w}_1}, \ldots \frac{\partial E^*}{\partial \mathbf{w}_l}], \tag{4.16}$$

where

$$\begin{aligned} \frac{\partial E^*}{\partial \mathbf{w}_j} &= \frac{\partial}{\partial \mathbf{w}_j}\frac{1}{m}\sum_{n=0}^{m} E(\mathbf{w}_j) \\ &= \frac{1}{m}\sum_{n=0}^{m}\frac{\partial}{\partial \mathbf{w}_j}E(\mathbf{w}_j) \\ &= \frac{1}{m}\sum_{n=0}^{m}\begin{bmatrix} \frac{\partial}{\partial w_{j,1}}E(\mathbf{w}_j) \\ \vdots \\ \frac{\partial}{\partial w_{j,l}}E(\mathbf{w}_j) \end{bmatrix}. \end{aligned} \tag{4.17}$$

4.2.3 Back Propagation

For a multi-layer network our error function does not change and M continues to grow in complication. Therefore, instead of calculating gradient descent for the entire network as a whole ANNs use a process known as back propagation where for each $\mathbf{x}_n \in X$, $\mathbf{x}_n$ is fed through the ANN to find $\mathbf{y}_n$ and calculate $\frac{\partial}{\partial h_{a,b}}E^*(\mathbf{h}_a)$

for each hidden node $h_{a,b} \in H$ where $h_{a,b}$ is the b^{th} weight of the a^{th} layer of the network and $\mathbf{h}_a \in \mathbb{R}^m$. Then,

$$\begin{aligned} h^*_{a,b} &= h_{a,b} - \alpha \frac{\partial}{\partial h_{a,b}} \frac{1}{m} \sum_{n=0}^{m} (\hat{y}_n - y_n)^2 \\ &= h_{a,b} - \frac{1}{m}\frac{1}{2} \sum_{n=0}^{m} y_i(1-y_i)(\hat{y}_n - y_n)^2 \end{aligned} \tag{4.18}$$

4.3 System Design

In Figure 4.1 we outline the basics of our system. We begin by taking the 'Final NCS-R Data' described in Figure 3.4 and developing individual graphs for each subject. Then we isolate the comorbid group we are focusing on. In this case we isolated General Anxiety Disorder (GAD) and Major Depressive Disorder (MDD). We take all of the individual graphs and run them through a hawkes process and DDP to output a cumulative directed graph for all subjects and our comorbid subset. We use this to isolate specific causal features for the comorbid subset extracting common edges from the all subject graph.

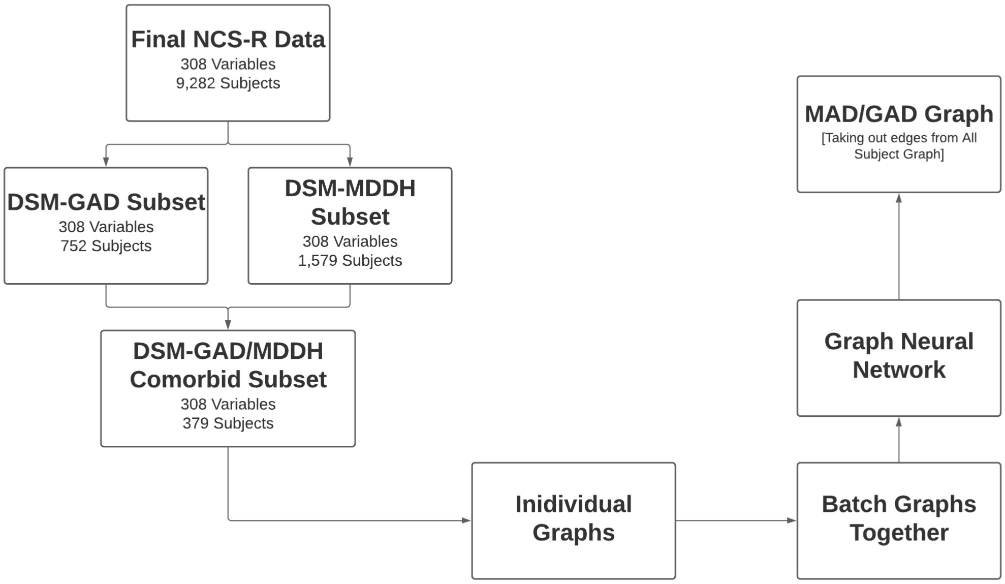


Figure 4.1: Overview of System Implementation

4.3.1 Individual Graphs

The 'Individual Graphs' are directed graphs depicting the flow of symptoms in a subject. The individual graphs are a level-based graph where nodes are separated into levels by age and then connected to all nodes in adjacent levels, as outlined in Algorithm C.1 in the appendix. The development of these graphs was based on Gomez-Rodriguez's work (Gomez-Rodriguez et al., 2012).

Below, Figure 4.2 depicts an individual graph for Case Id 3333. The values of each node can be found at `https://www.icpsr.umich.edu/web/ICPSR/studies/20240/variables`.

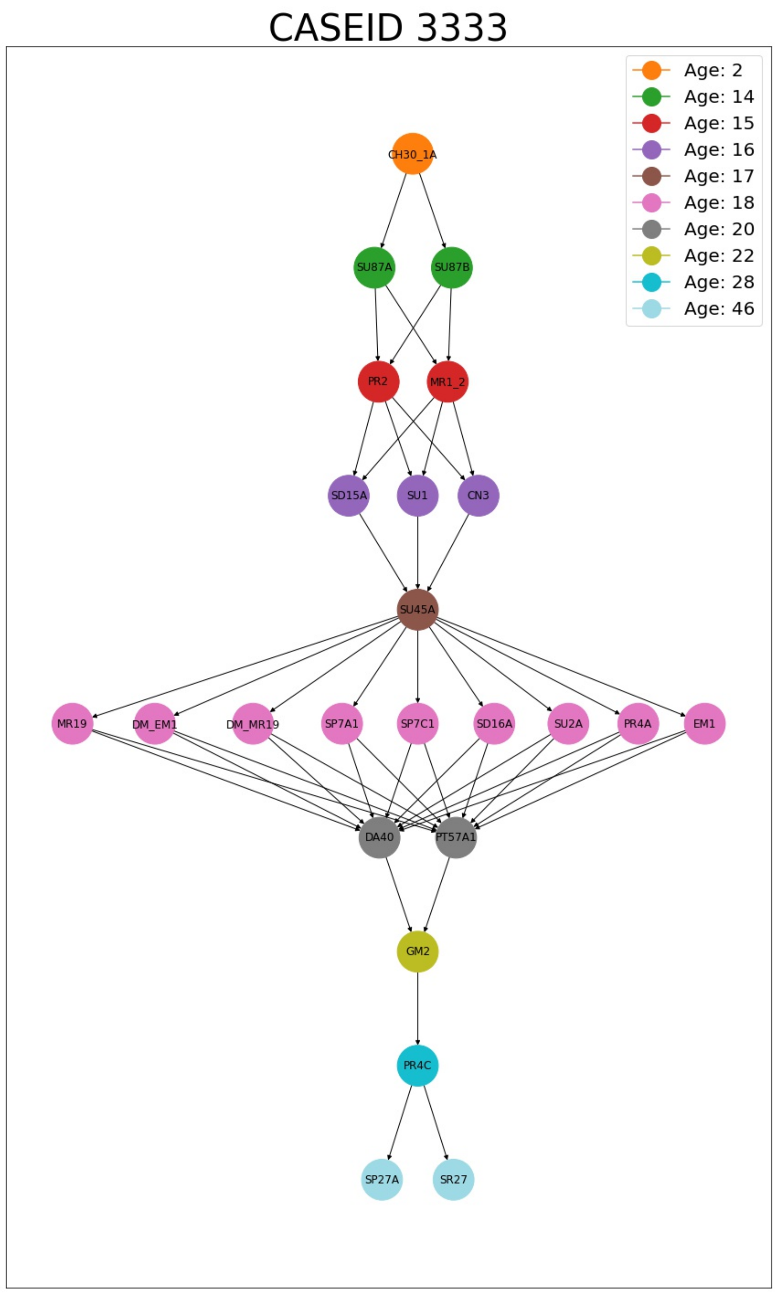


Figure 4.2: Individual Graph for Case ID 3333

We then use the individual graphs in a Graph Neural Network.

