The Edward S. Rogers Sr. Department of

Electrical and Computer Engineering

University of Toronto

ECE496Y Design Project Course - Group Final Report

Machine Learning on Brain Graph

Project number: 2018937

Supervisor: Dr. Ashish Khisti

Administrator / Section number: Mr. Ross Gillett / 07

Team members:

Shi Hu

Jixiong Deng

Yuxi Cai

Emails:

rock.hu@mail.utoronto.ca

jixiong.deng@mail.utoronto.ca

yuxijune.cai@mail.utoronto.ca

Date of Submission: March 21, 2019

ECE496Y

Group Final Report Attribution Table

This table should be filled out to accurately reflect who contributed to each section of the report and what they contributed. Provide a **column** for each student, a **row** for each major section of the report, and the appropriate codes (e.g. 'RD, MR') in each of the necessary **cells** in the table. You may expand the table, inserting rows as needed, but you should not require more than two pages. The original completed, and signed form must be included in the hardcopies of the final report. Please make a copy of it for your own reference.

Section	Student Names		
	1. Jixiong Deng	2. Shi Hu	3. Yuxi Cai
Cover page, attribution table, Group Highlights and Individual Contributions	RS RD MR	RS RD MR	RS RD MR
Acknowledgements	RS RD MR	ET	MR
Executive Summary	RS	RS RD MR	RS RD MR
Introduction	RS RD MR	RS	MR
Final Design	RS	RS RD MR	RS RD MR
Testing and Verification	RS RD MR	RS	RS MR
Summary and Conclusions	RS	RS RD MR	RS MR
References	RD MR	RD MR	RD MR
Appendices	RD MR	RD MR	RD MR
All	FP CM ET	FP CM ET	FP CM ET

Abbreviation Codes:

Fill in abbreviations for roles for each of the required content elements. You do not have to fill in every cell. The "**All**" row refers to the complete report and should indicate who was responsible for the final compilation and final read through of the completed document.

RS – responsible for research of information
RD – wrote the first draft
MR – responsible for major revision
ET – edited for grammar, spelling, and expression
OR – other
"All" row abbreviations:
FP – final read through of complete document for flow and consistency
CM – responsible for compiling the elements into the complete document
OR - other
If you put OR (other) in a cell please put it in as OR1, OR2, etc. Explain briefly below the role referred to:
OR1: enter brief description here
OR2: enter brief description here

Signatures

By signing below, you verify that you have read the attribution table and agree that it accurately reflects your contribution to this document.

Name	Jixiong Deng	Signature		Date:	Mar 21, 2019
Name	Shi Hu	Signature		Date:	Mar 21, 2019
Name	Yuxi Cai	Signature		Date:	Mar 21, 2019
Name		Signature		Date:	

ECE496Y

Voluntary Document Release Consent Form[1]

To all ECE496 students:

To better help future students, we would like to provide examples that are drawn from excerpts of past student reports. The examples will be used to illustrate general communication principles as well as how the document guidelines can be applied to a variety of categories of design projects (e.g. electronics, computer, software, networking, research).

Any material chosen for the examples will be altered so that all names are removed. In addition, where possible, much of the technical details will also be removed so that the structure or presentation style are highlighted rather than the original technical content. These examples will be made available to students on the course website, and in general may be accessible by the public. The original reports will not be released but will be accessible only to the course instructors and administrative staff.

Participation is completely voluntary and students may refuse to participate or may withdraw their permission at any time. Reports will only be used with the signed consent of all team members. Participating will have no influence on the grading of your work and there is no penalty for not taking part.

If your group agrees to take part, please have all members sign the bottom of this form. The original completed and signed form should be included in the hardcopies of the final report.

Sincerely,
Khoman Phang
Phil Anderson
ECE496Y Course Coordinators

Consent Statement

We verify that we have read the above letter and are giving permission for the ECE496 course coordinator to use our reports as outlined above.

Team #: 2018937 Project Title: Machine Learning on Brain Graph

Supervisor: Ashish Khisti Administrator: Ross Gillett

Name Jixiong Deng **Signature** **Date:** Mar 21, 2019

Name Shi Hu **Signature** **Date:** Mar 21, 2019

Name Yuxi Cai **Signature** **Date:** Mar 21, 2019

Name **Signature** **Date:**

[1] This form will be detached from the hardcopy of the final report. Please make sure you have nothing printed on the back page.

Executive Summary

In summary, this project is to provide a machine learning solution which takes functional magnetic resonance imaging (fMRI) as inputs and outputs proper labels to classify inputs in different types of brain diseases.

The human brain is one of the most complex networks known to man. The field of network neuroscience still remains in its very early stage while machine learning is rising and showing its power in recent years. Given machine learning on graph-structured data is relatively new and challenging, the team is motivated to apply machine learning techniques to brain graphs to explore the possibility of improvement in this area of study.

In general, our final design can be divided into two parts. First is the data science module, which converts raw fMRI signals from ADHD-200 and ABIDE databases to brain graphs. The dataset contains three types of label: healthy, attention deficit hyperactivity disorder (ADHD) and autism spectrum disorder (ASD). The module stores the resulted adjacency matrices and labels into matrix.csv and label.csv files. The second module consists of a machine learning model based on multi-layer perceptron (MLP) algorithm. The machine learning module is capable of assigning labels containing information of the type the brain disorder to subjects according to the brain graphs with validation accuracy of 93%. The model was decided based on the implementation and training of several different algorithm, including but not limited to Support Vector Machine (SVM) and Convolutional Neural Network (CNN). MLP was selected as the final solution because it provides the highest validation accuracy given the data output by the data science module.

In the final proposal, the team came up with a number of requirements, constraints and objective to ensure the quality of the final design. The final design satisfies all the requirement and is able to give predictions of diseases for the given brain graphs with an accuracy notably higher than the accuracy of random prediction. Further improvement on the dataset is needed to include preprocessing and the result will be presented in the design fair.

Group Highlights

In general, the progress has been delayed comparing to the original plan. Most of the progress was made in the middle of the fall semester and the beginning of the winter semester. The source of delay is two-folded: one being that we failed to take into account in the time we could spend on the project especially during the exam period in the fall semester; the other being that there was a change in the source of datasets.

The major change of our project is caused by the unavailability of preferred fMRI datasets. Since extra efforts were required to investigate more datasets, the data collection task has been delayed. However, we prioritized and created new tasks in order to proceed. For the data science module, instead of collecting data before investigating fMRI conversion tools, we tested tools like Nilearn with a smaller set of data. And the prototyping of conversion pipeline has been started before the collection of datasets is finalized. For the two machine learning modules, instead of adjusting the implementation based on actual adjacency matrices, the two modules were tested with images and synthetic brain graph datasets.

All team members have made a sufficient amount of effort into the project so that by the time of the final report, the system is in shape. However, more changes are still needed before the design fair. For the data science module, the scripts for fetchData and toBrainGraph are both completed and tested. However, because of insufficient communication with the supervisor, we made a wrong trade-off between preprocessing the data and the number of diseases and sample. As a result, at the time this report is written, the data used include more than 1000 samples of healthy control and two different diseases. However, as our supervisor advised, preprocessing data is more important than including multiple diseases in the data set. Therefore we will need to preprocess at least one group of the data before the design fair. For machine learning modules, a number of possible algorithms have been examined and the accuracy results are retrieved on time. However, as the datasets change, we will need to re-access those models to see whether the changes will affect the selection of models.

Individual Contributions

Yuxi Cai

The purpose of our project is to develop a machine learning solution to classify brain diseases using functional magnetic resonance imaging (fMRI) signals retrieved from public databases. The project is divided into two parts, one is the data science module, which transforms fMRIs into adjacency matrices, the other is the machine learning module, which is trained and evaluated with the output from the previous module. My main responsibility for this project is to find sources for, design and implement the data science module for our project.

To start with, I first configured my development environment and make sure the version of the tools are compatible. Since I have not learned about parallel programming before, I did some research on and learned about the library about parallel programming in Python.

After that, I started the data collecting process by investigating the public databases recommended by our supervisor. However, the release of several datasets from the Human Connectome Project, which was considered the best fit for our project in our proposal, has been delayed. Therefore, I looked into a list of other datasets and need confirmation from our supervisor. These changes have delayed this task significantly because retrieving data from different databases means there is a higher chance of inconsistency in the data collection process so more work was done to access the quality of the databases. Meanwhile, I decided to use resting-state fMRI, which presents information about brain activity when the subject is at rest. It is because resting-state fMRIs are widely available in those databases and there is less variance in the status of the subjects among different datasets. After researching the tools for converting fMRI to adjacency matrices, I decided to use the Nilearn package because the interface is easy to understand and it has been used in a number of published studies.

By the time of the report, the data used includes more than 1000 samples of healthy control and two different diseases. However, as our supervisor advised, preprocessing data is more important than including multiple diseases in the data set. I will need to preprocessing a certain amount of data for the machine training model before the design fair. And very likely the scope of the project will shrink from distinguishing two diseases to classifying between healthy control samples and samples with one disorder.

Shi Hu

The goal of this project is to develop a machine learning software solution that can classify brain diseases based on functional magnetic resonance imaging (fMRI) signals retrieved from public databases. To better provide data to the machine learning model, the resulted system is required to first convert fMRI to brain graphs in the form of adjacency matrices. Then, these matrices are fed into pre-trained machine learning models, which shall generate predictions of diseases for the given brain graphs. In general, my responsibility is to find some effective algorithms, implement them and train them. Then, select the best one for one of the two machine learning modules.

I started with reading materials recommended by the supervisor to understand the basic ideas of brain graph. One vital idea was the null model. Null models are synthetic models used by people to analyze brain graphs, which could be easily generated and maintain some basic characteristics of the brain graph. Therefore, at the first stage, I could use them to generate my own synthetic data to test my algorithms before I could have access to real data.

After that, I did research on algorithms may be suitable for this project. Four different algorithms came into my scope. MLP, CNN, DeepWalk and SDNE. MLP (multi-layer perceptron) was a typical basic neural network. CNN (convolutional neural network) was typically used to classify images. As our adjacency matrices could be viewed as mono-channel images, CNN was worth trying. DeepWalk and SDNE were two kinds of graph embedding techniques, which were designed especially for extracting information from graphs. I first tested them on synthetic data. All of them have good performance. Then, after real data were available, I trained and tested them on real data. However, this time only MLP could provide decent results, which was also chosen as our final model.

At this point, my current results support that MLP would be the best choice for our project. However, as suggested by our supervisor, if the data could be well-preprocessed, there may be a different result. Also, it is still expected that some techniques may be appended to MLP to improve its baseline. Additionally, our supervisor suggested an optional task that it could be an interesting idea to cluster the data into several sub-categories. Thus, before the design fair started, I may still work on some of these things.

Jixiong Deng

The goal of our project is to develop a machine learning solution to classify brain diseases using functional magnetic resonance imaging (fMRI) signals retrieved from public sources. The project is divided into two parts, one is the data science module, which transforms fMRIs into adjacency matrices, and the other one is the machine learning module, which is trained and evaluated with the output from the data science module. In our project, I am mainly responsible for developing machine learning algorithms in order to predict disease type by given brain graph data.

Machine Learning is widely used as the automation method and covers lots of various learning mechanisms. To achieve our goal, I need to explore learning algorithms and choose the most suitable models with optimal hyperparameters. I followed instructions from our supervisor. And I found I could not avoid implementing a python project with most popular library in this field, Tensorflow after my inspection in order to establish a strong solution. So it is necessary to make all environment programs and configurations are well set on my local working platform.

When all background situations are set down, I started to implement a supervised learning algorithm named Support Vector Machine (SVM), which can produce a binary decision boundary for multidimensional linear classifier. After long time debugging process, SVM could have an average predictive accuracy around 90% on binary test dataset with well-tuned hyperparameters. My next goal is to optimize data processing part in SVM in order to reduce overfitting phenomenon and increase the predictive accuracy. Because of the good result of our current progress, even higher than our objective, and the delay of our brain graph dataset, we decided not to implement Generative Adversarial Networks.

Besides that, I also attempted to implement a probabilistic reasoning solution named Expectation-Maximization algorithm. By the time of the report, the algorithm is on debugging process and almost down with the prior of bernoulli distribution. Once the current algorithm finished, I will try to implement an advanced version by replacing the prior of bernoulli distribution with the prior of high dimensional Gaussian distribution. The idea is called Gaussian Mixture Models (GMM) and I believe when the more data feed in GMM, the predictive accuracy will be better than our current solution.

Acknowledgments

“Stars, bright night sky; many people, wisdom wide.”

On the very outset of this report, we would like to extend our sincere & heartfelt obligation towards all the personages who have helped us in this endeavour. Without their active guidance, help, cooperation & encouragement, we would not have made headway in the project.

We would like to express our deepest appreciations to our project supervisor, Professor Ashish Khisti. His exemplary guidance, constant encouragement, and careful monitoring throughout our research process provides us with a brilliant pathway and leads us successfully completing our project.

We would like to sincerely thank our project administrator, Dr. Ross Gillett, who always generously guides us to achieve our goals, helps us when we met difficulties and educates us to be great engineers in our future careers, with our most profound gratitude.

We gratefully acknowledge the ADHD-200 Consortium and Autism Brain Imaging Data Exchange (ABIDE) for generously sharing their data with the scientific community.

In addition, we gratefully acknowledge all faculties in University of Toronto for giving us a great opportunity and supports to embark our project with such an amazing experience.

Table of Content

1. Introduction

This report summarizes the motivation, design, implementation and testing of “Machine Learning on Brain Graph” as part of our final year design project course ECE496. The report concludes with suggestions of improvements and future work.

