Prediction of Rate of Disease Progression in Parkinson's Disease Patients based on RNA-Sequence using Deep Learning

by

Siraj Ahmed

A thesis submitted to the University of Ottawa

in partial fulfillment of the requirements for the
Master of Applied Science degree in

Electrical and Computer Engineering

School of Electrical and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

The advent of recent high throughput sequencing technologies resulted in an unexplored big data of genomics and transcriptomics that might help to answer various research questions in Parkinson's disease(PD) progression. While the literature has revealed various predictive models that use longitudinal clinical data for disease progression, there is no predictive model based on RNA-Sequence data of PD patients. This study investigates how to predict the PD Progression for a patient's next medical visit by capturing longitudinal temporal patterns in the RNA-Seq data. Data provided by Parkinson Progression Marker Initiative (PPMI) includes 423 PD patients with a variable number of visits for a period of 4 years. We propose a predictive model based on a Recurrent Neural Network (RNN) with dense connections. The results show that the proposed architecture is able to predict PD progression from high dimensional RNA-seq data with a Root Mean Square Error (RMSE) of 6.0 and rank-order correlation of ($r=0.83$, $p<0.0001$) between the predicted and actual disease status of PD. We show empirical evidence that the addition of dense connections and batch normalization into RNN layers boosts its training and generalization capability.

Acknowledgments

I express my deepest gratitude to my supervisors Dr. Jeongwon Park and Dr. Majid Komeili for guiding me throughout my research work and providing invaluable feedback at every step. I attribute the level of my master's degree to their encouragement and selfless effort. I sincerely thank Dr. Jeongwon Park and CREATE-BEST program for providing me with financial support as Research Assistant to pursue my research interests. I acknowledge Compute Canada for providing high-performance computing resources without which it would have been impossible to carry out my research. My sincere thanks also go to my thesis committee members Dr. Mathew Holden and Dr. Hussein Al Osman for reviewing and giving suggestions on my thesis.

I would like to thank the administrative members of the Electrical Engineering and Computer Sciences Department especially Hélène St-Louis and Suzanne St-Michel for assisting me throughout my graduate studies for various requests. I would also like to thank Dr. Mohammed Hossam Ahmed who has been a great support and motivation while I worked as a Teaching Assistant.

A special thanks to graduate students Abdulla Kalandar and Saif for all those fruitful discussions that helped me academically and boosted my morale. A big shout-out to my mother, siblings, and friends who believed in me and helped me during tough situations. Lastly, I acknowledge the online blogs and forums specifically of Dr. Jason Brownlee, Dr. Najeeb.

Table of Contents

List of Tables viii

List of Figures ix

Chapter 1 Introduction 1

1.1 Motivation 1

1.1.1 Significance of Predicting Parkinson Disease Progression: 1

1.1.2 Studies Relevant to PD Disease Progression: 2

1.1.3 Studies Relevant to Transcriptomic Data: 3

1.1.4 Predictive Modelling with Recurrent Neural Networks(RNN): 4

1.2 Objectives 5

1.3 Dataset Overview 5

1.4 Challenges 5

1.5 Thesis Contributions 6

1.6 Thesis Outline 6

Chapter 2 Background and Related Work 8

2.1 Overview of Molecular Biology 8

2.1.1 Central Dogma of Molecular Biology: 8

2.1.2 Different types of Omics data: 11

2.1.3 Rationale to Choose Transcriptomic Data for our Study: 12

2.1.4 Transcriptomic Studies using RNA-Seq protocol 13

2.2 Deep Learning 15

2.2.1 RNN Layer: 15

2.2.2 Stacked RNN layers without skip/residual connections: 16

2.2.3 Stacked RNN layers with dense connections: 17

2.2.4 Batch Normalization: 18

Chapter 3 Dataset Description 21
3.1 Overview of Databases for Parkinson's Disease 21
3.1.1 Resources for Parkinson's Disease Study 21
3.1.2 Rationale to choose PPMI database for our study: 22
3.2 Description of Datasets in PPMI database 23
3.2.1 Description of the subjects under study in PPMI: 23
3.2.2 Description of the Input Features and Target Variable for our Study: 23
3.2.3 Number of Samples in our study: 27
3.3 Data Selection Criteria 28
Chapter 4 Data Pre-Processing 30
4.1 Preparation of Predictor Variables 30
4.1.1 Preparation of subset of RNA-Seq dataset: 30
4.1.2 Derived Variables: 32
4.2 Preparation of Target Variable 33
4.2.1 MDS-UPDRS: 33
4.2.2 Disease Progression Curves: 34
4.2.3 Time Window Shift: 36
4.3 Datasets Merging 36
4.4 Missing Value Imputation 37
4.4.1 Imputation for Continuous Data type variable: 38
4.4.2 Imputation for Categorical Data type variable: 39
4.5 Feature Selection 39
4.5.1 Minimum Redundancy Maximum Relevancy (MRMR) Method: 40
4.5.2 Notations: 40
4.5.3 Data Pre-Process for Feature Selection: 41