4.3.2 Graph Neural Network

Using the Deep Graph Library (DGL) we then batched all individual graphs from a single diagnosis into a large graph which we will call a ‘Diagnosis Graph’ these diagnosis graphs create unique nodes for each node in an individual graph ignoring whether the nodes represent the same system. To account for this we added node features that were transferred from the individual graphs to the diagnosis graphs. One feature represented the event/symptom of the node and the other was the age at which the node occurred in the subject.

Once the diagnosis graph was completed we developed our neural network which is a 3-Layer Convolutional Neural Network which uses the GraphSAGE encoding and Long Short-Term Memory on the back end to identify pathways between nodes. We then split the diagnosis graph into a training and a testing set by subsetting a section of edges which we could test and see if the model identified. We also subset our individual graphs before batching to mitigate any over fitting.

Our model then returned edge weight values for every pathway it identified between nodes which we used to isolate the most prominent edges and analyze for causal events/symptoms.

Chapter 5

Results & Conclusions

5.1 Introduction

We were able to successfully implement our GNN model on the NCS-R dataset with the GAD-MDDH Comorbid set. We used Area Under the Curve (AUC) to analyze the performance of our models before implementing them and drawing conclusions.

5.2 Model Building & Selection

There are 379 subject who are diagnosed with both Major Depressive disorder and General Anxiety disorder. To build our model with the Diagnosis Graphs for the comorbid diagnosis we built our GNN while changing parameters to find the best AUC. The best AUC we achieved was 78%, but that was a case of overfitting. Otherwise our AUC plateaued at 75% for GAD, MDD, and the comorbid diagnosis as seen in Figures A.1 - A.3, in the Appendix.

5.2.1 Identifying Edge Weights

Because our model was trained and tested on the batched data instead of the individual graphs the return values of the program, or the edge weights, are applied to the node of an individuals' symptom instead of to the symtom as a whole.

To return the graphs to the original format (i.e. every node represents a symptom), we used the node attributes to group all nodes together and then iterated through every edge & weight in the batched graph averaging the edge weight of repeated edges. Then we removed any nodes that had no edges connected to them.

With the edge weights in usable format and a general confidence that our model preforms better than average (which would be an AUC of .5) we began selecting a cutoff point for the edge weights. The edge weights were heavily skewed to the left, centering below 0.2. We decided to develop histograms to identify the best cutoff point.

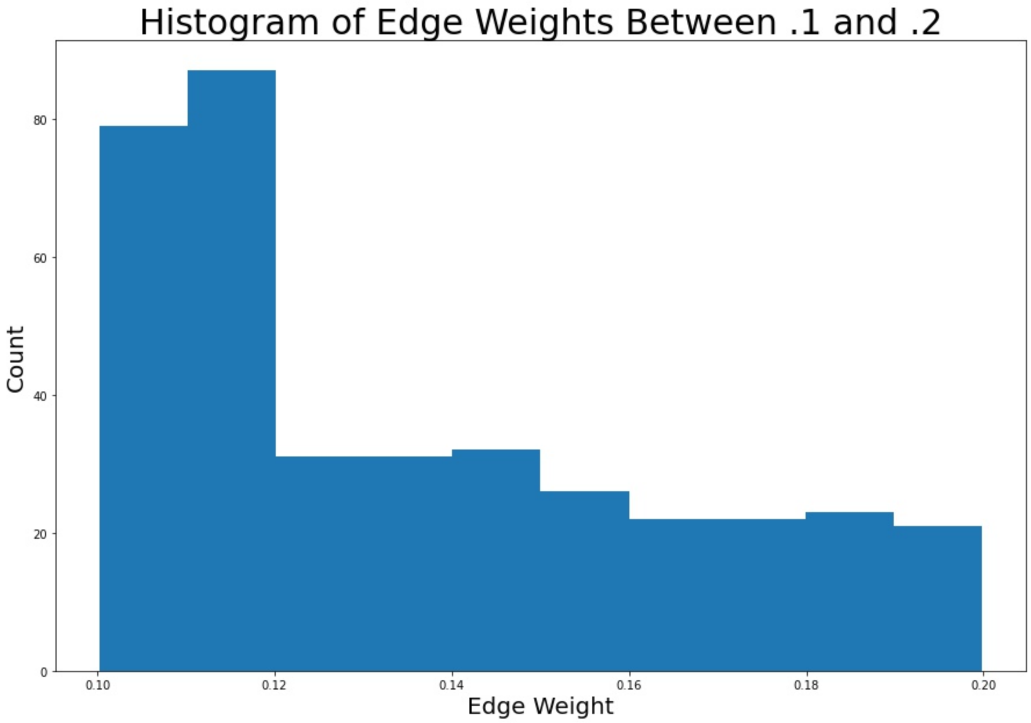


Figure 5.1

As seen in Figure 5.1, over 160 edges reside under 1.2. We continued to cut off from there.

Looking at the weights above .12 in Figure 5.2, over 350 edges are have weight below 0.5. We used the cutoff of 0.5 to allow for readability of the graphs we will develop.

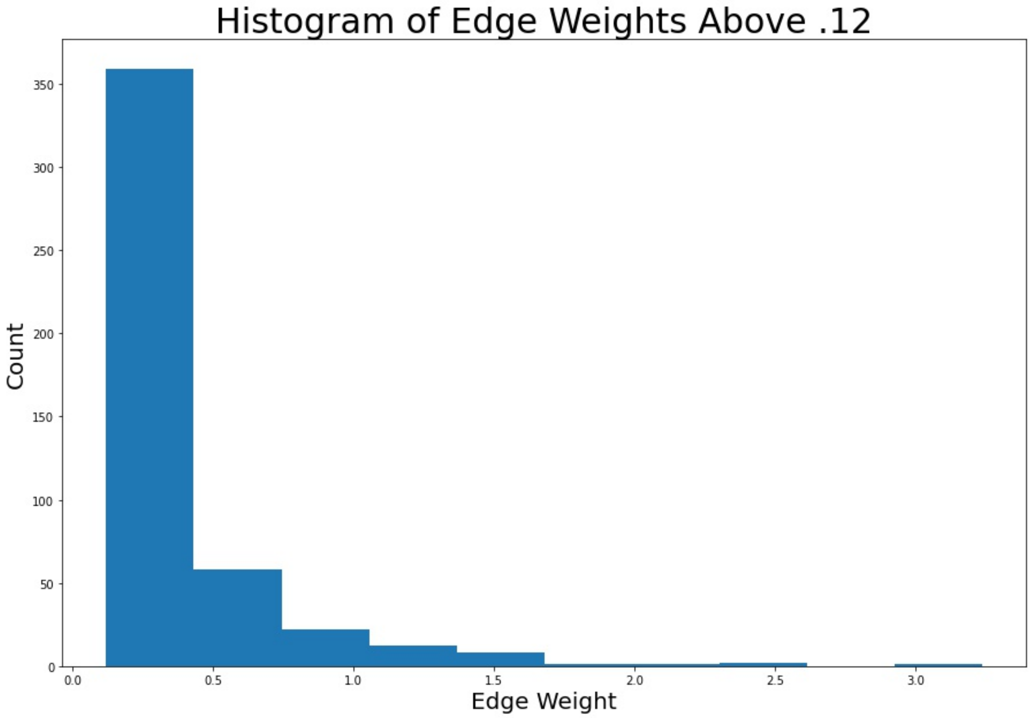


Figure 5.2

5.3 Comorbidity Graphs

To develop our Comorbidity Graphs that can identify latent pathways we used NetworkX (Hagberg et al., 2008). Figure 5.3 shows the results of the graphs in a main subplot. The model made 10 subplots for GAD-MDDH comorbid subject, but onely one had more than 4 nodes.