1.1 Background and Motivation (Author: Yuxi Cai)

The human brain is one of the most complex networks known to man while the field of network neuroscience, network science of the brain, still remains in its very early stage [1][2]. Given brain’s network-like structure, many types of research have been conducted on utilizing network sciences and graph theory methods to represent connectivity of different parts of the brain and to ultimately map its structures to functions [3]. One way to analyze the connectivity is to utilize adjacency matrices, which are 2-dimensional matrices that are either binary or weighted and can indicate the connectivity between two nodes on a system [4]. On the other hand, degenerative nerve diseases involving brain disorders can be serious to the extent of life-threatening depending on the type, while most of them have no cure [5]. Patients with those diseases, such as Alzheimer’s Disease, can be indicated by abnormal functional brain networks [6]. Therefore, brain graphs have been utilized in studies of different diseases, including the attempt to diagnose Alzheimer's disease, schizophrenia, and epilepsy [7].

Additionally, the classification of disease based on brain graphs utilizing machines learning is still rarely employed in the current stage [8]. Machine learning, being a branch of Artificial Intelligence based on the idea that systems are able to learn from data and extract patterns with minimal human interventions, is a data analysis technique that builds analytical model automatically [9]. Recent papers on a machine learning model differentiating among diseases are usually estimating the severity and probability of one disease or determining whether the subject carries any of the target diseases [8][10]. Future improvement will be focussing on improving those researches so that the machine learning model will be able to distinguish among different diseases.

As an imaging technique, functional magnetic resonance imaging (fMRI) is widely employed in brain disease diagnosis and studies because of its reliability in revealing the activity level of different brain regions [11]. A number of publicly shared datasets of fMRI raw data are available for research uses [12].

1.2 Project Goal (Author: Jixiong Deng)

The goal of this project is to develop a machine learning solution to classify brain diseases using functional magnetic resonance imaging (fMRI) signals retrieved from public sources [11].

1.3 Project Requirements (Author: Jixiong Deng)

Table 1. Project requirements

ID	Requirements	Description
1.1	Output: Brain Graph Matrix.	**Primary Functional Requirement:** The design shall be able to convert fMRI signals to brain graphs, which are adjacent matrices representing the connectivity among parts of human brains[13][14][15][16].
1.1a	Valid Brain Graph format: Diagonal matrix.	**Sub Functional Requirement:** Output brain graph is a diagonal matrix and represents strength of connections.
1.1b	Information consistency between brain visualization and brain graph matrix.	**Sub Functional Requirement:** Information of brain graph shall match with visualization of fMRI signal.
1.2	Output valid prediction: Label as 0 or 1 for each single brain graph matrix.	**Primary Functional Requirement:** The design's machine learning models shall generate predictions of diseases for given brain graph converted by fMRI signals.
1.3	Accuracy on test dataset: 50% or higher.	**Primary Functional Requirement:** The accuracy of machine learning models' predictions on test dataset* shall be higher than random prediction, which is 50% for binary classifier.
2.1	Limit of training time: Within 1 hour.	**Constraint:** Maximum training time for the model must be within 1 hour under GTX 1080 and i7 8700K platform.

2.2	Limit of Random-Access Memor : Within 8GB.	**Constraint:** Converting process from fMRI to brain graph must not cause a stack overflow on 8GB's random-access memory.
2.3	Compatibility of python and tensorflow.	**Constraint:** The design must run on any environment with Python 3.6 and TensorFlow r1.11.
2.4	Size of dataset: 200 or higher.	**Constraint:** The number of generated brain graph database must be at least with a size of 200.
3.1	Accuracy on test dataset: 65% or higher.	**Objective:** The accuracy of machine learning models' predictions on test dataset* should be higher than 65% [11].
3.2	GAN improvements.	**Objective:** Applying Generative Adversarial Networks (GAN) should improve the accuracy of the design on test dataset* [10].

Note: *Test dataset accuracy is the metric that is used to evaluate machine learning models.

Corresponding tests and results will be talked in Section 3.

2. Final Design

2.1 System-level Overview (Author: Shi Hu)

To present the initial technical design from the higher level, this section includes a system block diagram of the main modules as shown in Figure 1 for visualization and explains the functionality of each module. This system block diagram is from our final proposal, including an optional GAN generator that was planned to implement to improve the accuracy. However, as of the day of the report, it is not implemented as a part of the project and Figure 2 is the implemented system block diagram.

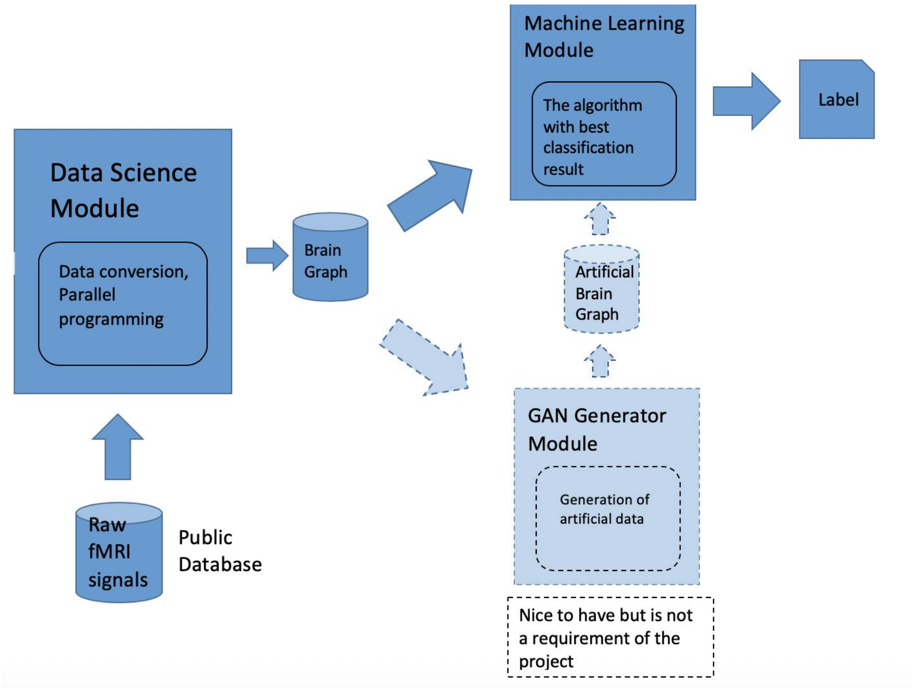

Machine Learning Module
The algorithm with best classification result
Label
Data Science Module
Data conversion, Parallel programming
Brain Graph
Artificial Brain Graph
GAN Generator Module
Generation of artificial data
Raw fMRI signals
Public Database
Nice to have but is not a requirement of the project

Figure 1: System Block Diagram of the project including optional module.

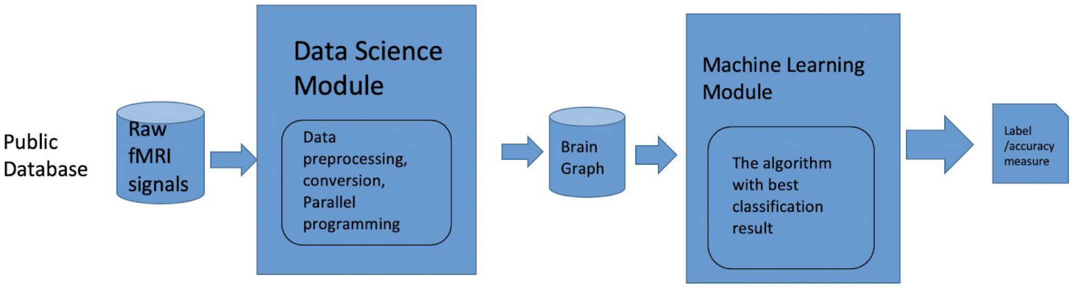


Figure 2. system block diagram implemented.

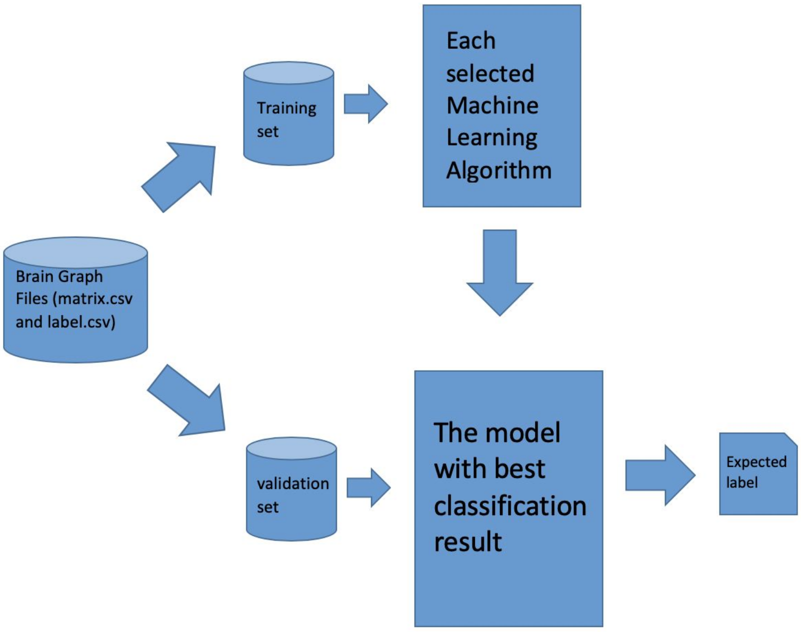

Training set
Each selected Machine Learning Algorithm
Brain Graph Files (matrix.csv and label.csv)
validation set
The model with best classification result
Expected label

Figure 3. Detailed block diagram for the machine learning module.

In general, this project has two key modules: data science model and machine learning module. The two modules are connected as a chain to form the whole system. The data science module takes raw fMRI signals as input and outputs brain graphs, which will be the input for the machine learning module. Then, the machine learning module will predict whether a brain has a certain type of disease or not.

The Data Science module fetches raw fMRI signals from public databases, then converts the signals into brain graph in the form of adjacency matrices. This is done by an automation script, existing API for converting fMRI signals to brain graphs, and involving parallel programming considering the size of the source files [13][14][15].

The Machine Learning Module, as shown in Figure 3, trains a model for each possible machine-learning algorithm and selects the model having the best performance. The whole selection procedure and rationale is detailed later in section 2.2.2. The resulted model takes brain graphs as input and outputs the label providing information about possible brain disorder of the subject.

The optional GAN Generator Module takes in existing brain graphs so that after learning generates artificial brain graphs which amply the input for the Training Module.

2.2 Module-level Descriptions

2.2.1 Data Science Module (Author: Yuxi Cai)

Inputs are ftp links to archived fMRI images, including resting state fMRI images, in .gz file type. The module outputs adjacency matrices and labels in forms of .csv files. This module contains the automation scripts for firstly, grepping raw fMRI signals from public databases; secondly, preprocessing and labelling the images with corresponding labels; lastly, converting the raw fMRI signals to and storing the resulted adjacency matrices and corresponding label into .csv files, as illustrated in Figure E-4 in Appendix E . A partial example of the .csv file is can be found in Figure E-3 of Appendix E. Another intuitive way to view these matrices are plotting them as images as shown in Figure 4.

It is notable that the mask used for correlation calculation is the 48 cortical regions defined by the Harvard-Oxford parcellations [16]. This atlas mask is well-studied, used by a number of publications and is available in several public sources [17]. The reason for using cortical region instead of subcortical is because the parts of the brain that are in charge of motor and cognitive activities are the cortical region, which are likely to be affected by the brain disorders [18].

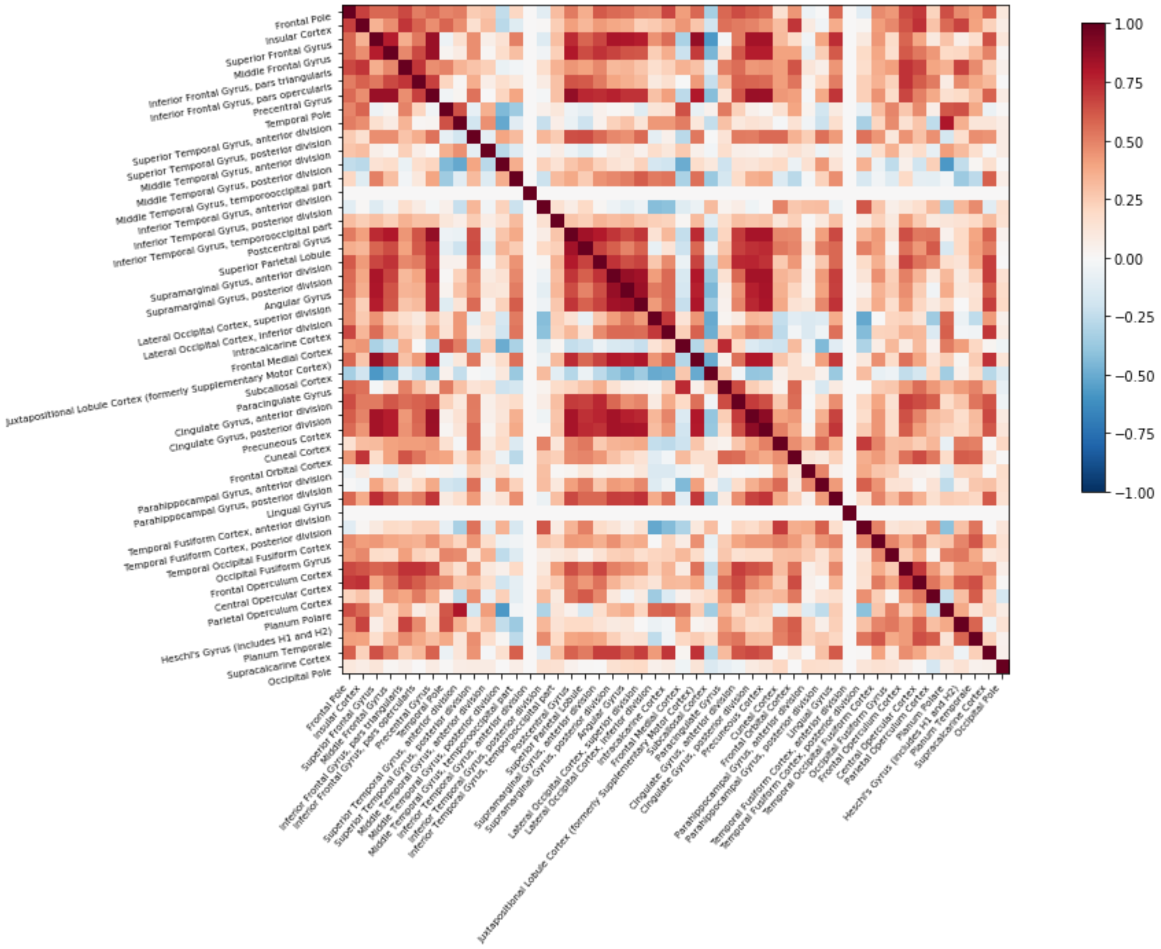


Figure 4. Visualization of the converted matrix of an ADHD sample

2.2.2 Machine Learning Module (Author: Shi Hu)

Inputs are brain graphs in the form of adjacency matrices. Final outputs are labels providing information about possible brain disorder of the subjects. First, as shown in Figure 2, this module separates the incoming dataset into a training set and a validation set. Then, it trains a machine learning model for every alternative, based on the training set. Each model is capable of assigning labels containing information of the type the brain disorder to subjects. The validation set is not used to train the model but verify how well the model does on "unseen" data. Finally, the model having the best performance on validation set is chosen. Afterall, if a new sample comes in, the module could directly use the selected model to make a prediction, instead of going through the whole process again.