4.5.4 Implementation of Feature Selection Scheme: 42
4.6 Other Important Pre-Processing Steps 42
4.6.1 Handle Variable Number of Visits: 42
4.6.2 Train, Validation, and Test Split: 43
4.6.3 Normalization: 43
Chapter 5 Proposed Methods 44
5.1 Modelling of Problem Statement 44
5.1.1 Problem Statement: 44
5.1.2 Mathematical Notation for the Problem Statement: 45
5.2 Proposed RNN Model: Densely Connected Recurrent Neural Networks for PD Progression 46
5.2.1 Proposed Densely Connected RNN Model: 48
5.3 Training Pipeline and Hyperparameter Tuning 51
5.3.1 Algorithm : Model Training and Hyper Parameter Tuning 54
5.4 Hardware and Software Environment 55
5.4.1 Cloud Computing Resource for Model Training and Tuning: 55
5.4.2 Software Environment: 55
Chapter 6 Baseline Models and Evaluation Metrics 56
6.1 Baseline Methods 56
6.1.1 Machine Learning Baseline Methods: 56
6.1.2 Training Strategy for Machine Learning based Baseline Methods: 56
6.1.3 Experimental Setup for Machine Learning based Baseline Methods: 58
6.1.4 Other RNN based Baseline Models: 59
6.1.5 Experimental Setup for Other RNN based Baseline Models: 61
6.2 Evaluation Metrics 62

6.2.1 Prerequisites: 62
6.2.2 Mathematical Definition of our Evaluation Metrics: 64
6.2.3 Statistical Significance Test for Model Comparison: 68
Chapter 7 Experimental Results and Discussion 70
7.1 Results Overview 70
7.2 Prediction of Disease Progression in PD Patients using our Predictive Model: 72
7.2.1 Discussion: 75
7.3 Performance Comparison Between our Proposed Model and Baseline Methods 76
7.4 Results from Other RNN based Baseline Models 79
7.5 Performance Comparison Between Densely Connected RNN Models and Plain RNN Configurations 80
7.5.1 Discussion: 82
7.6 Performance Comparison Between Densely Connected RNN Models with Batch Normalization and Densely Connected RNN Models without Batch Normalization Configurations 83
7.6.1 Discussion: 86
Chapter 8 Conclusion and Future Work 87
8.1 Conclusion 87
8.2 Future Work 87
References 88

List of Tables

Table 2-1: Different Types of Omics Data 11
Table 2-2: Simplified Version of Output from RNA-Seq 13
Table 3-1: Comprehensive Databases available for Parkinson's Disease 21
Table 3-2: Description of Dataset of our Study 23
Table 3-3: Frequency of Information Collection in PPMI 28
Table 4-1: Analysis of Missing Values for various subsets of data of our finalised dataset 38
Table 4-2: MRMR Criterion Functions for Different Data Types 41
Table 4-3: Scheme to discretize the RNA-Sequence dataset for the purpose of feature selection 41
Table 4-4: Computing Configuration for Feature Selection 42
Table 5-1: Hyper Parameter Tuning Search Space 52
Table 5-2: Default Initialization values of Hyperparameters 53
Table 5-3: Computing Configuration for Model Training and Hyper Parameter Tuning 55
Table 6-1: Aggregated Value for Different Data Types 57
Table 6-2: Other Baseline RNN Model Configurations that were trained 60
Table 6-3: Computing Configuration for Other RNN based Baseline Methods 62
Table 6-4: Evaluation Metrics 67
Table 7-1: "Dense-Vanilla" Net Architecture and Optimized Hyper Parameters 70
Table 7-2: Results of the model with and without non-transcriptomic input features 72
Table 7-3: Performance Comparison Between our Proposed Model and Baseline Methods 76
Table 7-4: Statistical Significance Test for Model Comparison 78
Table 7-5: Results from Other RNN based Baseline Models 79
Table 7-6: Performance Comparison Between Densely Connected RNN Models and Plain RNN Configuration 81
Table 7-7: Performance Comparison Between Densely Connected RNN Models with Batch Normalization and Densely Connected RNN Models without Batch Normalization: 84