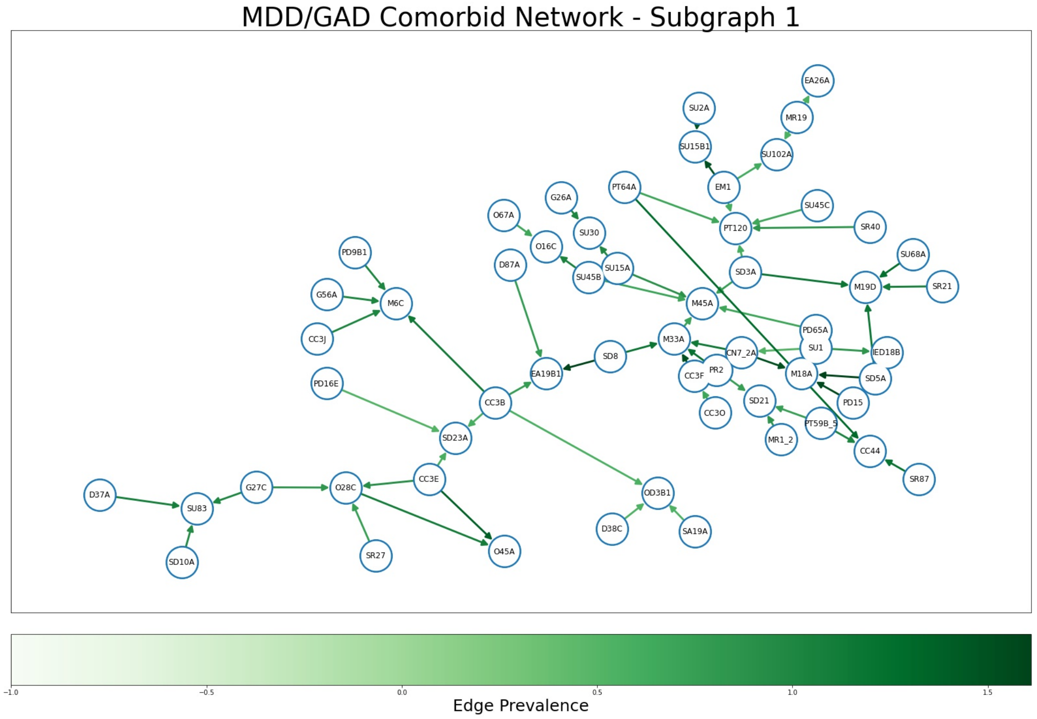


Figure 5.3: Comorbid GAD, MDD Graph 1

Code	Description
D37A	Age 1st depressed episode
D38C	Age last time with episode
D87A	Age 1st hospitalized overnight for sadness
M6C	Age most recent irritable period with most behavioral changes
M18A	Age of 1st excited/grouchy episode and behavioral changes
M19D	Age last time excited/grouchy and behavioral episodes
M33A	Age 1st time talk to professional
M45A	Age 1st time received helpful/effective treatment for episodes
PD9B1	Estimate age 1st attack
PD15	Age 1st month of worry/change activities/avoid sits due to attack
PD16E	Age last time had month of worry about attack
PD65A	Age 1st time hospitalized overnight for attacks
G26A	Age 1st worry/anxious/nervous episode + other problems
G27C	Age last time had 1 of these worried episodes
G56A	Age 1st received helpful/effective treatment for worry/anxiety/nerves
IED18B	Approximate age 1st attack
SD3A	Age last seriously thought about suicide
SD5A	Age last made suicide plan
SD8	Age 1st time attempted suicide
SD10A	Age last time attempted suicide
SD21	Age 1st time attempted suicide
SD23A	Age last time attempted suicide
SR21	Age last saw psychiatrist
SR27	Age 1st saw medical doctor for mental health/sub use
SR40	Age 1st saw psychologist about mental health/sub use

Table 5.1: Node Descriptions for Figure 5.3

5.4 Conclusions

The graphs gained through this process with GNNs have the potential to be a revolutionary way to combat mental illness. By identifying directed pathways of diseases mental health professionals will be able to identify causal events and symptoms that have not been identified previously. This will aid in the mitigation of mental illnesses and potentially the identification and treatment of comorbid illnesses early before symptoms and life impact become severe

5.4.1 Issues

This process is not refined as of yet. While the AUC was above random, averaging .75, there was not a significant accuracy in its identification of pathways. In the future we would like to work with mental health professionals to create more accurate individual graphs with which to train our dataset. Also we would like to implement a Hawkes Process (Hawkes, 1971) and Deep Diffusion Process (Qian et al., 2020) to look at the impact of symptoms and events beyond one layer from the current event.

5.4.2 Next Steps

In the future we would like to compare more thoroughly the generated pathways of comorbid diagnosis with those of individual diagnosis to identify more bridge symptoms and events/symptoms unique to comorbidity. We would also like to do a further analysis of the variables in the NCS-R (Kessler et al., 2004) dataset to identify more pathways that can be inferred. We would further like to consult with mental health professionals on the accuracy of the generated pathways and do a comparison against the current DSM guidelines.

5.4.3 Acknowledgements

I'd like to thank Dr. Vance for his perseverance through my process of finding a working method. Thanks to Dr. Hezy for pushing me to complete good work. Finally, I'd like to thank the entire University of Virginia's College at Wise's Math & Computer Science Faculty for putting up with my constant questions and encouraging me to learn more every day.

Bibliography

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*. 10

Borsboom, D. (2008). Latent variable theory. 3

Cramer, A. O., Waldorp, L. J., Van Der Maas, H. L., and Borsboom, D. (2010). Comorbidity: A network perspective. *Behavioral and brain sciences*, 33(2-3):137. 0, 3, 6, 7, 8

Danks, D., Fancsali, S., Glymour, C., and Scheines, R. (2010). Comorbid science? *Behavioral and Brain Sciences*, 33(2-3):153–155. 6

Gomez-Rodriguez, M., Leskovec, J., and Krause, A. (2012). Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):1–37. 0, 8, 9, 16, 22

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA. 12, 27

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. *CoRR*, abs/1706.02216. 3, 10, 16

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90. 30

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95. 12

Kessler, R. C., Berglund, P., Chiu, W. T., Demler, O., Heeringa, S., Hiripi, E., Jin, R., Pennell, B.-E., Walters, E. E., Zaslavsky, A., et al. (2004). The us national comorbidity survey replication (ncs-r): design and field procedures. *International journal of methods in psychiatric research*, 13(2):69–92. 3, 4, 5, 12, 30, 43, 44, 45, 46, 47, 48

Kokkevi, A. and Stefanis, C. (1995). Drug abuse and psychiatric comorbidity. *Comprehensive Psychiatry*, 36(5):329–337. 2

Prendergast, L. (2019). What makes python so versatile? *Medium*. 11

Qian, Z., Alaa, A., Bellot, A., Schaar, M., and Rashbass, J. (2020). Learning dynamic and personalized comorbidity networks from event data using deep diffusion processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3295–3305. PMLR. 0, 9, 12, 30