The major work involved here are designing algorithms and selecting the best model. There are five candidate algorithms, Support Vector Machine (SVM), multi-layer perceptron

(MLP), Convolutional Neural Network (CNN), DeepWalk[19] and Structural Deep Network Embedding (SDNE)[20]. SVM is a traditional algorithm proven strong as a classifier. MLP is the basic version of neural network, which is proven to be very effective to do classification. CNN is typically used to classify images. However, since an adjacency matrix can be viewed as a mono-channel image, as shown in Figure 4, it is not a bad idea to try CNN out. Both of the last two algorithms belong to the area of graph embedding techniques. They could compute a vector representation for each vertex of a graph. Then, these vectors could be fed into another simple MLP model to do the classification work. In order to better tune these models to get a better performance for each of them. Learning curves like Figure 5 are plotted to assist us. Some actual learning curves are included in Appendix F. The results of all models are shown in Table 2.

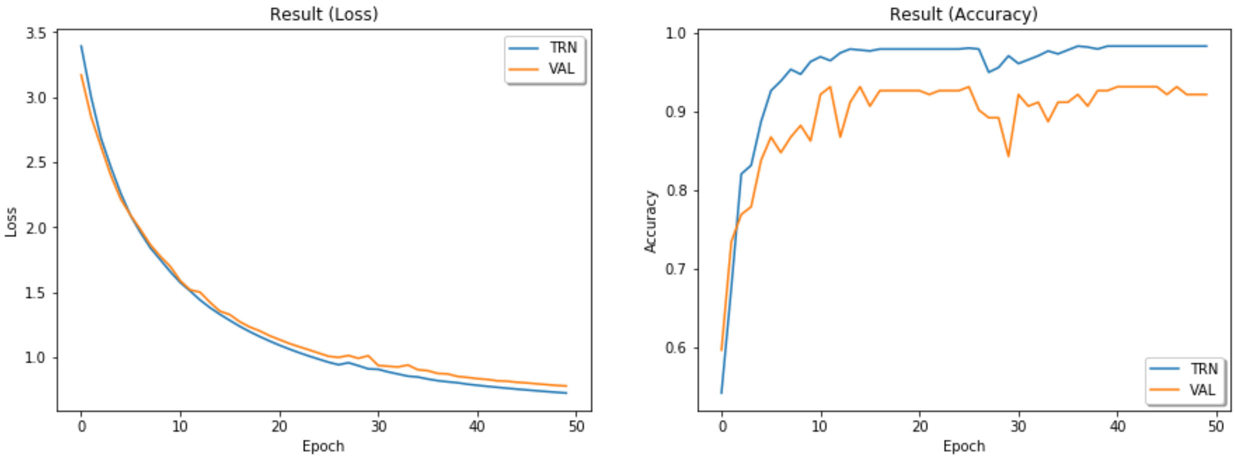


Figure 5. A sample of learning curves

Table 2. Performance of each model

Algorithm	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuracy**
SVM (Binary class)	0.06107	99%	0.46986	88%
MLP	0.747962	98%	0.801159	93%
CNN	0.807990	58%	0.802296	61%
DeepWalk + MLP	0.690240	94%	1.123779	48%
SDNE + MLP	0.620779	98%	1.046600	53%

As shown in Table 2, MLP has the best performance based on validation accuracy. Therefore, it is selected as the final model to use.

2.3 Assessment of Final Design (Author: Shi Hu, Yuxi Cai)

The data science module finally uses Nilearn [13] to fetch the Harvard-Oxford parcellations atlas and do correlation calculation. It is recommended by our supervisor and it is well developed compared to other alternatives. It is also our original first choice and turns out to be powerful enough. It converts raw fMRI signals to correlation matrices efficiently, and provides an intuitive way to visualize these matrices. It could also provide concrete labels which show exactly which brain regions are interactive, as shown clearly in as shown in Figure 4. For dataset solution, going through the datasets suggested by our supervisor, we found two suitable datasets that can be used for our project, which are the ADHD-200 database and the Autism Brain Imaging Data Exchange (ABIDE) database. The selection was done based on a set of criteria decided by the team, which can be found in Appendix D. The data used in this report consists of 104 healthy control, 99 individuals with Autism or autism spectrum disorder (ASD) collected from three different sites retrieved from ABIDE database; 469 healthy control, 342 individual with attention deficit hyperactivity disorder (ADHD) collected from five different sites retrieved from the ADHD-200 database. However, as per the date of the report, all the evaluation was done with raw data, which is subjected to noises as defined in Table D-1 in Appendix D. Further improvement is needed.

In terms of the machine learning module, we finally choose the model based on MLP algorithm, which is sort of unexpected in the beginning. Since the major challenge of the project is performing machine learning on graph-structured data, it is assumed that more specialized algorithms, such as DeepWalk and SDNE, would have better performance because they are designed to extract information from graphs. Also, when tested by synthetic data earlier, these algorithms showed good potential and had a good performance. One of the reasons these algorithms do not work well on real-world data might be that, compared to synthetic data, real data have noises and they may interfere the process of training. Another possible reason is that, compared to synthetic data, real data are much more complicated; thus, the structural information hidden in those graphs could not be easily extracted by these algorithms. Therefore, though they were us proposed solutions and developing and testing them were very time consuming, MLP algorithm is eventually selected based on the testing results.

In conclusion, the data science module is well developed and do required work well. The machine learning module, though deviates from the proposed solution in the beginning, still achieves its goal by selecting the best model based on our current results.

3. Testing and Verification

3.1 Verification table or Validation matrix (Author: Jixiong Deng)

The essential metric for the design is accuracy on test dataset, which is the most widely used for machine learning models. In general, the dataset is randomly split into training dataset, validation dataset and test dataset, commonly with portions 0.70, 0.15 and 0.15. Machine learning models are trained on the training dataset and optimized based on the accuracy on validation dataset. After training, machine learning models only run once on test dataset and the accuracy on test dataset stands for the performance of machine learning models.

In our design, functional requirements the acceptance tests.

Table 3. Details of verification tests

ID	Requirement	Verification Result and Proof	Requirement Verification Method			
			Similarit y	Review of Design	Analysis	Test
1.1	Conversion from fMRI signal to brain graph.	**TEST:** Visualize brain graphs after converting and compare brain graphs with their original labels. **PASSED** (Appendix C-2 & E)	X			
1.1a	Output brain graph is a diagonal matrix.	**TEST:** The output brain graph is an N * N matrix; the matrix is symmetric with the diagonal line from left top to right bottom. **PASSED** (Appendix C-2 & E)		X		
1.1b	Information matchup between brain visualization and brain graph matrix.	**TEST:** Check the connectivity on brain visualization and brain graph matrix. **PASSED** (Appendix C-2 & E)				X
1.2	Output predictions.	**TEST:** Record the prediction for each input brain graph and check whether predictions are valid. The valid prediction for classifier should be 0 or 1. **PASSED** (Appendix C-3)		X		
1.3	Accuracy on test dataset higher than random picks.	**TEST:** Direct measurement of accuracy on test dataset, and compare the accuracy with the expected probability of random picks. **PASSED** (Appendix C-4 & F)			X	

2.1	Training time within 1 hour.	**TEST:** Direct measurement by timer under the specific platform. **PASSED** (Appendix C-4)		X		
2.2	Stack overflow on 8GB's RAM.	**TEST:** Direct observation on runtime. The test fails, when converting process crashes. **PASSED** (Appendix C-5)		X		
2.3	Compatibility of python and tensorflow.	**TEST:** Direct observation on compiling time. The test fails, when the compilation process fails on different computers with the same version of python and tensorflow. **PASSED** (Appendix C-4)		X		
2.4	Size of dataset need to be more than 200.	**TEST:** Check the number of valid label-matrix pairs to be more than 200. **PASSED** (Appendix C-6)		X		
3.1	Accuracy on test dataset higher than 65%.	**TEST:** Direct measurement of accuracy on the test dataset. **PASSED** (Appendix C-4 & F)				X
3.2	GAN improvements (Optional).	**TEST:** Run machine learning models on test dataset before applying GAN, and then run machine learning models using GAN on test dataset again. Compare the accuracies and check the progress made by GAN. **UNTESTED**			X	

3.2 Final test results (Author: Jixiong Deng)

To test the solution, we have to evaluate our solutions based on the project requirements. When our team implemented the project, we separately developed different modules and unified the interfaces of each of them. Thus, we can evaluate all modules independently. Below is a table of the all tests' results:

3.2.1 System test results (Author: Jixiong Deng)

Table 4. Final result of system-level test

Tests Name	**Target Specification**	**Final Result**	**Compliance**
Accuracy	Functional requirement: 50% Objective: 65%	Average: 93%	Passed and met objective
Connectivity between modules	Data science module: be able to store converted matrices and labels into .csv file. Machine learning module: be able to read data from .csv file	Data science module is able to persist converted matrices and labels in the same order in two separated .csv files. Machine learning module is able to read from .csv files and shuffle the data for training and verifying	Passed
Compatibility with Python & Tensorflow	Compiled and ran successfully	Compiled and ran successfully	Passed
Stack Overflow	Error free with 8GB random-access memory	No error appears	Passed
Training Time	Within 1 Hour	Average: 65 seconds	Passed and met objective

3.2.2 Module-level testing results (Author: Yuxi Cai, Jixiong Deng)

Table 5. Module 1 Test 1 - fetch data

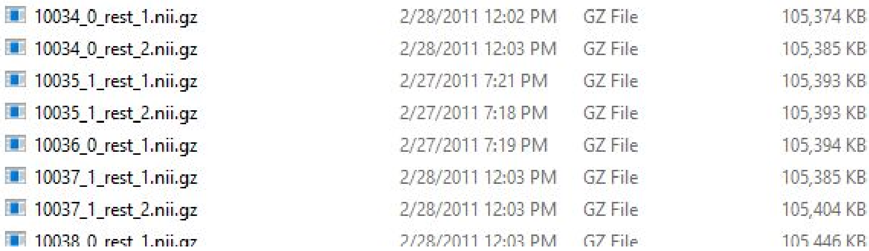
Module 1 - Data Science Module - Test 1
Description: The data science module takes fMRI data as input and produce brain graphs in the form of adjacency matrices stored in csv files. Test 1 makes sure that the module is able to download fMRI data from public databases by making ftp call, then to label each of the resting state fMRI image with the correct label.
Expected behavior by test 1:

Input(multiple fMRI data archived in .gz files, phenotype information in .csv files) ->

Output(multiple .nii fMRI images with corresponding labels)

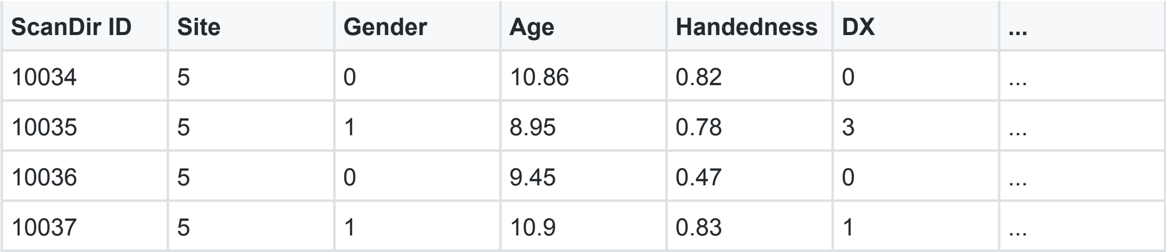
Testing procedures:
1. Add.ftp links for fMRI images .gz file and the corresponding phenotype .csv to the .url file
2. Run the fetchData script
3. Check the \data folder to see if the correct number of .nii file are extracted
4. Randomly pick 5 .nii file and record the ids and labels, then compare to the id and label pairs in the phenotype file

Final Result: Pass
The correct number of .nii.gz file is extracted as reported by the script output as shown in figure 5-1. The extracted matrice id is mapped to the correct id and label as shown in table X.



```
10034_0_rest_1.nii.gz    2/28/2011 12:02 PM    GZ File    105,374 KB
10034_0_rest_2.nii.gz    2/28/2011 12:03 PM    GZ File    105,385 KB
10035_1_rest_1.nii.gz    2/27/2011 7:21 PM     GZ File    105,393 KB
10035_1_rest_2.nii.gz    2/27/2011 7:18 PM     GZ File    105,393 KB
10036_0_rest_1.nii.gz    2/27/2011 7:19 PM     GZ File    105,394 KB
10037_1_rest_1.nii.gz    2/28/2011 12:03 PM    GZ File    105,385 KB
10037_1_rest_2.nii.gz    2/28/2011 12:03 PM    GZ File    105,404 KB
10038_0_rest_1.nii.gz    2/28/2011 12:03 PM    GZ File    105,446 KB
```

Figure 5-1. extracted .nii.gz files by the test, images are correctly labelled according to Table 5-1.
Table 5-1. NYU_phenotypic.csv that contains the phenotype information for the downloaded subjects [30]. Note that the DX(diagnosis) is: 0-Typically Developing Children, 1-ADHD-Combined, 2-ADHD-Hyperactive/Impulsive, 3-ADHD-Inattentive. Hence 0 corresponds to 0 and 1/2/3 corresponds to 1 in our label file.