List of Figures

Figure 2-1: Central Dogma of Molecular Biology 9
Figure 2-3: A Simple RNN Layer with hidden RNN cells.. 16
Figure 2-4: Plain Configuration of Two Stacked RNN layers without skip connections..... 17
Figure 2-5: Densely Connected RNN Architecture.. 18
Figure 4-1: Preparation of "Subset of RNA-Seq" dataset from PPMI RNA-Resource..... 32
Figure 4-2: Variation of MDS-UPDRS final score for a sample of Parkinson Disease subjects..34
Figure 4-3: Data smoothing on the disease progression curve.. 35
Figure 4-4: Time Window shifts performed on annual visits of MDS-UPDRS final score. 36
Figure 4-5: Overview of Datasets Merging 37
Figure 4-6: Missing value imputation technique for continuous data type variable..... 39
Figure 5-1: Block diagram of the problem statement. 46
Figure 5-2: Composite Block(CB).. 49
Figure 5-3: Dense Block(DB). 50
Figure 5-4 : Proposed Densely Connected RNN Architecture. 50
Figure 6-1: Data Aggregation for a single patient with N visits. 58
Figure 6-2: Two RNN layers stacked on each other 60
Figure 7-1: Predicted and Ground Truth disease progression curves for Test Patients.. 73
Figure 7-2: Predicted and Ground Truth disease progression curves for Remaining Test Patients 74
Figure 7-3: Comparison of RMSE produced by our proposed model and the baseline methods in predicting MDS-UPDRS score.. 77
Figure 7-4: Comparison in rank correlation produced by our proposed model and the baseline methods in predicting MDS-UPDRS score.. 78
Figure 7-5: Comparison of RMSE produced by Densely Connected RNN Models and Plain RNN Configurations in predicting MDS-UPDRS score.. 82
Figure 7-6: Comparison of Correlation scores obtained by Densely Connected RNN Models and Plain RNN Configurations in predicting MDS-UPDRS score.. 82

Figure 7-7: Comparison of RMSE produced by Densely Connected RNN Models with and without Batch Normalization layers..... 85
Figure 7-8: Comparison of Correlation scores obtained by Densely Connected RNN Models with and without Batch Normalization layers..... 85

Chapter 1 Introduction

1.1 Motivation

1.1.1 Significance of Predicting Parkinson Disease Progression:

Parkinson's Disease (PD) is a degenerative disorder that progresses over time affecting the central nervous system. PD affects 7-10 million people worldwide and is clinically characterized by a decline in both motor and non-motor abilities. PD patients suffer from motor illness such as shakiness in body, slowness in movement, forward stooped posture, shuffling gait, difficulty in speech, muscle stiffness, and various non-motor symptoms such as declined cognitive skills, fatigue, anxiety, depression, visual hallucinations, sleep disturbances and slowness of thinking. PD is caused by the death of neurons in the substantia nigra region of the brain which produces dopamine [1].

The disease is highly idiopathic and currently, there is no cure. However, the symptoms can be managed to improve the quality of life. The treatments to manage the symptoms include medications (artificial dopamine and levodopa therapies), deep brain stimulation, rehabilitation, etc. are highly personalized. Although the symptoms are the same across all PD patients, the disease progresses differently across different patients. For instance, few patients follow the fastest trajectory of disease progression and their condition worsens quickly. On the other hand, there are few patients who follow a slow trajectory. Such heterogeneity in PD patients hinders the practitioners from prescribing appropriate treatment.

In addition, such heterogeneity makes a clinical trial of disease-modifying therapies challenging as we now need a larger number of subjects to be enrolled that makes the process expensive and time-consuming [2]. Therefore, there is an unmet need for a prognostic tool to help the practitioners know beforehand if the newly diagnosed PD patient will progress quickly in the disease or has a slow progression rate. This will help support their decision to refer the patient to a larger, more experienced center for care very early on. This project focuses on building such a predictive model for the prognosis of PD by employing deep learning. Before we formulate our

problem statement, let us see how previous researchers have been looking for approaches to help diagnosis and prognosis of PD.