Regier, D. A., Farmer, M. E., Rae, D. S., Locke, B. Z., Keith, S. J., Judd, L. L., and Goodwin, F. K. (1990). Comorbidity of mental disorders with alcohol and other drug abuse: results from the epidemiologic catchment area (eca) study. *Jama*, 264(19):2511–2518. 2

Sanchez-Gonzalez, A., Heess, N., Springenberg, J. T., Merel, J., Riedmiller, M., Hadsell, R., and Battaglia, P. (2018). Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*, pages 4470–4479. PMLR. 10

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80. 10

Sokal, J., Messias, E., Dickerson, F. B., Kreyenbuhl, J., Brown, C. H., Goldberg, R. W., and Dixon, L. B. (2004). Comorbidity of medical illnesses among adults with serious mental illness who are receiving community psychiatric services. *The Journal of nervous and mental disease*, 192(6):421–427. 2

Volkow, N. D. (2001). Drug abuse and mental illness: progress in understanding comorbidity. *American Journal of Psychiatry*, 158(8):1181–1183. 2

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., et al. (2019). Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*. 3, 10, 16

Appendix A

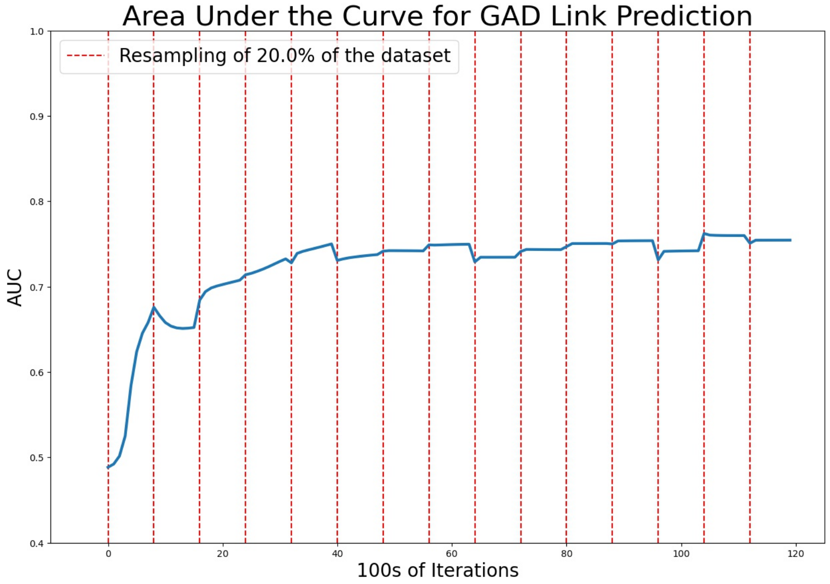


Figure A.1: AUC for Model with General Anxiety Disorder

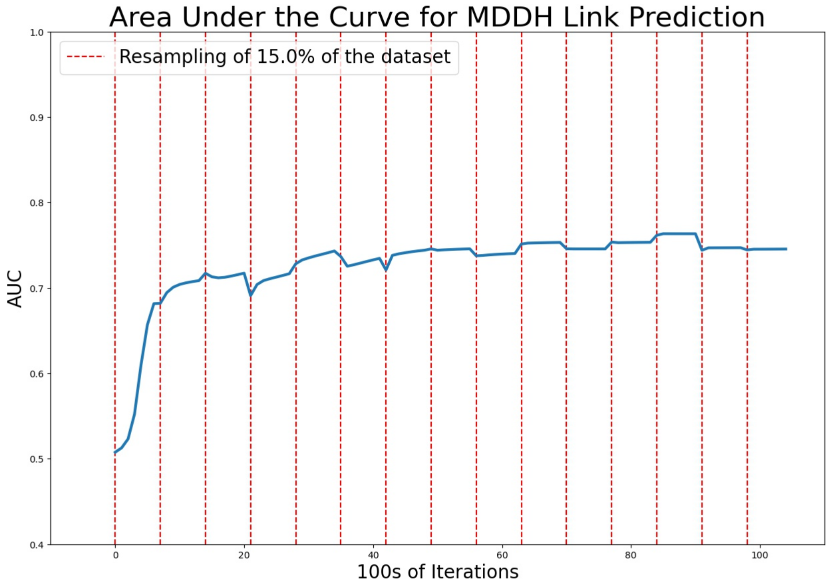


Figure A.2: AUC for Model with Major Depressive Disorder

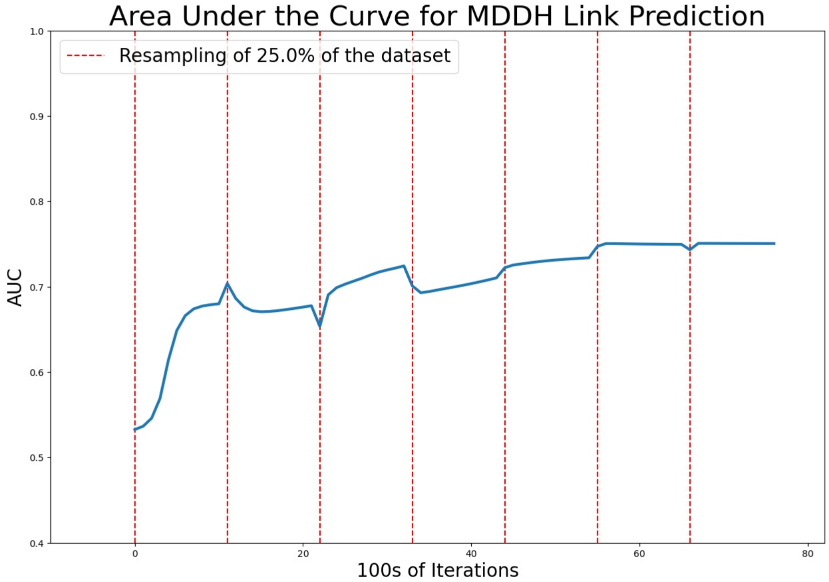


Figure A.3: AUC for Model for Comorbid GAD/MAD Subjects

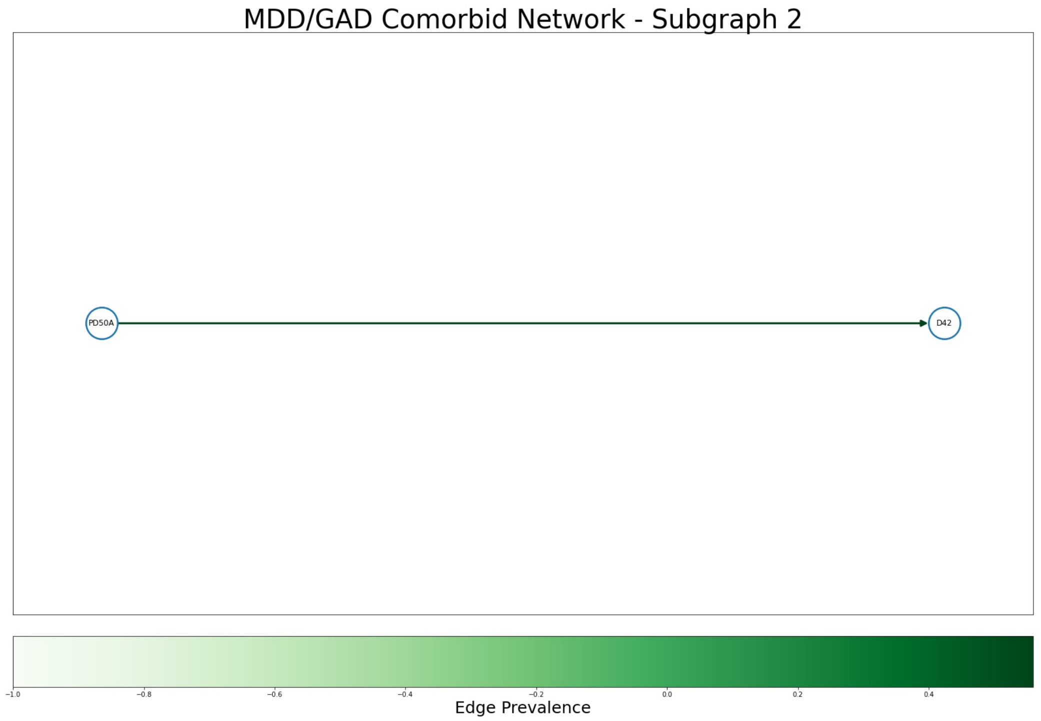


Figure A.4: Comorbid GAD, MDD Graph 2

Code	**D**escription
D42	Age had 1st year of episodes every month
PD50A	Age 1st see professional for attacks