ScanDir ID	Site	Gender	Age	Handedness	DX	...
10034	5	0	10.86	0.82	0	...
10035	5	1	8.95	0.78	3	...
10036	5	0	9.45	0.47	0	...
10037	5	1	10.9	0.83	1	...

Table 6. Module 1 Test 2 - store matrices

Module 1 - Data Science Module - Test 2
Description: The data science module takes fMRI data as input and produce brain graphs in the form of adjacency matrices stored in csv files that are readable by machine learning module. Test 2 makes sure that the module is able to convert .nii fMRI images to adjacency matrices and is able to store the adjacency matrices and the labels in the correct order in .csv files.
Expected behavior by test 2:

Input(labeled fMRI images in .nii file type, the output from the previous test) ->

Output(Brain Graph Matrices stored in .csv file and label stored in another .csv file)

Testing procedures:
1. Take the output from Module 1 Test 1
2. Run the ToBrainGraph script
3. Check that new data are available in the correct Label.csv file and the Matrix.csv file

Final Result: Pass
The output matrices and the corresponding labels are available in the two .csv file. Table X shows the label for the output matrices, it contains the same information as shown in figure x.

Table 6-1. a part of the label.csv file resulted from the test. 0 means healthy control, 1 means ADHD patient. Note that there are multiple occurance of the same id because of multiple fMRI data included for the same subject.

label	id
0	10034
0	10034
1	10035
1	10035
0	10036
1	10037
1	10037

Table 7. Module 2 Test

Module 2 - Machine Learning Module
Description: The machine learning module takes brain graph data from data science module as input and produce different predictions based on various machine learning mechanisms.
Expected behavior:

Input(Brain Graph Matrix) -> [Machine Learning Module] -> Output(Prediction)

Testing procedures:
1. Obtained a set of brain graph matrices as training dataset
2. Run machine learning module
3. Trained various machine learning mechanisms with specific iterations
4. Obtained another set of graph matrices as testing dataset
5. Predicted labels for testing dataset
6. Checked whether the label is valid (label must be either 0 or 1)

7. Compared predictions and ground truths and checked the accuracy of prediction

Final Result: Pass
The predictions of label are all valid, and the average accuracy is higher than both functional requirements and objectives. The detailed test results of machine learning module is shown.

Algorithm	**Training Accuracy**	**Validation Accuracy**	**Meet Constraint**	**Meet Objective**
SVM (Binary class)	99%	88%	Yes	Yes
MLP	98%	93%	Yes	Yes
CNN	58%	61%	Yes	No
DeepWalk + MLP	94%	48%	No	No
SDNE + MLP	98%	53%	Yes	No

Table 8. Module 3 Test

Module 3 - GAN Generator Module (Optional)
Description: The Generative Adversarial Network generator module takes brain graph data from data science module and produces artificial brain graph matrix as the input of machine learning model in order to improve the accuracy of predictions.
Expected behavior:

Input(Brain Graph Matrix) -> [GAN Generator Module] -> Output(Artificial Brain Graph Matrix)

Testing procedures:
1. Obtained a training dataset
2. Run GAN Generator Module with specific iterations
3. Obtained the artificial brain graph matrix generated by GAN Generator Module
4. Checked validation of generated brain graph matrix (Matrix should be 48*48 cells and diagonal)

Final Result: Untested
The accuracy of machine learning module is much higher than objective. Considering the difficulty of implementation and shortage of time, we decided not to implement the GAN Generator Module.

4. Summary and Conclusions (Author: Yuxi Cai)

To summarize, as an outcome of this project, a machine learning model has been developed to classify brain diseases based on fMRI data. The team utilized public databases, existing Python image conversion tools and machine learning packages to guarantee the reliability of the final design.

In general, we met all the goals and requirement defined in our final proposal. The final system is able to first convert fMRI to brain graphs in the form of adjacency matrices. Then, it can generate predictions of diseases for the given brain graphs with an accuracy higher than the accuracy of random prediction. Various tests were applied throughout the lifetime of the project. For example, the data science module can successfully download fMRI images with ftp links and is able to store labels and converted adjacency matrices into two different .csv files. Those tests guarantee that the final design is able to meet all the requirement defined in the final proposal.

In conclusion, the machine learning model based on multi-layer perceptron (MLP) algorithm gives the highest prediction accuracy given three types of label. MLP is the basic version of neural network, which is proven to be effective to do classification. The resulted machine learning model is able to achieve a validation accuracy of 93%, which is significantly higher than a random guess (in our case 33%). In previous paper, Heinsfeld et al achieved the highest accuracy of 70% identifying ASD patients versus healthy control with the ADHD database [21]. Another paper on a classifier developed based on SVM for both ADHD and ASD retrieves accuracy that are less than 70% for both disease [22]. Although it seems like our accuracy is higher than the result from those previous model, the data size we used is significantly smaller than those published papers. In our case, our data consists of 104 healthy control, 99 individuals with Autism or autism spectrum disorder (ASD) collected from three different sites retrieved from ABIDE database; 469 healthy control, 342 individual with attention deficit hyperactivity disorder (ADHD) collected from five different sites retrieved from the ADHD-200 database. In both paper mentioned before used training data set that is more than 10,000 samples, which is 10 time higher than our data set. Therefore, the actual accuracy may not be as high as stated in the report. However, it does prove that MLP model improves the accuracy of classifying ASD ADHD and healthy control.

Lastly, although the current design meets the defined requirements and constraints, we made the wrong tradeoff when selecting between more types of diseases as well as data sizes and preprocessing of data. As a result the reported conclusion for the machine learning model was made based on training done with raw fMRI images. We may need to re-access the models after performing necessary preprocessing of the fMRI images. This improved result will be presented in the design fair.

Reference

[1] A. Fornito, A. Zalesky and X. Zhu, Fundamentals of brain network analysis. Amsterdam: Elsevier, 2016, p. 1.

[2] D. S. Bassett and O. Sporns, “Network neuroscience,” *National Center for Biotechnology Information*, 23-Feb-2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5485642/. [Accessed: 16-Sep-2018].

[3] F. Vecchio, F. Miraglia and P. M. Rossini, “Connectome: Graph theory application in functional brain network architecture,” *Clinical Neurophysiology Practice*, 24-Oct-2017. [Online]. Available: https://www.cnp-journal.com/article/S2467-981X(17)30027-6/pdf. [Accessed: 16-Sep-2018].

[4] A. Fornite, A. Zalesky and E. Bullmore, Fundamentals of Brain Network Analysis, 2016, p. 89-113. [Accessed: 24-Oct-2018]

[5] MedlinePlus, “Degenerative Nerve Diseases”.[Online]. Available: https://medlineplus.gov/degenerativenervediseases.html. [Accessed: 24-Oct-2018]

[6] C.J. Stam, B.F. Jones, G. Nolte, M. Breakspear and P. Scheltens, “Small-world network and functional connectivity in Alzheimer’s disease,” *National Center for Biotechnology Information*, 17-Jan-2007. [Online]. Available:https://www.ncbi.nlm.nih.gov/pubmed/16452642. [Accessed: 24-Oct-2018]

[7] H. Oniasa, A. Violb, F. Palhano-Fontesa, K. C. Andradea, M. Sturzbecherc, G. Viswanathanb and D. B.de Araujo, “Brain complex network analysis by means of resting state fMRI and graph analysis: Will it be helpful in clinical epilepsy?,” *Epilepsy & Behavior*, Sep-2014. [Online]. https://www.sciencedirect.com/science/article/pii/S1525505013006173. [Accessed: 17-Sep-2018].

[8] Y. Dodonova, S. Korolev, A. Tkachev, D. Petrov, L. Zhukov, M. Belyaev, “Classification of structural brain networks based on information divergence of graph spectra,” *ieeexplore*, 10-Nov-2016. [Online]. Available: https://ieeexplore.ieee.org/document/7738852. [Accessed: 1-Oct-2018].

[9] SAS, “Machine Learning, what it is and why it matters”. [Online]. Available: https://www.sas.com/en_ca/insights/analytics/machine-learning.html. [Accessed: 24-Oct-2018]

[10] H. Guo, M. Qin, J. Chen, Y. Xu, J. Xiang, “Machine-Learning Classifier for Patients with Major Depressive Disorder: Multifeature Approach Based on a High-Order Minimum Spanning Tree Functional Brain Network,” *Comput Math Methods Med.*, 14-Dec-2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5745775/. [Accessed: 1-Oct-2018].

[11] J. Richiardi, H. Eryilmaz, S. Schwartz, P. Vuilleumier and D. V. De Ville, “Decoding brain states from fMRI connectivity graphs,” *Neuroimage*, May-2011. [Online]. Available: https://journals-scholarsportal-info.myaccess.library.utoronto.ca/details/10538119/v56i0002/616_dbsffcg.xml. [Accessed: 16-Sep-2018].

[12] OpenfMRI, “OpenfMRI”. [Online]. Available: https://openfmri.org/. [Accessed: 17-Sep-2018].

[13] Nilearn, “Nilearn”. [Online]. Available: https://nilearn.github.io/index.html. [Accessed: 17-Sep-2018]

[14] M. Mijalkov, E. Kakaei, J. B. Pereira, Eric Westman, Giovanni Volpe “BRAPH: A graph theory software for the analysis of brain connectivity.” *PLOS ONE*, 1-Aug-2017. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0178798. [Accessed: 17-Sep-2018]

[15] M. Rubinov, O. Sporns. “Complex network measures of brain connectivity: uses and interpretations”. *Neuroimage,* Sep-2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S105381190901074X?via%3Dihub. [Accessed: 17-Sep-2018]

[16] B. He, Y. Dai, L. Astolfi, F. Babiloni, H. Yuan, L. Yang. “eConnectome: A MATLAB toolbox for mapping and imaging of brain functional connectivity”. *Journal of Neuroscience Methods,* 15-Feb-2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165027010006497?via%3Dihub. [Accessed: 17-Sep-2018]

[17] Connectome, “Connectome”. [Online]. Available: https://www.humanconnectome.org/. [Accessed: 25-Oct-2018].

[18] OpenNeuroscience, “Open Neuroscience”. [Online]. Available: https://openeuroscience.com/. [Accessed: 25-Oct-2018].

[19] B. Perozzi, R. Al-Rfou, S. Skiena. “DeepWalk: Online Learning of Social Representations”. Stony Brook University, 27-Jun-2014. [Online]. Available: https://arxiv.org/pdf/1403.6652.pdf. [Accessed: 17-Mar-2019]

[20] Daixin Wang, Peng Cui, Wenwu Zhu. “Structural Deep Network Embedding”. Tsinghua University, Aug-2016. [Online]. Available: https://www.kdd.org/kdd2016/papers/files/rfp0191-wangAemb.pdf. [Accessed: 17-Mar-2019]

[21] A. Heinsfeld, A. Franco, C. Craddock, A. Buchweitz, F. Meneguzzia, “Identification of autism spectrum disorder using deep learning and the ABIDE dataset”. NCBI, Aug-2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5635344/ [Accessed: 20-Mar-2019]

[22] B. Sen, N. Borle, R. Greiner, M. Brown. “A general prediction model for the detection of ADHD and Autism using structural and functional MRI”. NCBI. Apr-2017. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5903601/ [Accessed: 20-Mar-2019]

[23] A. Badhwar, A. Tam, C. Dansereau, P. Orban, F. Hoffstaedter, and P. Bellec, “Resting-state network dysfunction in Alzheimer′s disease: a systematic review and meta-analysis,” Feb. 2017.[Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5436069/. [Accessed: 13-Jan-2019]

[24] Vigneshwaran S, Mahanand B. S., Suresh S, Sundararajan N. Identifying differences in brain activities and an accurate detection of autism spectrum disorder using resting state functional-magnetic resonance imaging: A spatial filtering approach. Medical image analysis. 2017;35:375–389. doi:10.1016/j.media.2016.08.003.

[25] Müller, R-A.; Shih, P.; Keehn, B.; Deyoe, J.; Leyden, K.; Shukla, D. (2011). "Underconnected but how? A survey of functional connectivity MRI studies in autism spectrum disorders". Cerebral Cortex. 21 (10): 2233–2243. doi:10.1093/cercor/bhq296. PMC 3169656. PMID 21378114.