1.1.2 Studies Relevant to PD Disease Progression:

Machine learning algorithms have been applied to classify Parkinson's Disease (PD) from healthy subjects and to predict the disease progression of PD patients using a variety of clinical features data. The disease status of Parkinson's disease is well represented by Movement Disorder Society-sponsored unified Parkinson's disease rating scale (MDS-UPDRS) which reflects the motor symptoms and clinometric properties of PD on a scale of 0 to 272 with 0 being normal and 272 being severe motor and non-motor decline [3]. Kaur et al. [4] used the voice recordings of Parkinson's disease patients to predict the MDS-UPDRS score using an ensemble of machine learning models for the diagnosis of PD. Maachi et al. [5] investigated with longitudinal gait disturbances data to build a Deep Neural Network (DNN) to predict the severity of PD based on the MDS-UPDRS score. Moises et al. [6] used dynamically enhanced handwriting images to build a classifier to support the clinicians in the diagnosis of PD. Prashanth et al. [7] used clinical data such as Rapid Eye Movement (REM) sleep Behaviour Disorder and olfactory loss with a combination of Cerebrospinal fluid (CSF) measurements to classify early PD subjects from healthy subjects using Support Vector Machines (SVM), boosted trees, and random forest classifiers. Ilianna Kollia et al. [8] designed DNN whose inputs are Magnetic Resonance Imaging and DaTscan data and the output is the classification of subjects into PD and healthy.

The above-mentioned studies have investigated how various data categories such as speech signals, brain images, DaTscans, gait sensor data that capture clinical features such as gait disturbances, speech disorders, handwriting, sleep behavior disorders, motor decline, etc. have been used to classify Parkinson's patients from healthy individuals and to predict disease progression in Parkinson's disease. In addition to the clinical features data, we have diverse "Omics" data that has the potential to provide a global view of the complex biological processes related to human diseases[9]. Recent advances in high-throughput technologies allow efficient investigation of various omics data, such as genomics, epigenomics, transcriptomics, proteomics, metabolomics, microbiomics [10]. Studying a single type of omics data is generally insufficient to explore underlying mechanisms of complex human diseases [9], since each of these biological data focuses on different mechanisms of cell viz DNA replication, RNA transcription, protein synthesis,

etc. For example, genomics study such as genotyping is the study of identifying genetic variants associated with the disease whereas epigenomics study like DNA methylation profiling focuses on reversible modifications of DNA. The transcriptomic study, such as RNA-Sequence which is relatively newer omics data, examines both qualitative and quantitative levels of RNA levels [10].

1.1.3 Studies Relevant to Transcriptomic Data:

Eliza Courtney et al.[11] investigated the usefulness of RNA Sequence data in the research of neurodegenerative diseases such as Alzheimer's disease, Parkinson's disease, and Huntington's disease. We know that neurodegenerative diseases are characterized by a significant change in gene expression levels (RNA transcripts) and splicing patterns that occur before the onset and also during the progression of the disease. Hence, the authors called for a need to investigate RNA-Seq data in these diseases as RNA-Seq has the advantage of digital expression profiling and is able to identify alternative splicing patterns that have become a major focus of degenerative disease research [11].

To the best of our knowledge, no study investigates the utility of transcriptomic data such as RNA-Sequence for either classification of Parkinson's Disease or to predict disease severity/progression in Parkinson's Disease. However, few earlier studies have investigated the utility of Genomics data to study how Single Nucleotide Polymorphism (SNPs), genetic variants are able to explain discrimination of Parkinson's disease from healthy control. Nadella Kumudini et al. [12] used SNPs related to Parkinson's disease to predict the risk factor of PD using Artificial Neural Network (ANN) in a classification framework. Babu et al.[13] used Micro-Array gene expression data to build a meta-cognitive radial basis function network classifier to classify PD patients from healthy control.

Mansu Kim et al. [14] used the technique of Imaging genetics to study the relationship between MDS-UPDRS score and a combination of DNA genotyping and neuroimaging data. Both DNA genotyping and neuroimaging features of a subject at a visit were given as input to a linear regression model to predict the MDS-UPDRS score for that visit. The authors reported that combined features of DNA genotyping and neuroimaging were able to explain 72.5% of the variance in MDS-UPDRS and the same combination of features were able to predict the MDS-

UPDRS with an RMSE of 7.82 and showed a correlation of ($r=0.788$, $p<0.001$) with actual MDS-UPDRS.

The authors of [14] have made promising studies in establishing a relationship between PD disease status and a combination of neuroimaging and genomics. It will be interesting to investigate how transcriptomic data like RNA-Seq will explain the disease status as compared to genotyping data. Currently, the RNA-Seq test costs more than neuroimaging, however, it will be worthy to study the usefulness of transcriptomic data for the purpose of disease progression.

Moreover, the authors use the classical machine learning approach to utilize the genotyping information of a known single time point to predict the disease status for that time point. There is an unmet need to use this omics information of past time points to predict the disease status of future time steps so that we stay ahead of the disease and plan the treatment accordingly. This might also help to identify the subjects with fast, moderate, and slow disease progressions as recent studies suggest Parkinson patients follow different rates of disease progression [15][2]. We believe there is no such predictive model whose inputs are genomic/transcriptomic information of baseline first visit and the output is disease status (MDS-UPDRS) for future time steps.