Table A.1: Node Descriptions for Figure A.4

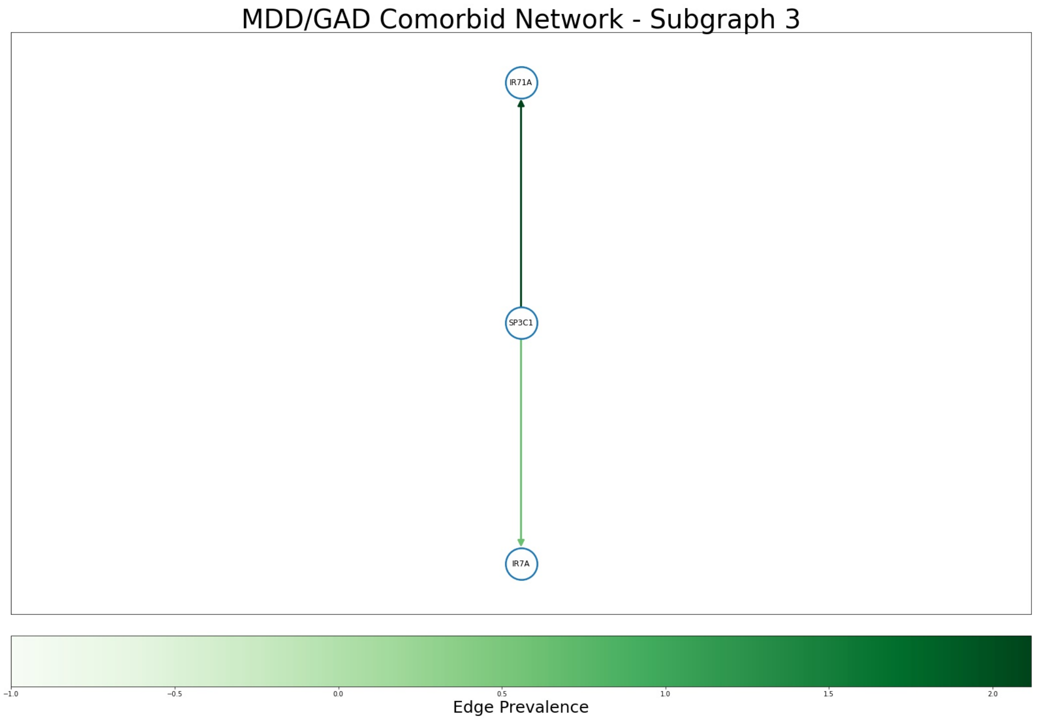


Figure A.5: Comorbid GAD, MDD Graph 3

Code	Description
SP3C1	Age 1st avoided situations to see animals
IR71A	Age 1st time hospitalized overnight for irritability
IR7A	Age worst episode occurred

Table A.2: Node Descriptions for Figure A.5

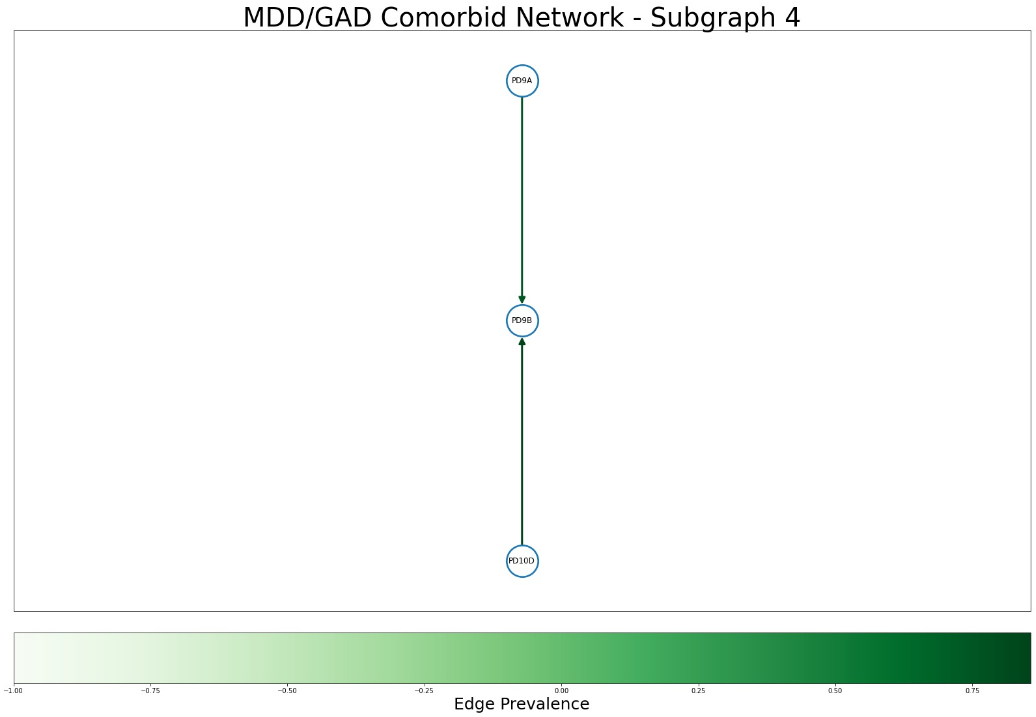


Figure A.6: Comorbid GAD, MDD Graph 3

Code	**Description**
PD9B	Estimate age 1st attack-computed
PD9A	Age 1st attack
PD10D	Age at last attack