[26]K. Murphy, R. M. Birn, and P. A. Bandettini, “Resting-state fMRI confounds and cleanup,” NeuroImage, vol. 80, pp. 349–359, Apr. 2013.

[27]Harvard Medical School, “Basics of fMRI Analysis: Preprocessing, First Level Analysis, and Group Analysis”. [Online]. Available: https://ftp.nmr.mgh.harvard.edu/pub/docs/SavoyfMRI2014/fmri.april2011.pdf. [Accessed: 14-Jan-2019]

[28] 1000 Functional Connectomes Project, "1000 Functional Connectomes Project". [Online]. Available: http://fcon_1000.projects.nitrc.org/index.html. [Accessed: 7-Jan-2019]

[29] 1000 Functional Connectomes Project, "Release Notes". [Online]. Available: https://www.nitrc.org/docman/view.php/296/717/fcon_1000_ReleaseNotes.pdf. [Accessed: 7-Jan-2019]

[30] 1000 Functional Connectomes Project, "The ADHD-200 Sample". [Online]. Available: http://fcon_1000.projects.nitrc.org/indi/adhd200/index.html. [Accessed: 7-Jan-2019]

[31] 1000 Functional Connectomes Project, "Autism Brain Imaging Data Exchange". [Online]. Available: http://fcon_1000.projects.nitrc.org/indi/abide/. [Accessed: 7-Jan-2019]

[32] 1000 Functional Connectomes Project, "ABIDE DATA LEGEND". [Online]. Available: http://fcon_1000.projects.nitrc.org/indi/abide/. [Accessed: 7-Jan-2019]

[33] 1000 Functional Connectomes Project, "ABIDE II Phenotypic Data Legend". [Online]. Available: http://fcon_1000.projects.nitrc.org/indi/abide/. [Accessed: 7-Jan-2019]

[34] 1000 Functional Connectomes Project, "COBRE". [Online]. Available: http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html. [Accessed: 7-Jan-2019]

[35] 1000 Functional Connectomes Project, "Consortium for Reliability and Reproducibility (CoRR)". [Online]. Available: http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html. [Accessed: 7-Jan-2019]

[36] 1000 Functional Connectomes Project, "Quality Control: Functional Images". [Online]. Available: http://fcon_1000.projects.nitrc.org/indi/CoRR/html/qc_func.html. [Accessed: 7-Jan-2019]

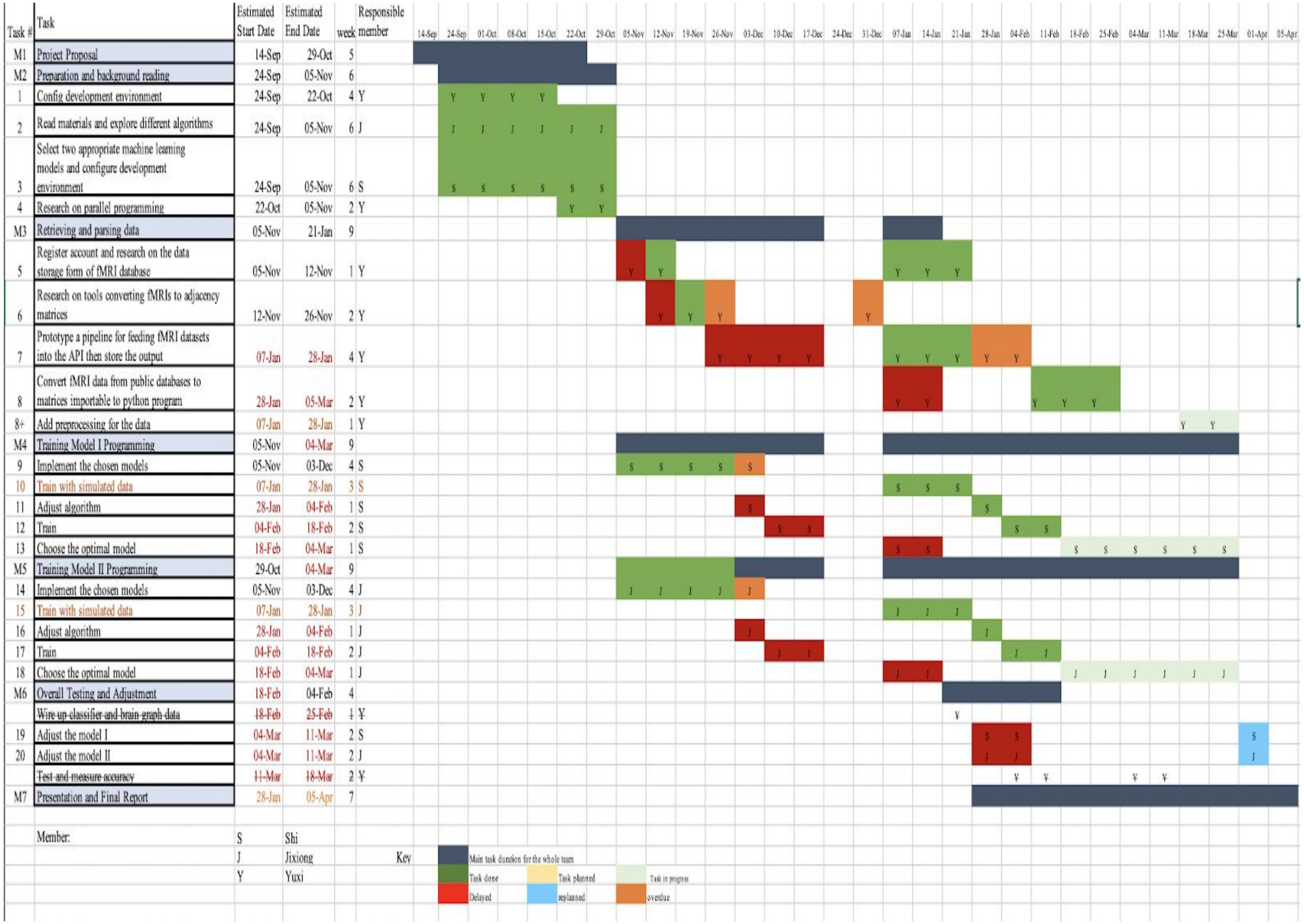
[37] OASIS, "OASIS". [Online]. Available: http://www.oasis-brains.org/. [Accessed: 14-Jan-2019]

[38] Human Connectome Project, "Human Connectome Project". [Online]. Available: https://www.humanconnectome.org/. [Accessed: 14-Jan-2019]

[39] Human Connectome Project, "HCP1200 July 2017 release of high-level rfMRI connectivity analyses". [Online]. Available: https://www.humanconnectome.org/storage/app/media/documentation/s1200/HCP1200-DenseConnectome+PTN+Appendix-July2017.pdf. [Accessed: 14-Jan-2019]

Appendices:

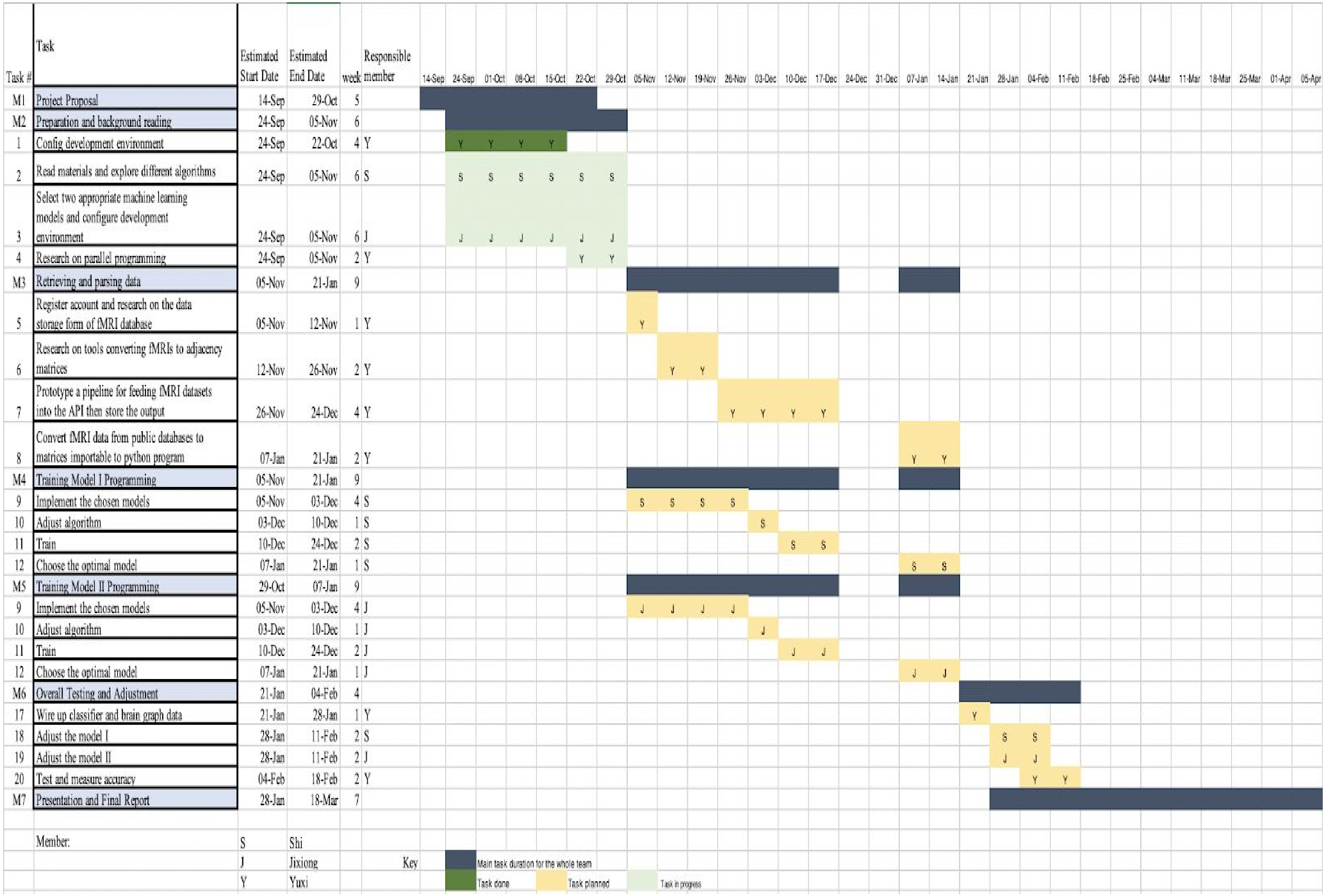
Appendix A: Gantt Chart History



Task #	Task	Estimated Start Date	Estimated End Date	week	Responsible member
M1	Project Proposal	14-Sep	29-Oct	5	
M2	Preparation and background reading	24-Sep	05-Nov	6	
1	Config development environment	24-Sep	22-Oct	4	Y
2	Read materials and explore different algorithms	24-Sep	05-Nov	6	J
3	Select two appropriate machine learning models and configure development environment	24-Sep	05-Nov	6	S
4	Research on parallel programming	22-Oct	05-Nov	2	Y
M3	Retrieving and parsing data	05-Nov	21-Jan	9	
5	Register account and research on the data storage form of fMRI database	05-Nov	12-Nov	1	Y
6	Research on tools converting fMRIs to adjacency matrices	12-Nov	26-Nov	2	Y
7	Prototype a pipeline for feeding fMRI datasets into the API then store the output	07-Jan	28-Jan	4	Y
8	Convert fMRI data from public databases to matrices importable to python program	28-Jan	05-Mar	2	Y
8+	Add preprocessing for the data	07-Jan	28-Jan	1	Y
M4	Training Model I Programming	05-Nov	04-Mar	9	
9	Implement the chosen models	05-Nov	03-Dec	4	S
10	Train with simulated data	07-Jan	28-Jan	3	S
11	Adjust algorithm	28-Jan	04-Feb	1	S
12	Train	04-Feb	18-Feb	2	S
13	Choose the optimal model	18-Feb	04-Mar	1	S
M5	Training Model II Programming	29-Oct	04-Mar	9	
14	Implement the chosen models	05-Nov	03-Dec	4	J
15	Train with simulated data	07-Jan	28-Jan	3	J
16	Adjust algorithm	28-Jan	04-Feb	1	J
17	Train	04-Feb	18-Feb	2	J
18	Choose the optimal model	18-Feb	04-Mar	1	J
M6	Overall Testing and Adjustment	18-Feb	04-Feb	4	
	~~Wire up classifier and brain graph data~~	~~18-Feb~~	~~25-Feb~~	~~1~~	~~Y~~
19	Adjust the model I	04-Mar	11-Mar	2	S
20	Adjust the model II	04-Mar	11-Mar	2	J
	~~Test and measure accuracy~~	~~11-Mar~~	~~18-Mar~~	~~2~~	~~Y~~
M7	Presentation and Final Report	28-Jan	05-Apr	7	