1.1.4 Predictive Modelling with Recurrent Neural Networks(RNN):

For disease prediction problems, the traditional machine learning algorithms predict the disease status by aggregating the longitudinal features rather than leveraging through the temporal patterns [16]. To address this problem of capturing patterns from longitudinal data, Wang et al. [16] aimed to employ Recurrent Neural Networks (RNN) to predict Alzheimer's Disease (AD) status for a patient based on the historical clinical information of patients such as demographic data, health history, physical examination, etc. in a multi-class classification framework. The authors managed to leverage the longitudinal temporal "clinical" information of patient's historical visits to predict disease progression in a classification framework. However, for Parkinson's disease progression, we believe there is no study that leverages the longitudinal temporal "genomic" or "transcriptomic" information of historical visits to predict future disease status in a regression framework.

1.2 Objectives

To address the above-mentioned overdue problems, we formulated our problem statement as follows:

Problem Statement: Single Step ahead predictive model for disease progression

To investigate whether artificial neural networks especially Recurrent Neural Networks (RNN) can predict the rate of disease progression in Parkinson's disease by fully leveraging through longitudinal temporal information on RNA sequencing data. We aim to predict the MDS-UDRS score (disease status) of a patient for the immediate future hospital visit by considering the RNA sequence data of multiple previous visits in a regression framework.

In other words, the problem is modeled in a way that when Baseline year (1st year) RNA-seq data is given to a predictive model it predicts the next year's MDS-UPDRS, and when baseline year and 2nd year's RNA-seq data is given to the model, it predicts the 3rd year's MDS-UPDRS score and finally, when Baseline year's, 2nd year's and 3rd year's data is given, the model predicts the 4th year's MDS-UPDRS score. Thus, as we move forward in time sequences, the model leverages the temporal patterns in the historical RNA sequence data and tries to predict the immediate future time disease status.

1.3 Dataset Overview

The datasets used for this study were provided by Parkinson's Progression Markers Initiative (PPMI) which has longitudinal 4 years of data for 400 PD subjects and 200 Healthy control subjects. The data includes clinical, imaging, and biological data available publicly to be downloaded [17].

1.4 Challenges

i) Omics data such as RNA-sequence is characterized as high dimensional data of the order of 35,000 RNA transcripts as features.

ii) Health record data is characterized by the presence of a large number of missing values. Furthermore, the number of visits for a patient is not uniform across all the patients.

The patients in our dataset have a variable length of visits with an average of 2.5 visits for a subject.

iii) For a disease progression problem in neurodegenerative diseases, it is extremely difficult to capture temporal patterns from the longitudinal electronic health record (EHR) since each patient's EHR is unique and highly heterogeneous[18][19][20].

1.5 Thesis Contributions

i) RNA sequence is relatively new, and this study is the first to report benchmark results on the use of transcriptomic data such as RNA sequence for PD disease progression using deep learning.

ii) Traditional time series methods have difficulty in handling high dimensional longitudinal data and do not consider the temporal patterns of longitudinal data. We show empirical evidence that densely connected deep Recurrent Neural Networks have a strong potential to be used to capture the inherent patterns from transcriptomic data of PD patient's historical visits.

iii) Predictive Model for Disease Progression:
We build a deep learning-based predictive model for Parkinson's Disease progression that can predict the MDS-UPDRS score for a PD patient by capturing the patterns from longitudinal data of transcriptome of the patient. We proposed a deep Recurrent Neural Network architecture that was able to reach an RMSE of 6.0 in the prediction of MDS-UPDRS and high-rank order correlation ($r=0.83$, p-value <0.0001) of predicted MDS-UPDRS score with the actual MDS-UPDRS score.

1.6 Thesis Outline

This thesis is organized as follows:

- Chapter 2 provides the background on the topics of molecular biology and artificial neural networks that are relevant to our study.

- Chapter 3 presents literature on the database resources available for Parkinson's Disease followed by a detailed description of the PPMI database. Further, we explain the input features and target variables for our study.
- Chapter 4 explains the various pre-processing steps performed on the datasets such as missing value imputation, feature selection algorithm, normalization of data etc.
- Chapter 5 presents the modeling of the prediction problem. Further, it discusses in detail the mechanics of our proposed Densely connected RNN architecture. Also, it presents the training and hyperparameter tuning pipeline for our proposed work and the experimental setup.
- Chapter 6 discusses the baseline models (classical machine learning techniques viz LR, SVM, Decision Trees, etc.) that are used to compare the performance of our proposed model. Further, it defines the evaluation metrics for model comparison.
- Chapter 7 present the summary of the various experiments that were conducted as part of our study and the relevant results and discussion.
- Chapter 8 summarises the work done in this thesis and presents the direction of future research in the predictive modeling of Parkinson's Disease Progression.