Table A.3: Node Descriptions for Figure A.6

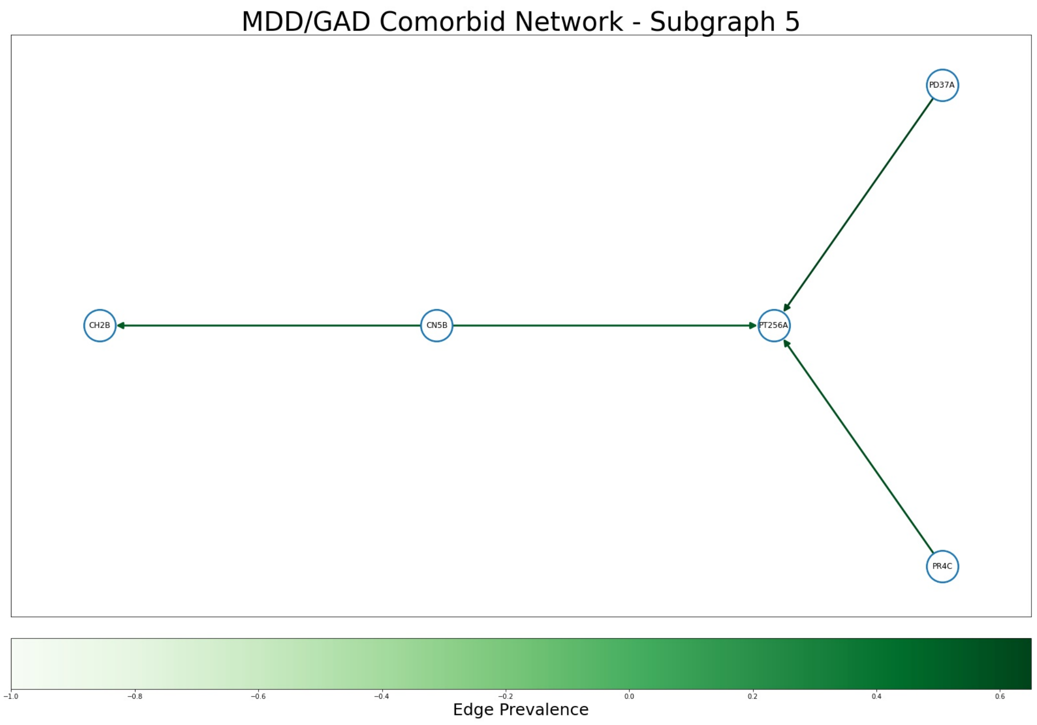


Figure A.7: Comorbid GAD, MDD Graph 5

Code	Description
CH2B	Your age when your father died
PD37A	Age at last sudden attack
CN5B	Age 1st miscarriage/stillbirth
PT256A	Age 1st received helpful/effective treatment for reactions
PR4C	Age stopped taking birth control pills

Table A.4: Node Descriptions for Figure A.7

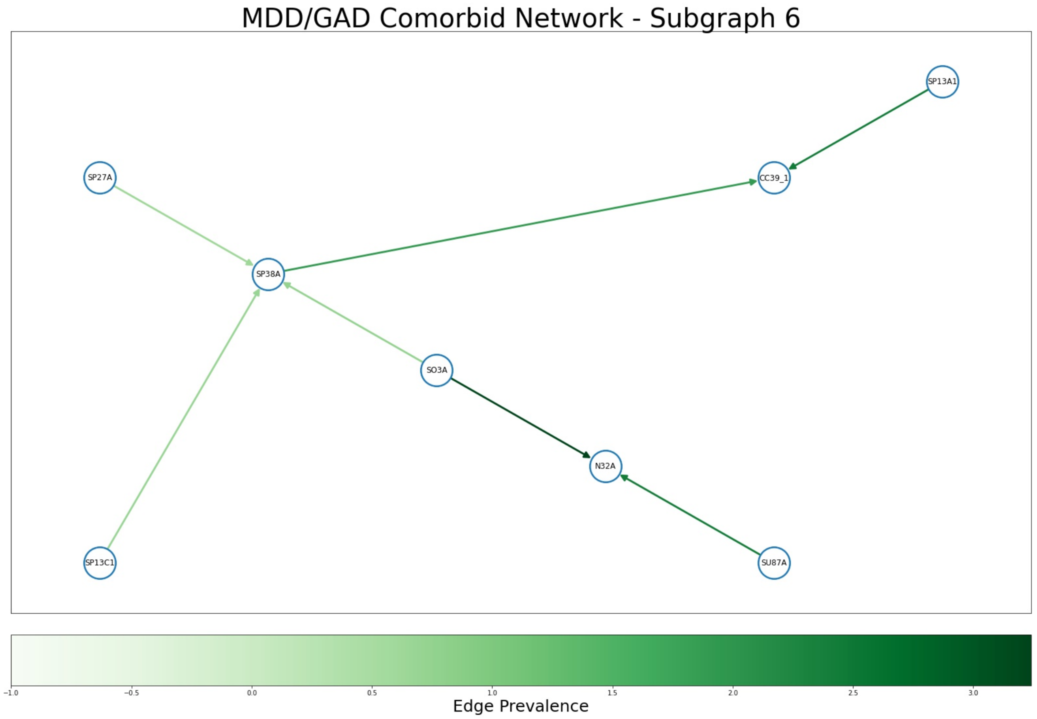


Figure A.8: Comorbid GAD, MDD Graph 6

Code	**Description**
SP13A1	Age 1st feared flying
SU87A	Age 1st opportunity to use alcohol
SP38A	Age 1st received helpful/effective treatment for fear
SO3A	Age 1st fear of social situations
N32A	Age 1st talked to professional about tiredness
SP13C1	Age 1st avoided flying
SP27A	Age 1st talk to medical doc/professional about fear
CC39˙1	Age when received diagnosis for colitis/Crohn's Disease

Table A.5: Node Descriptions for Figure A.8

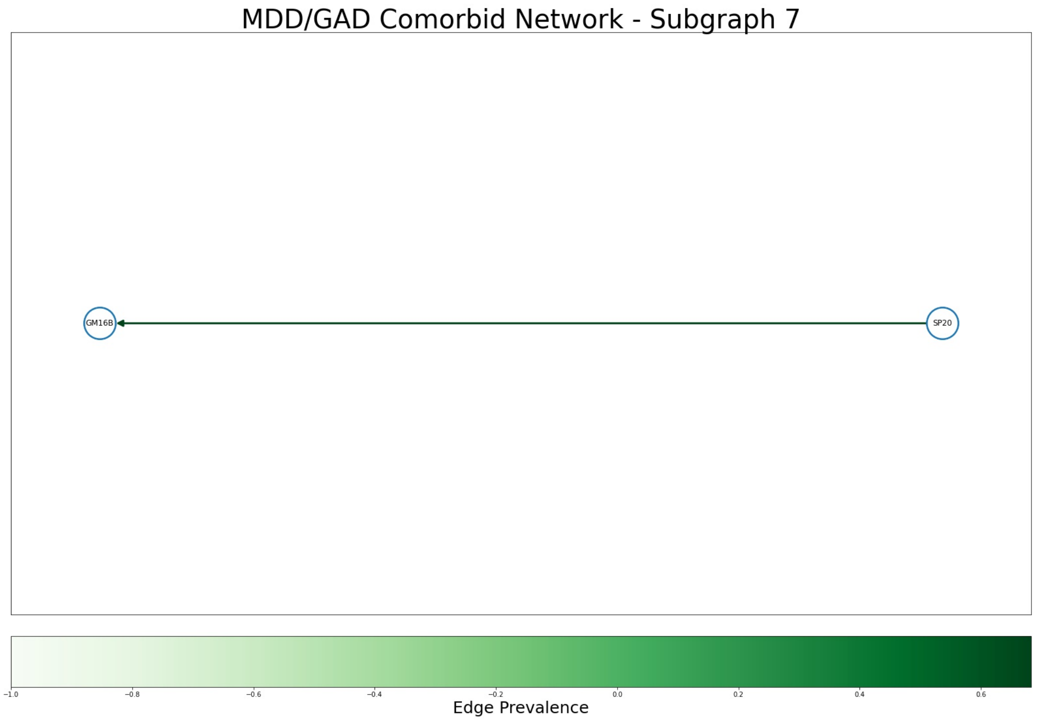


Figure A.9: Comorbid GAD, MDD Graph 7

Code	**Description**
SP20	Age last time strongly feared or avoided things
GM16B	Estimate age 1st problem because of gambling