Member:		
	S	Shi
	J	Jixiong
	Y	Yuxi

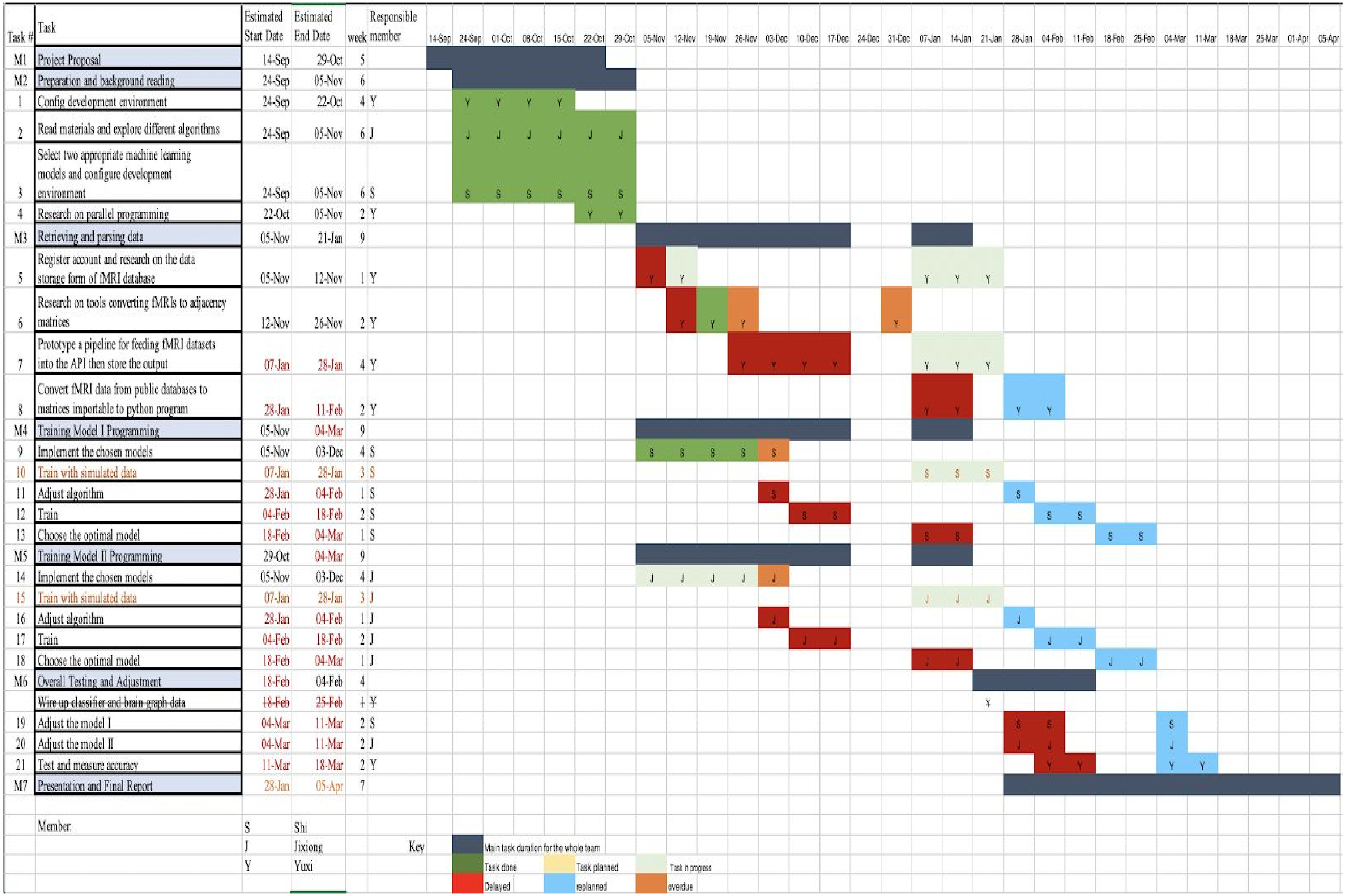
Figure A-1 Updated Gantt Chart for the project



Task #	Task	Estimated Start Date	Estimated End Date	week	Responsible member	14-Sep	24-Sep	01-Oct	08-Oct	15-Oct	22-Oct	29-Oct	05-Nov	12-Nov	19-Nov	26-Nov	03-Dec	10-Dec	17-Dec	24-Dec	31-Dec	07-Jan	14-Jan	21-Jan	28-Jan	04-Feb	11-Feb	18-Feb	25-Feb	04-Mar	11-Mar	18-Mar	25-Mar	01-Apr	05-Apr
M1	Project Proposal	14-Sep	29-Oct	5		■	■	■	■	■	■																								
M2	Preparation and background reading	24-Sep	05-Nov	6			■	■	■	■	■	■																							
1	Config development environment	24-Sep	22-Oct	4	Y		Y	Y	Y	Y																									
2	Read materials and explore different algorithms	24-Sep	05-Nov	6	S		S	S	S	S	S	S																							
3	Select two appropriate machine learning models and configure development environment	24-Sep	05-Nov	6	J		J	J	J	J	J	J																							
4	Research on parallel programming	24-Sep	05-Nov	2	Y						Y	Y																							
M3	Retrieving and parsing data	05-Nov	21-Jan	9									■	■	■	■	■	■	■			■	■												
5	Register account and research on the data storage form of fMRI database	05-Nov	12-Nov	1	Y								Y																						
6	Research on tools converting fMRIs to adjacency matrices	12-Nov	26-Nov	2	Y									Y	Y																				
7	Prototype a pipeline for feeding fMRI datasets into the API then store the output	26-Nov	24-Dec	4	Y											Y	Y	Y	Y																
8	Convert fMRI data from public databases to matrices importable to python program	07-Jan	21-Jan	2	Y																	Y	Y												
M4	Training Model I Programming	05-Nov	21-Jan	9									■	■	■	■	■	■	■			■	■												
9	Implement the chosen models	05-Nov	03-Dec	4	S								S	S	S	S																			
10	Adjust algorithm	03-Dec	10-Dec	1	S												S																		
11	Train	10-Dec	24-Dec	2	S													S	S																
12	Choose the optimal model	07-Jan	21-Jan	1	S																	S	S												
M5	Training Model II Programming	29-Oct	07-Jan	9									■	■	■	■	■	■	■			■	■												
9	Implement the chosen models	05-Nov	03-Dec	4	J								J	J	J	J																			
10	Adjust algorithm	03-Dec	10-Dec	1	J												J																		
11	Train	10-Dec	24-Dec	2	J													J	J																
12	Choose the optimal model	07-Jan	21-Jan	1	J																	J	J												
M6	Overall Testing and Adjustment	21-Jan	04-Feb	4																				■	■	■	■								
17	Wire up classifier and brain graph data	21-Jan	28-Jan	1	Y																			Y											
18	Adjust the model I	28-Jan	11-Feb	2	S																				S	S									
19	Adjust the model II	28-Jan	11-Feb	2	J																				J	J									
20	Test and measure accuracy	04-Feb	18-Feb	2	Y																					Y	Y								
M7	Presentation and Final Report	28-Jan	18-Mar	7																					■	■	■	■	■	■	■	■	■	■	■

Member:	S	Shi		
	J	Jixiong	Key	■ Main task duration for the whole team
	Y	Yuxi		■ Task done ■ Task planned ■ Task in progress

Figure A-2. Gantt Chart for the project from the final proposal



Task #	Task	Estimated Start Date	Estimated End Date	week	Responsible member
M1	Project Proposal	14-Sep	29-Oct	5	
M2	Preparation and background reading	24-Sep	05-Nov	6	
1	Config development environment	24-Sep	22-Oct	4	Y
2	Read materials and explore different algorithms	24-Sep	05-Nov	6	J
3	Select two appropriate machine learning models and configure development environment	24-Sep	05-Nov	6	S
4	Research on parallel programming	22-Oct	05-Nov	2	Y
M3	Retrieving and parsing data	05-Nov	21-Jan	9	
5	Register account and research on the data storage form of fMRI database	05-Nov	12-Nov	1	Y
6	Research on tools converting fMRIs to adjacency matrices	12-Nov	26-Nov	2	Y
7	Prototype a pipeline for feeding fMRI datasets into the API then store the output	07-Jan	28-Jan	4	Y
8	Convert fMRI data from public databases to matrices importable to python program	28-Jan	11-Feb	2	Y
M4	Training Model I Programming	05-Nov	04-Mar	9	
9	Implement the chosen models	05-Nov	03-Dec	4	S
10	Train with simulated data	07-Jan	28-Jan	3	S
11	Adjust algorithm	28-Jan	04-Feb	1	S
12	Train	04-Feb	18-Feb	2	S
13	Choose the optimal model	18-Feb	04-Mar	1	S
M5	Training Model II Programming	29-Oct	04-Mar	9	
14	Implement the chosen models	05-Nov	03-Dec	4	J
15	Train with simulated data	07-Jan	28-Jan	3	J
16	Adjust algorithm	28-Jan	04-Feb	1	J
17	Train	04-Feb	18-Feb	2	J
18	Choose the optimal model	18-Feb	04-Mar	1	J
M6	Overall Testing and Adjustment	18-Feb	04-Feb	4	
	~~Wire up classifier and brain graph data~~	~~18-Feb~~	~~25-Feb~~	~~1~~	~~Y~~
19	Adjust the model I	04-Mar	11-Mar	2	S
20	Adjust the model II	04-Mar	11-Mar	2	J
21	Test and measure accuracy	11-Mar	18-Mar	2	Y
M7	Presentation and Final Report	28-Jan	05-Apr	7	

Member:		
S	Shi	
J	Jixiong	
Y	Yuxi	

Key

■	Main task duration for the whole team				
🟩	Task done	▨	Task planned	▨	Task in progress
🟥	Delayed	▨	replanned	🟧	overdue

Figure A-3. Gantt Chart for the project from the progress report as per Jan 15th, 2019

Appendix B: Financial plan

The following is the financial plan for our proposed design including the potential capital equipment.

Table B-1. Capital Equipment Breakdown

Item	**Priority**	**Cost** (CAD)	**Quantity**	**Total Cost**	Requires Funding
IDE Software (Sublime)	1	$0	3	$0	N
Computers (without GPU & CPU)	1	$1200	3	$3600	N
GPU (GTX 1080)	1	$690	3	$2070	N
CPU (i7 8700k)	1	$553	3	$1659	N
Internet Data Plan	1	$0	3	$0	N
Other Associated software	1	$103/month	7 months	$721	N
Total Student Labour Cost				$8050	

Table B-2. Student Labours Time Breakdown

Tasks	**Jixiong** (hours)	**Shi** (hours)	**Yuxi** (hours)
Research on Brain Graph conversion	0	0	40
Research on Machine Learning Models	50	50	0
Data Retrieving and Manipulation	0	0	70
Implementation of Machine Learning Models	100	100	0
Data Adjustment	0	0	30
Stability Checking	0	0	20
Hyperparameter Justification	10	10	0
Compatibility of Machine Learning Models	20	20	20
Debugging	20	20	20
Total	**200**	**200**	**200**

Table B-3. Student Labour Cost Breakdown

Item (Group Member)	**Cost/Unit** (CAD/hour)	**Quantity** (labour hours)	**Total Cost**
Jixiong Deng	$26	200	$5200
Shi Hu	$26	200	$5200
Yuxi Cai	$26	200	$5200
Total Student Labour Cost			$15600

Table B-4. Total Cost of Project and Total Required Funding

Total Cost of Project	$23650
Total Required Funding	$0

Note: As this project's main purpose is to design a machine learning technique, it does not require any materials and funding will not be needed.

Appendix C: Validation and Acceptance Tests

Table C-1. Updated Validation and Acceptance Test

Change	**ID**	**Requirement**	**Verification Result and Proof**	**Requirement Verification Method**			
				Similarity	**Review of Design**	**Analysis**	**Test**
	1.1	Conversion from fMRI signal to brain graph	**TEST:** Visualize brain graphs after converting and compare brain graphs with their original labels. **PASSED**	X			
Added	1.1a	Output brain graph is a diagonal matrix	**TEST:** The output brain graph is an N * N matrix; the matrix is symmetric with the diagonal line from left top to right bottom. **PASSED**		X		
Added	1.1b	Information matchup between brain visualization and brain graph matrix	**TEST:** Check the connectivity on brain visualization and brain graph matrix. **PASSED**				X
	1.2	Output predictions	**TEST:** Record the		X		

			prediction for each input brain graph and check whether predictions are valid. The valid prediction for classifier should be 0 or 1. **PASSED**				
	1.3	Accuracy on test dataset higher than random picks	**TEST:** Direct measurement of accuracy on test dataset, and compare the accuracy with the expected probability of random picks. **PASSED**			X	
Modified	2.1	Training time within 1 hour	**TEST:** Direct measurement by timer under the specific platform. **PASSED**		X		
Modified	2.2	Stack overflow on 8GB's RAM	**TEST:** Direct observation on runtime. The test fails, when converting process crashes. **PASSED**		X		
	2.3	Compatibility of python and tensorflow	**TEST:** Direct observation on compiling time. The test fails, when the compilation process fails on different computers with the same version of python and tensorflow. **PASSED**		X		
Added	2.4	Size of dataset need to be more than 200	**TEST:** Check the number of valid label-matrix pairs to be more than 200 **PASSED**		X		
Deleted	3.1	Training time within 10 hours	**TEST:** Direct measurement by timer under the specific platform. **PASSED**				X
	3.2	Accuracy on test dataset higher than 65%	**TEST:** Direct measurement of accuracy on the test dataset. **PASSED**				X
Not Impleme nted	3.3	GAN improvements (Optional)	**TEST:** Run machine learning models on test dataset before applying GAN, and then run			X	

			machine learning models using GAN on test dataset again. Compare the accuracies and check the progress made by GAN. **UNTESTED**				