Chapter 2 Background and Related Work

2.1 Overview of Molecular Biology

2.1.1 Central Dogma of Molecular Biology:

In this section, we define the basic definitions of the central dogma of molecular biology that helps us to understand the background of our research study.

A cell is the basic structural and functional unit of life. A group of similar cells forms tissue and a functional group of multiple tissues forms an organ. A group of multiple organs integrates to form an organ system. The group of multiple organ systems then work together to form a complete and functional living organism like us. In human beings, there are 11 organ systems which include circulatory, respiratory, digestive systems and so on that perform different functions necessary for life by producing different proteins like enzymes and hormones. All these systems are built upon cells that differ in structure and function, however, every cell in the human body has a nucleus that house a special macromolecule called DNA organized into 23 pairs of chromosomes. Each of these chromosomes are organized into a short segment of DNA called genes that store the genetic information necessary for defining who we are [21][22].

Every cell in the human body whether belonging to the brain (neuronal cell) or kidneys (nephron cell) has the same set of 23 chromosomes with the same DNA. If every cell has the same DNA then how are the two cells so structurally and functionally different? And how are two organs so different from each other in terms of enzymes they produce? This is because a different type of cells has a different subset of genes being active and thus produce a different type of proteins[21]. This differentiation of cells is controlled by intermediate molecules called RNA.

DNA and RNA are the two types of nucleic acids that control protein synthesis and are responsible for the transformation of genetic information from one generation to another generation. The stable form of DNA is a double helical structure that is organized into short segments called genes. As per the Human genome project, there are around 25,000 genes. Each DNA segment is composed of four nucleotide base sequences- Adenine (A), Guanine (G), Cytosine (C) and Thymine(T). RNA is a single-stranded molecule made up of nucleotide base sequences- Adenine (A), Guanine (G),

Cytosine (C) and Uracil(U). The interrelationship between DNA, RNA, and proteins constitutes the "central dogma of molecular biology" [23]. The central dogma states that DNA makes RNA, and RNA makes Protein. The processes involved as part of the central dogma of molecular biology are replication, transcription, and translation and defined as follows:

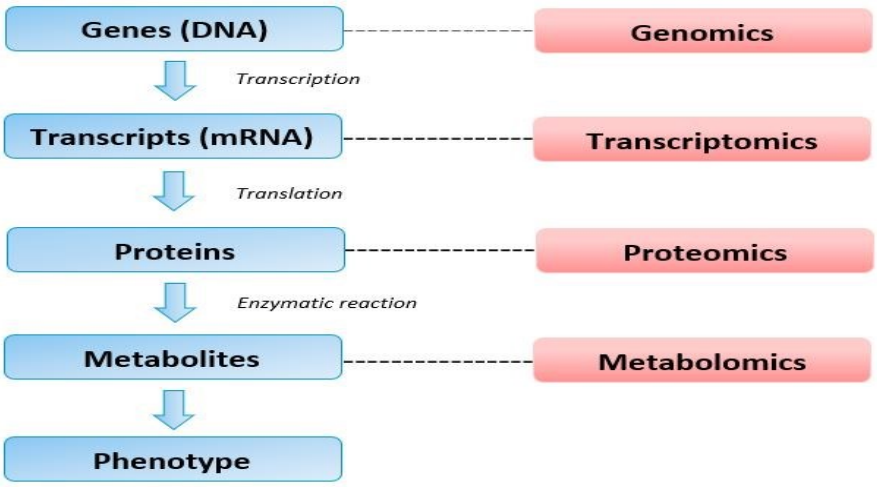


Figure 2-1: Central Dogma of Molecular Biology

DNA Replication:

DNA Replication is the process in which DNA replicates itself to produce two copies of daughter DNAs. This process happens each time a cell divides into two daughter cells and mostly a faithful replication is performed. However, there could be small errors during copying of DNA such as Single Nucleotide Polymorphisms or SNPs commonly called 'Snips'. They can have positive or negative or no effect on how the body functions. Some snips can change the way you look, the way you talk, or might determine your health. Also, mutations in genes can lead to genetic disease or cancer or they may make an organism evolve by giving an advantage over its own species. Defect in the naturally inbuilt repair system of DNA can cause diseases like xeroderma pigmentosa (XP), hereditary nonpolyposis colon cancer (HNPCC). The total content of DNA in a cell or organism is called genome and the study of the genome is known as genomics.