Table A.6: Node Descriptions for Figure A.9

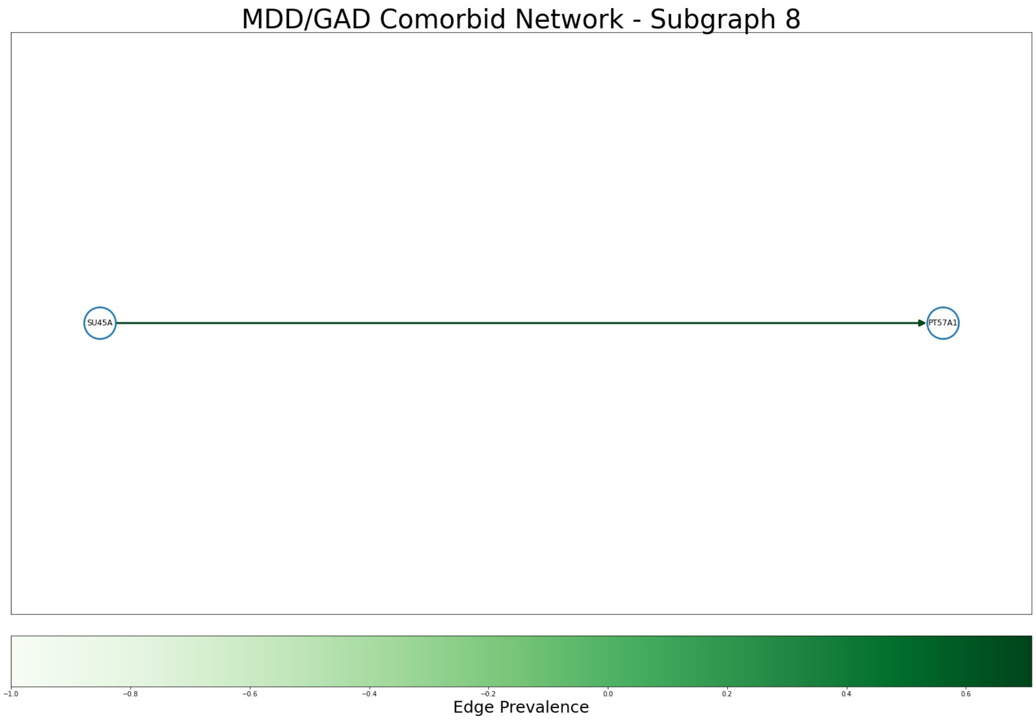


Figure A.10: Comorbid GAD, MDD Graph 8

Code	**Description**
SU45A	Age 1st used marijuana/hash
PT57A1	Age at most upsetting traumatic event not reported

Table A.7: Node Descriptions for Figure A.10

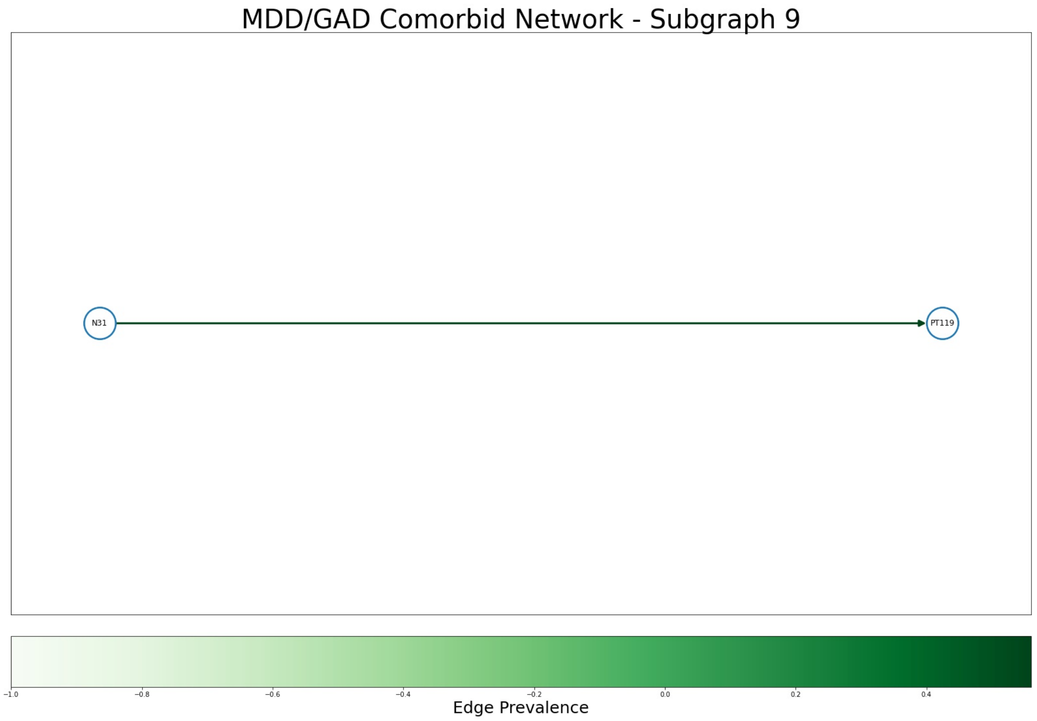


Figure A.11: Comorbid GAD, MDD Graph 9

Code	**Description**
PT119	Age at RANDOM EVENT
N31	Age 1st talked to professional about tiredness

Table A.8: Node Descriptions for Figure A.11

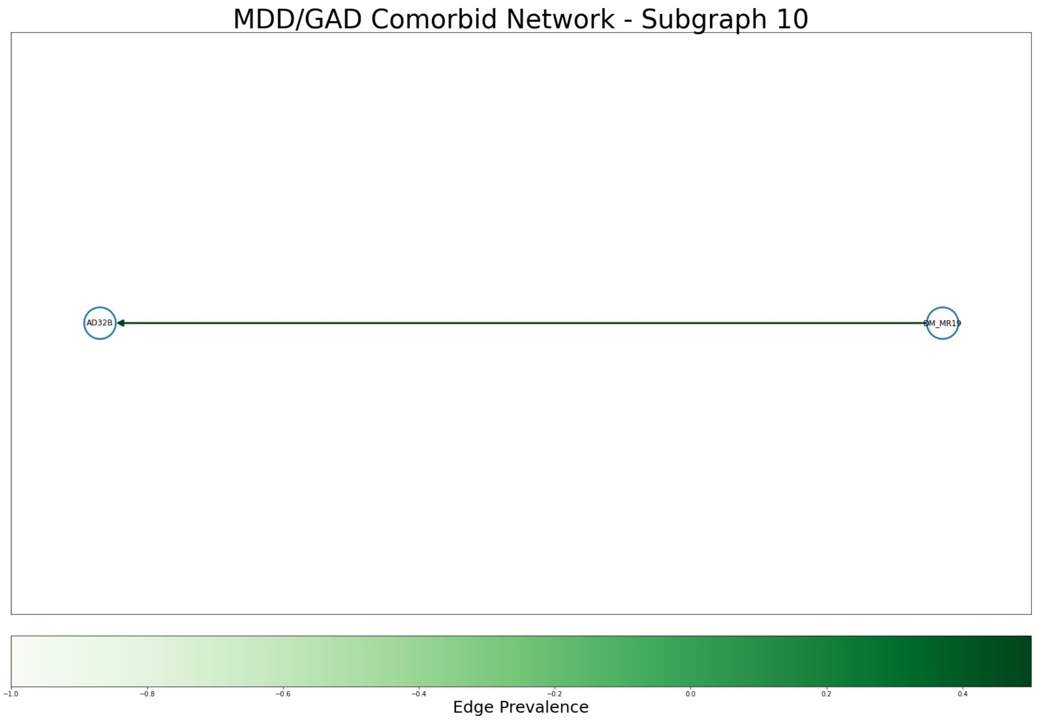


Figure A.12: Comorbid GAD, MDD Graph 10

Code	Description
DM˙MR19	Age you married 1st time: Top/Bot Code
AD32B	Estimate age of 1st restless/impatient difficulty-computed

Table A.9: Node Descriptions for Figure A.12

Table A.10: DSM Diagnosis Counts in the NCSR Kessler et al. (2004).