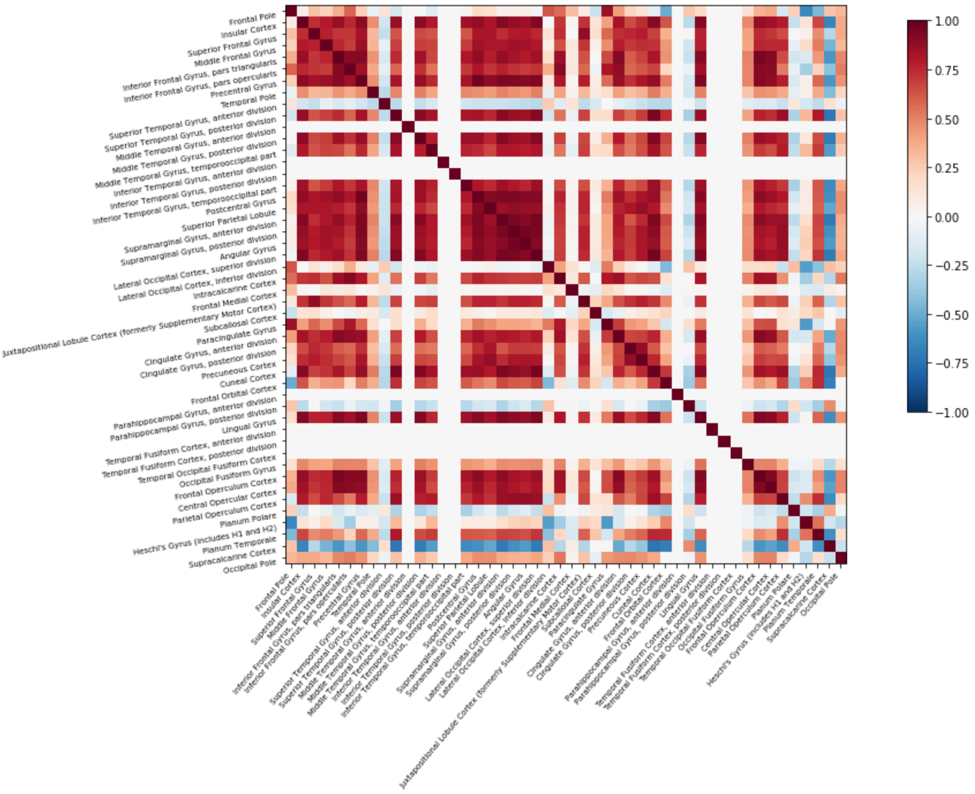
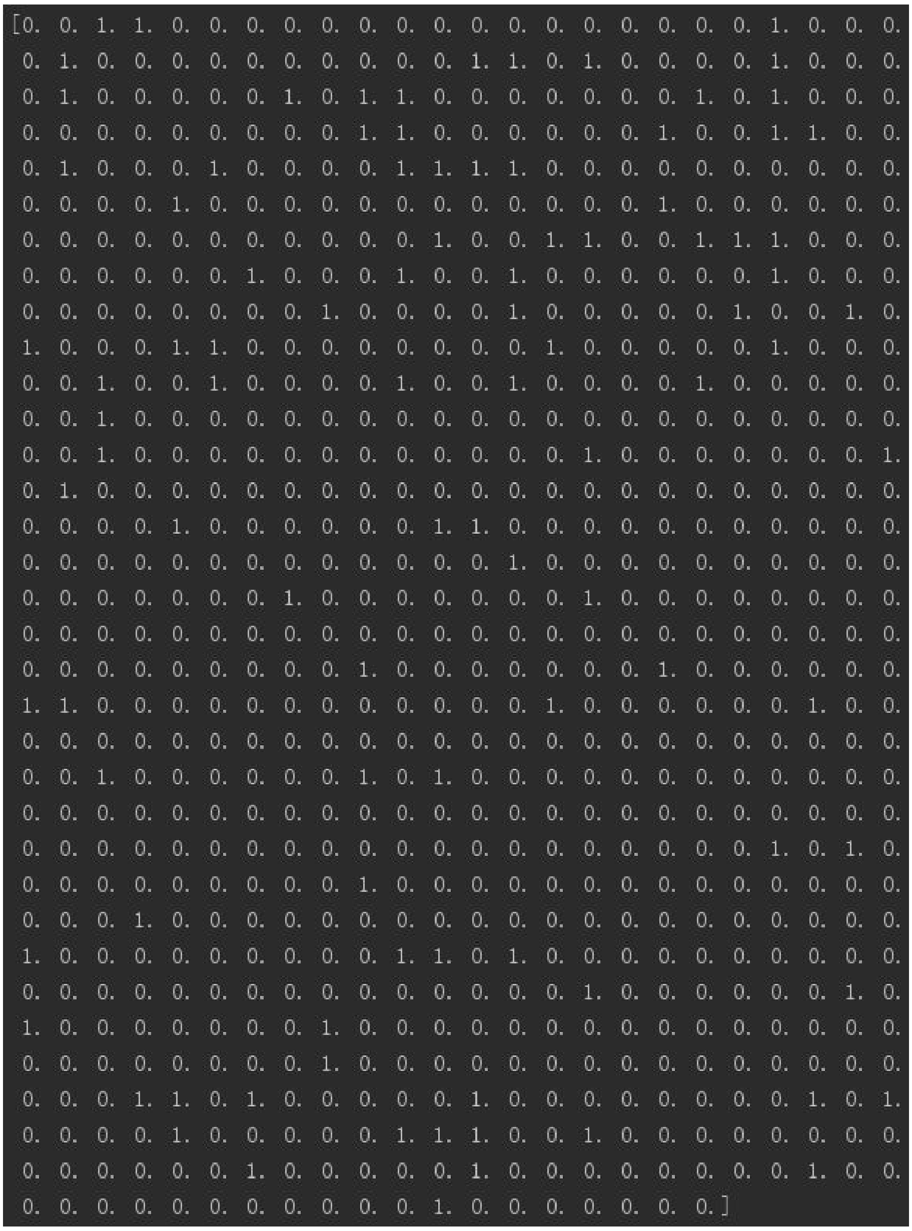
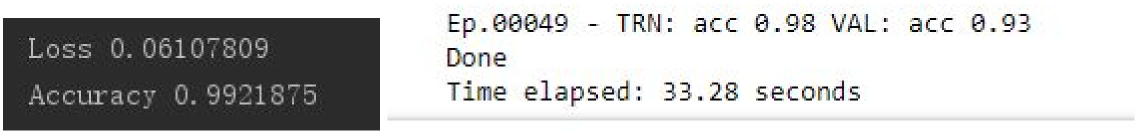


Figure C-2. Brain Graph Matrix



```
[0.  0.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.
 0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  1.  0.  0.  0.  0.  1.  0.  0.  0.
 0.  1.  0.  0.  0.  0.  0.  1.  0.  1.  1.  0.  0.  0.  0.  0.  0.  0.  1.  0.  1.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  1.  1.  0.  0.
 0.  1.  0.  0.  0.  1.  0.  0.  0.  0.  1.  1.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  1.  1.  0.  0.  1.  1.  1.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  1.  0.  0.  1.  0.
 1.  0.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  1.  0.  0.  0.
 0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.
 0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  1.
 0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.
 1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  1.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  1.  0.
 1.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  1.  1.  0.  1.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  1.
 0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  1.  1.  1.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.
 0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.
 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.]
```

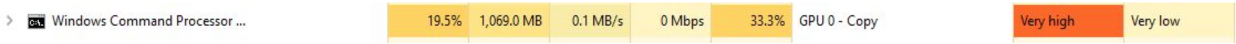
Figure C-3. Prediction Output List (All are 0 and 1)



```
Loss 0.06107809
Accuracy 0.9921875
```

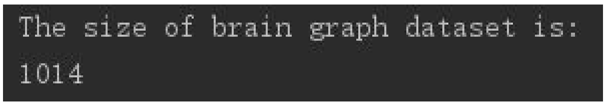
```
Ep.00049 - TRN: acc 0.98 VAL: acc 0.93
Done
Time elapsed: 33.28 seconds
```

Figure C-4. Training Time and Accuracy



Windows Command Processor ...	19.5%	1,069.0 MB	0.1 MB/s	0 Mbps	33.3%	GPU 0 - Copy	Very high	Very low

Figure C-5. Random-Access Memory Usage



```
The size of brain graph dataset is:
1014
```

Figure C-6. The Size of Brain Graph Dataset

Appendix D: Objectives for datasets selection

Table D-1: Objectives for datasets and the corresponding reasoning

Objectives for selecting datasets	Reasoning
Must include resting-state fMRI data	In order to study the influence of disorders on brain graph connectivity, we need to make sure the subjects being recorded are in a similar state. Furthermore, in current studies, it has been proven that connectivity in resting-state functional magnetic resonance imaging (rsfMRI) holds promise in the diagnosis of the disorders of interest [23][24][25].
Must have preprocessing techniques clearly outlined	During acquisition of the fMRI data, confounds can be caused by a number of sources in the MRI environment. This includes motions and physiological noises [26]. The following preprocessing are required to be done as suggested in [27]: 1. Motion Correction 2. Slice-Timing Correction 3. B0 Distortion Correction 4. Spatial Normalization 5. Spatial Smoothing However, if the related correction parameters are available, the datasets can also be used after processing in our data

	science module.
Must have data acquisition parameters outlined	This enables comparison among different datasets.
If it includes samples with disorders, must have relevant labels included	This is crucial for creating the training set and the validation set.

Table D-2: evaluation of the suggested databases

Database name	Sample size	Preprocessing and quality control	data acquisition parameters	class	Available
1000 Functional Connectomes Project [28]	1200+	Differs among individual uploads, also lack of documentation of preprocessing. [29]	RPI orientation with some exception	Differs among individual projects	Partially
ADHD 200 [30]	491 typically developing + 285 ADHD	Preliminary quality control assessments (usable vs. questionable) based upon visual timeseries inspection, preprocessed versions available	-	Typically Developing Children, ADHD-Combined, ADHD-Hyperactive/Impulsive, ADHD-Inattentivestatus	Raw data Requested, preprocessed available
Autism Brain Imaging Data Exchange (ABIDE) [31]	I: 539 individuals with ASD and 573 controls II: 521 individuals with ASD and 593 controls	Quality matrix provided, preprocessed versions available	-	No clear label but each sample comes with a list of testing scores [32][33]	Raw data Requested, preprocessed available

COBRE [34]	72 with Schizophrenia and 75 healthy controls	Not mentioned, raw	TR: 2 s, TE: 29 ms, matrix size: 64x64, 32 slices, voxel size: 3x3x4 mm3	Patient (Schizophrenia), Control	Requested
CORR [35]	5093 Resting Functional Scans	Raw, accessed with PCP [36]	Differs among uploads	Healthy control	Requested
OASIS [37]	No fMRI available, mostly structural MRI	-	-	Demented, Nondemented	-
Human Connectome Project -1200 Younge Adults [38]	1200+	Both raw and processed [39]	Normalized as mentioned in the protocol [21]	Healthy control	Access granted

Appendix E: Adjacency matrix, brain connectivity visualization and conversion results