RNA Transcription:

RNA Transcription is a process in which genetic information of DNA is copied to RNA. The number of copies of RNA is called transcripts. Only a subset of genes within the whole genome is transcribed in a type of cell and thus the behavior of that cell depends on the subset of genes being selected for transcription. During the process of transcription, the genes subsets are copied to produce four types of RNA molecules viz. messenger-RNA (mRNA), transfer-RNA (tRNA), ribosomal-RNA (rRNA), and small nuclear-RNA (sRNAs). The messenger-RNA contains the genetic code to be transferred to synthesize proteins and is also called coding-RNA while the remainder of RNA molecules are called non-coding RNAs. The non-coding RNAs help in the process of synthesis of proteins from the coding RNA. Out of total RNA content, mRNA is the largest molecule but constitutes only 4% of RNA content and the rest 96% RNAs are non-coding RNAs with smaller sizes. Total active RNA produced at a time point by a cell in which codes for proteins are termed as transcriptomes and the study of transcriptomes is known as transcriptomics**.**

Reverse transcription is a process in which DNA is synthesized from RNA. For example, retroviruses such as HIV-virus can form double-stranded complementary DNA (cDNA) from its RNA.

Protein Translation:

The translation is a process in which the synthesis of proteins takes place from mRNAs with the assistance of non-coding RNAs such as tRNA. The mRNA is organized into a three-base stretch called ‘codons’ and each codon has the information to synthesize a particular kind of amino acid. A long chain of multiple amino acids forms a protein. After the translation process, proteins undergo post-translational changes to become active proteins. Defect in post-translational modifications can cause degradation and accumulation of proteins and their products which leads to certain neurological diseases like Alzheimer’s disease, Huntington’s disease, Creutzfeldt-Jakob’s disease, and mad cow disease. Mutated proteins can lead to diseases like cystic fibrosis. The set of proteins expressed by a cell or organism is termed as proteome and the study of the proteome is called proteomics.

The proteins perform a wide range of functions in an organism, including providing structures to cells, transporting molecules from one part of the organism to other, catalyzing metabolism, etc. The study of the products and intermediates of the process of metabolism is called metabolomics. A healthy individual is the one whose metabolic pathways are all functioning normally. Any disturbance in the metabolism of a cell or group of cells results in abnormal functioning of the organism or commonly called symptoms or illness.

2.1.2 Different types of Omics data:

Recent advances in high-throughput technologies allow efficient investigation of various omics data, such as genomics, epigenomics, transcriptomics, proteomics, metabolomics, and microbiomics. Each of these biological data focuses on different mechanisms of cell viz DNA replication, RNA transcription, protein synthesis, etc. For example, genomics study such as genotyping is the study of identifying genetic variants associated with the disease whereas epigenomics study like DNA methylation profiling focuses on reversible modifications of DNA. A transcriptomic study done via RNA-Sequence is relatively newer omics data that examine both qualitative and quantitative levels of RNA levels [10]. The following is the summary of different types of Omics studies and their application to provide valuable insights about various Mendelian and complex diseases:

Table 2-1: Different Types of Omics Data

Omics Data	**Molecule under study**	**Approach**	**Application to help diseases**
Genomics	DNA	Genome wide association studies (GWAS)	To identify genetic variants contributing to complex diseases
Epigenomics	Reversible modifications of DNA- For	epigenome-wide	differentially methylated regions of DNA can be used as indicators of

	example: DNA methylation	association studies	disease status for metabolic syndrome cardiovascular disease, cancer and many other pathophysiologic states
Transcriptomics	RNA	RNA-Seq	To study the role of RNAs in many physiological processes
Proteomics	Proteins	MS-based	to investigate global proteome interactions and quantification post-translational modifications
Metabolomics	Products and intermediates involved in metabolism	MS-based	To quantify multiple small molecule types, such as amino acids, fatty acids, carbohydrates, or other products of cellular metabolic functions.
Microbiomics	Microbes such as bacteria, fungi, archaea living in an organism	NGS application for 16S ribosomal abundance and metagenomics quantification	substantial variations in microbiota composition between individuals resulting from seed during birth and development, diet and other environmental factors, drugs, and age

2.1.3 Rationale to Choose Transcriptomic Data for our Study:

Eliza Courtney et al.[11] investigated the usefulness of transcriptome in the research of neurodegenerative diseases such as Alzheimer's disease, Parkinson's disease, and Huntington's disease. Neurodegenerative diseases are characterized by a significant change in expression levels of genes i.e. the number of RNA transcripts being produced and the splicing patterns. There is a significant change in the transcriptome before the onset and during the progression of the disease. Hence, the authors called for a need to investigate the transcriptome using the recent high throughput sequencing method called RNA-Sequence or RNA-Seq in these diseases. RNA-Seq has the advantage of digital expression profiling and is also able to identify alternative splicing patterns that have become a major focus of degenerative disease research [11]. The next section

explains the typical pipeline involved in studying the transcriptome of a cell using the RNA-Seq protocol.