Begin of Table	
DSM Diagnosis	**Count**
DSM-IV Attention Deficit Disorder (LifeT)	365
DSM-IV Agoraphobia without Panic Disorder (LifeT)	231
DSM-IV Agoraphobia with Panic Disorder (LifeT)	126

Continuation of Table A.10 - DSM Diagnosis Counts in the NCSR Kessler et al. (2004)

DSM Diagnosis	**Count**
DSM-IV Alcohol Abuse (Lifetime)	1034
Lifetime Alcohol Abuse w/ hierarchy	590
DSM-IV Alcohol Dependence (Lifetime)	444
DSM-IV Adult Separation Anxiety Disorder (LifeT)	558
DSM-IV Bi-Polar I (Lifetime)	101
DSM-IV Bi-Polar II (Lifetime)	105
Lifetime Bi-Polar Subthreshold	210
DSM-IV Conduct Disorder (Lifetime)	405
DSM-IV Drug Abuse (Lifetime)	651
DSM-IV Drug Dependence (Lifetime)	248
DSM-IV Dysthymia (Lifetime)	386
DSM-IV Dysthymia with hierarchy (LifeT)	232
DSM-IV Generalized Anxiety Disorder (LifeT)	752
DSM-IV Gen Anxiety Disorder w/hierarchy (LifeT)	553
DSM-IV Hypomania (Lifetime)	77
DSM-IV Intermittent Explosive Disorder (LifeT)	728
DSM-IV Intermittent Explosive Disorder w/ hierarchy (LifeT)	678
DSM-IV Mania (Lifetime)	339
DSM-IV Major Depressive Disorder w/ hierarchy (LifeT)	1579
DSM-IV Major Depressive Episode (Lifetime)	1829
DSM-IV Oppositional Defiant Disorder (LifeT)	453
DSM-IV Oppositionl Defiant Disorder w/ hierarchy (LifeT)	375
DSM-IV Panic Attack (Lifetime)	2573
DSM-IV Panic Disorder (Lifetime)	455
DSM-IV Posttraumatic Stress Disorder (LifeT)	604

Continuation of Table A.10 - DSM Diagnosis Counts in the NCSR Kessler et al. (2004)

DSM Diagnosis	**Count**
DSM-IV Separation Anxiety Disorder (LifeT)	331
DSM-IV Social Phobia (Lifetime)	1143
DSM-IV Specific Phobia (Lifetime)	1198
DSM-IV Nicotine Dependence (Lifetime)	626
DSM-IV Alcohol Abuse (30Day)	64
DSM-IV Alcohol Abuse w/hierarchy (30Day)	31
DSM-IV Alcohol Dependence (30 day)	43
DSM-IV Adult Separation Anxiety Disorder (30Day)	68
DSM-IV Drug Abuse (30 day)	32
DSM-IV Drug Abuse w/ hierarchy (30 day)	18
DSM-IV Drug Dependence (30 day)	14
DSM-IV Dysthymia (30 day)	122
DSM-IV Dysthymia with hierarchy (30 day)	71
DSM-IV Generalized Anxiety Disorder (30Day)	157
DSM-IV Gen Anxiety Disorder w/hierarchy (30Day)	97
DSM-IV Hypomania (30 day)	9
DSM-IV Intermittent Explosive Disorder (30day)	161
DSM-IV Intermittent Explosive Disorder w/ hierarchy (30Day)	151
DSM-IV Mania (30 day)	65
DSM-IV Major Depressive Disorder w/ hierarchy (30Day)	233
DSM-IV Major Depressive Episode (30 day)	301
DSM-IV Panic Attack (30 day)	306
DSM-IV Panic Disorder (30 day)	105
DSM-IV Social Phobia (30 day)	334
DSM-IV Specific Phobia (30 day)	592

Continuation of Table A.10 - DSM Diagnosis Counts in the NCSR Kessler et al. (2004)

DSM Diagnosis	**Count**
DSM-IV Agoraphobia without Panic Disorder (30Day)	77
DSM-IV Agoraphobia with Panic Disorder (30Day)	38
DSM-IV Nicotine Dependence (30 day)	193
DSM-IV Anorexia (Lifetime)	21
DSM-IV Binge Eating Disorder w/ hierarchy (Lifetime)	105
DSM-IV Binge Any (Lifetime)	192
DSM-IV Bulimia (Lifetime)	53
DSM-IV Bulimia w/ hierarchy (Lifetime)	52
DSM-IV Binge Eating Disorder w/ hierarchy (12Mo)	51
DSM-IV Binge Any (12Mo)	86
DSM-IV Bulimia (12Mo)	16
DSM-IV Bulimia w/ hierarchy (12Mo)	16
DSM-IV Attention Deficit Disorder (12Mo)	190
DSM-IV Agoraphobia without Panic Disorder (12Mo)	138
DSM-IV Agoraphobia with Panic Disorder (12Mo)	73
DSM-IV Alcohol Abuse (12Mo)	213
DSM-IV Alcohol Abuse w/ hierarchy (12Mo)	113
DSM-IV Alcohol Dependence (12 month)	106
DSM-IV Adult Separation Anxiety Disorder (12Mo)	156
DSM-IV Bi-polar I (12Mo)	65
DSM-IV Bi-polar II (12Mo)	74
DSM-IV Bi-Polar Subthreshold (12Mo)	123
DSM-IV Conduct Disorder (12 month)	33
DSM-IV Drug Abuse (12 month)	102
DSM-IV Drug Abuse w/ hierarchy (12 month)	61

Continuation of Table A.10 - DSM Diagnosis Counts in the NCSR Kessler et al. (2004)

DSM Diagnosis	**Count**
DSM-IV Drug Dependence (12 month)	36
DSM-IV Dysthymia (12 month)	226
DSM-IV Dysthymia w/hierarchy (12 month)	137
DSM-IV Generalized Anxiety Disorder (12Mo)	394
DSM-IV Gen Anxiety Disorder w/hierarchy (12Mo)	261
DSM-IV Hypomania (12 month)	32
DSM-IV Intermittent Explosive Disorder (12Mo)	404
DSM-IV Intermittent Explosive Disorder w/ hierarchy (12Mo)	377
DSM-IV Mania (12 month)	190
DSM-IV Major Depressive Disorder w/ hierarchy (12Mo)	658
DSM-IV Major Depressive Episode (12Mo)	805
DSM-IV Oppositional Defiant Disorder (12Mo)	55
DSM-IV Oppositionl Defiant Disorder w/ hierarchy (12Mo)	48
DSM-IV Panic Attack (12 month)	995
DSM-IV Panic Disorder (12 month)	262
DSM-IV Posttraumatic Stress Disorder (12Mo)	326
DSM-IV Social Phobia (12 month)	652
DSM-IV Specific Phobia (12 month)	843
DSM-IV Nicotine Dependence (12 month)	312
DSM-IV Binge Eating Disorder w/ hierarchy (30Day)	28
DSM-IV Binge Any (30Day)	45
DSM-IV Bulimia (30Day)	8
DSM-IV Bulimia w/ hierarchy (30Day)	8
DSM-IV Bi-polar I (30Day)	35
DSM-IV Bi-polar II (30Day)	37

Continuation of Table A.10 - DSM Diagnosis Counts in the NCSR Kessler et al. (2004)

DSM Diagnosis	Count
DSM-IV Bi-Polar Subthreshold (30Day)	41
DSM-IV Posttraumatic Stress Disorder(30Day)	160

End of Table A.10 - DSM Diagnosis Counts in the NCSR Kessler et al. (2004)

Algorithm 1: Individual Graph Creation

```
for $subject = 1, 2, \ldots$ do
    Sort $subject$ lifetime events by age
    $levels$ = {
    $age_1$: [$event_1$, $\ldots$],
    $age_2$: [$event_1$, $\ldots$],
    $\vdots$
    }
    for $n = 1, 2, \ldots$ do
        if $n = 1$ then
            continue
        else
            for $event_n \in age_n$ do
                for $event_{n-1} \in age_{n-1}$ do
                    draw_edge_from_to($event_n$, $event_{n-1}$)
                end
            end
        end
    end
end
```