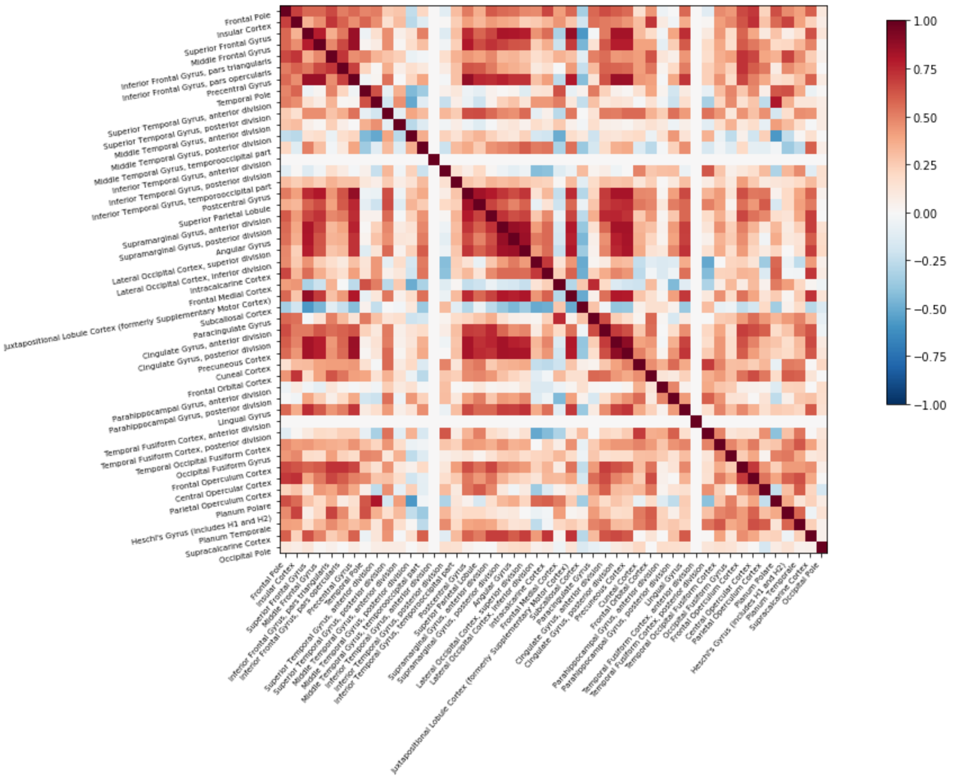


Figure E-1. Visualization of the converted matrix of an ADHD sample

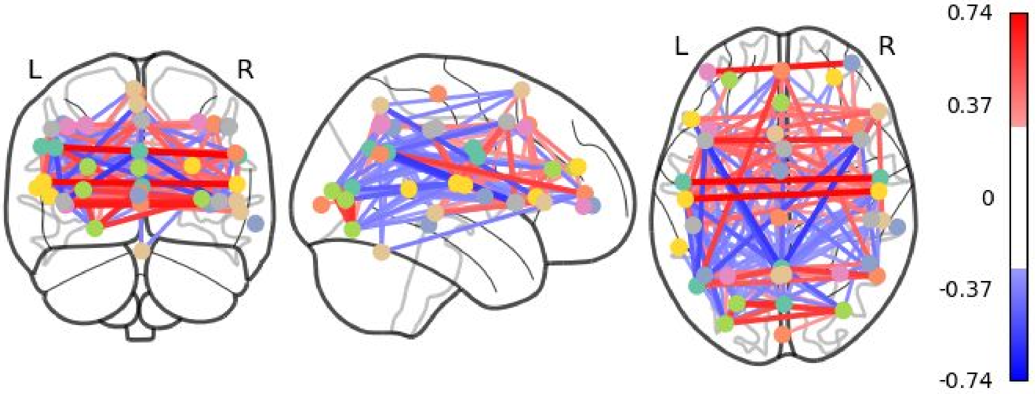
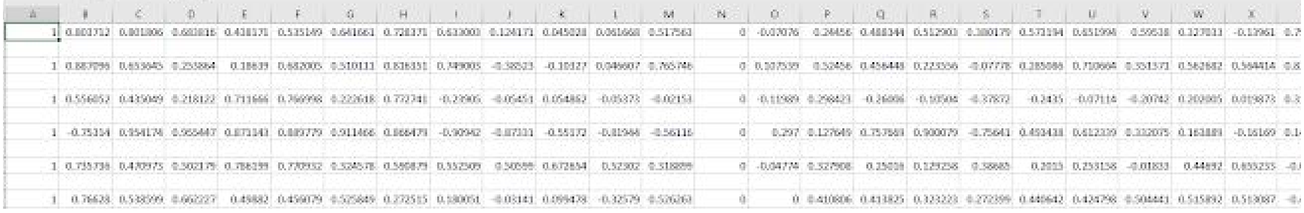
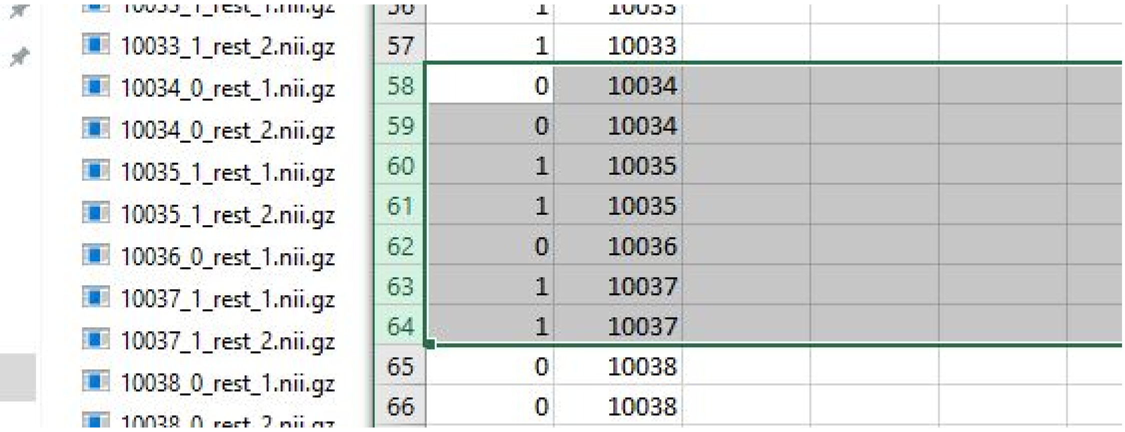


Figure E-2. Visualization of connectome from the first sample of ADHD preprocessed dataset



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
1	0.803712	0.801806	0.683816	0.438171	0.535149	0.641661	0.728371	0.633003	0.124171	0.045028	0.061668	0.517563	0	-0.07076	0.24456	0.488344	0.512903	0.380179	0.573194	0.651996	0.59538	0.127013	-0.13961	0.7
1	0.887096	0.653645	0.253864	0.18639	0.682005	0.510111	0.816351	0.749005	-0.38523	-0.10327	0.046607	0.765746	0	0.107539	0.52456	0.456448	0.223536	-0.07778	0.285086	0.710664	0.351371	0.562682	0.564414	0.8
1	0.556052	0.435049	0.218122	0.711666	0.766998	0.222618	0.772741	-0.22905	-0.05451	0.054862	-0.05373	-0.02153	0	-0.11989	0.298423	-0.26886	-0.10504	-0.37872	-0.2435	-0.07114	-0.20742	0.202805	0.019873	0.3
1	-0.75314	0.954174	0.955447	0.871143	0.889779	0.911466	0.866479	-0.90942	-0.87331	-0.55172	-0.81944	-0.56116	0	0.297	0.127649	0.757669	0.900079	-0.75641	0.453438	0.612339	0.332075	0.163883	-0.16169	0.1
1	0.735736	0.470975	0.302179	0.786199	0.770932	0.324578	0.590879	0.552509	0.50599	0.672654	0.52302	0.318899	0	-0.04774	0.327908	0.25026	0.129258	0.38665	0.2015	0.253158	-0.01833	0.44892	0.655235	-0.
1	0.76628	0.538599	0.662227	0.49882	0.456079	0.525849	0.272515	0.180051	-0.03141	0.099478	-0.32579	0.526263	0	0	0.410806	0.413825	0.123223	0.272399	0.448642	0.424798	0.504441	0.515892	0.513087	-0.

Figure E-3. partial view of a Matrix.csv file that stores each the 48x48 matrix in each row, note that because each line consists of 48^2 comma separated double, we are not able to display the whole row in this document.



10033_1_rest_1.nii.gz
10033_1_rest_2.nii.gz
10034_0_rest_1.nii.gz
10034_0_rest_2.nii.gz
10035_1_rest_1.nii.gz
10035_1_rest_2.nii.gz
10036_0_rest_1.nii.gz
10037_1_rest_1.nii.gz
10037_1_rest_2.nii.gz
10038_0_rest_1.nii.gz
10038_0_rest_2.nii.gz

56	1	10033
57	1	10033
58	0	10034
59	0	10034
60	1	10035
61	1	10035
62	0	10036
63	1	10037
64	1	10037
65	0	10038
66	0	10038

Figure E-4. extracted .nii.gz file and the corresponding label in label.csv file.

Appendix F: Actual learning curves of some algorithms

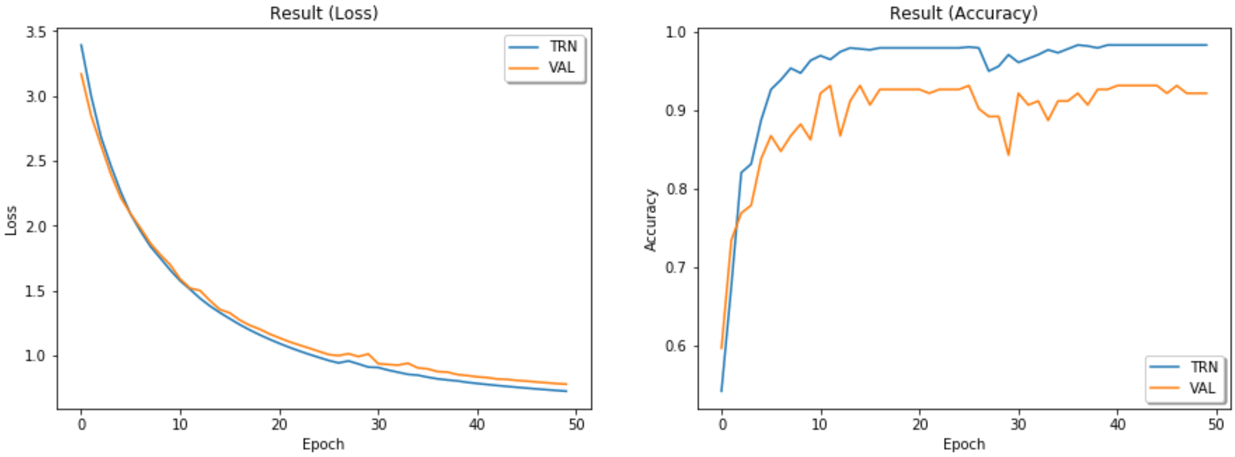


Figure F-1. Learning curves of MLP

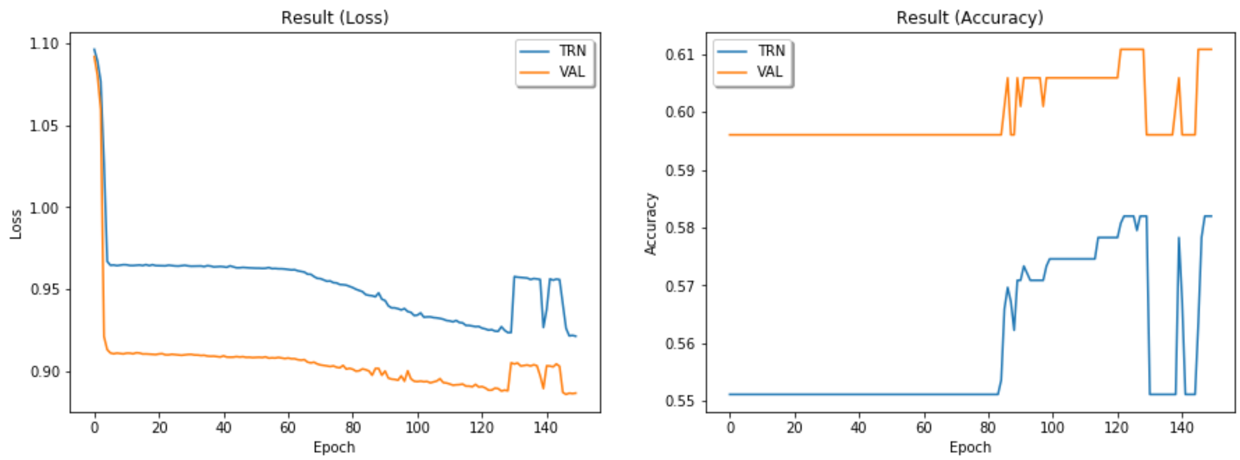


Figure F-2. Learning curves of CNN

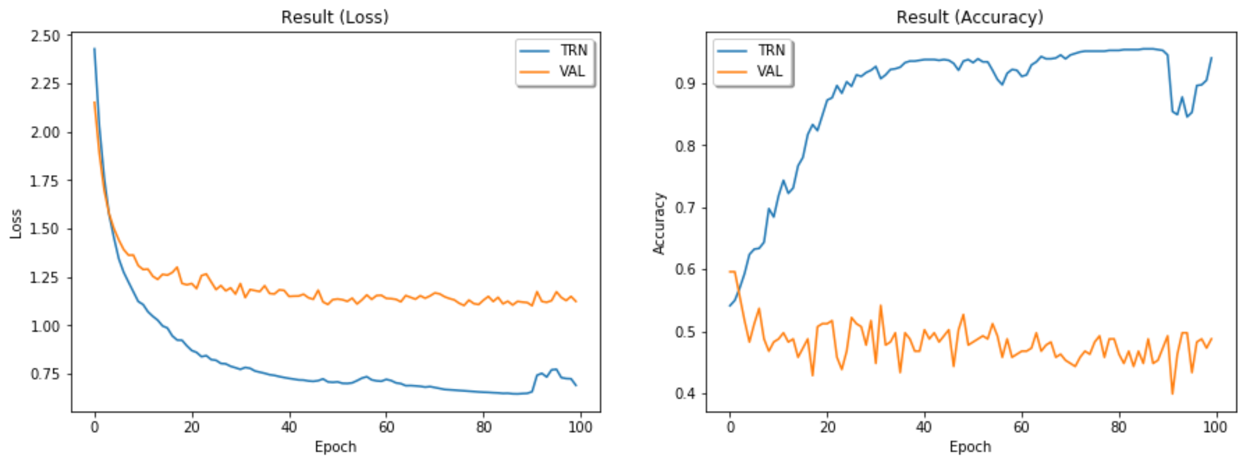


Figure F-3. Learning curves of DeepWalk + MLP

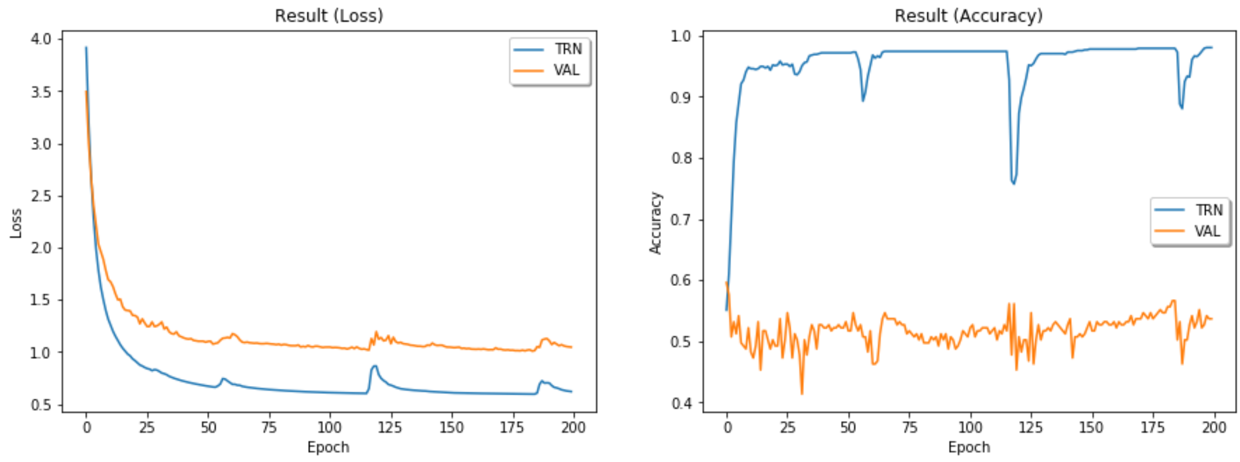


Figure F-4. Learning curves of SDNE + MLP