2.1.4 Transcriptomic Studies using RNA-Seq protocol

Transcriptomic studies are performed to find out how are the abnormal cells different from normal cells. We want to look at the differences in the gene expressions of the two cells i.e. what genes are active and how much each gene is transcribed. High throughput sequencing such as RNA-Seq [24][25] helps us to perform the following two studies on the transcriptome:

i) Qualitative study:
To answer what genes in the abnormal cell are active that should not have been active. Also, it could be another way to find what genes are not getting transcribed in an abnormal cell that should have been otherwise, transcribed.

ii) Quantitative study:
To study how much a gene is getting transcribed in the cell i.e. to answer the question If the gene is under expressing or overexpressing in the cell.

Both the studies are performed by mapping the RNA transcripts (or reads) in the sample cells to a standard reference human genome library and posting the number of counts of transcripts mapped per Gene. To understand what a transcriptomic study is, consider a very simplified version of an output file from RNA-Seq studies, tabulated in Table 2-2:

Table 2-2: Simplified Version of Output from RNA-Seq

Reference Human Genome		**Raw RNA transcript per gene**	
Gene	**Position in Human Genome**	Sample—1 (Healthy Cell)	Sample-2 (Abnormal Cell)
Gene1	Chromosome-1, position (341752 to 361254)	1200	1200
Gene2	Chromosome-1, position (375413 to 385124)	0	1400
Gene3	Chromomsome-1, position (425000 to 433125)	2600	1800

Gene4	Chromomsome-1, position (455400 to 464125)	1800	3600

Consider Table 2-2, There are 23 chromosomes in the human cell with each chromosome arranged in short segments called genes. There are close to 25000 genes in the human genome as per the Human genome project, however, for the sake of simplicity we have shown only four genes in the above table. Each gene is indexed by its chromosome and its start and end position in the DNA molecule, for ex: gene1 is situated in the chromosome-1 with the base pair starting position-341752 and ending at 361254. We have this information for all the 25000 genes, and this is saved as part of the reference human genome library. An RNA-Sequencing method is applied to samples from a healthy cell and abnormal cell using the RNA-Seq protocol to study the gene expressions in these two samples. For example, from Table first row, gene-1 in an abnormal cell is transcribed with no difference in expression from gene-1 in a normal cell. However, gene-2, which is not active in a normal cell, is active in the abnormal cell, as we see that gene-2 is expressing about 1400 transcripts in the abnormal cell. Similarly, gene-3 is expressing 1800 transcripts in abnormal cells as compared to 2600 transcripts in normal cells and thus is "under expressed" in an abnormal cell. Furthermore, the gene-4 is "overexpressed" in the abnormal cell as compared to the normal cell. These types of studies are synonymous with "transcriptomic studies", "gene expression studies", "RNA-Seq", "expression profiling".

The raw read counts are usually normalized to account for various biases that occur during the RNA-Seq analysis such as the effect of longer gene length, the effect of longer RNA sequencing depth, low-quality reads, and the proportion of reads that mapped to a gene in each sample, etc. There are several methods to post the normalized raw reads counts such as 'RPKM', 'FPKM', and 'TPM' that help to remove the bias from RNA-Seq analysis. However, TPM or 'Transcripts per million' is the designated reporting method because of relatively easy comparison of the proportion of read counts mapped to a gene in each sample.

RNA-Seq protocol [24][25]**:**

Transcriptomic studies are performed using high throughput sequencing technologies such as RNA-Seq protocol and are composed of the following three steps:

i) Preparation of Sequencing Library:
As per the Illumina NovaSeq6000 protocol, the RNA is extracted from the cells of a sample and is broken into small fragments. Each fragment is converted into a complementary-DNA strand because cDNA is much stable than RNA. The cDNA fragments are further attached to sequencing adaptors and amplified.

ii) Sequencing the library:
The amplified reads are now ready for sequencing and placed vertically over the flow cell. With the help of fluorescent probes that are attached to the first base of fragments, nucleotides are identified using the color codes. A file called FASTQ is generated with the nucleotide sequence of each read and the quality scores. The sequenced reads are then aligned and mapped to the reference genome to generate the transcripts data.

iii) Data Analysis:
Analysis of summarized reads, Normalization of gene expression takes place during this phase, and result files are generated.

For more information and a detailed understanding of the above steps in RNA-Seq protocol, the reader is referred to the resources [24]–[26].

2.2 Deep Learning

In this section, we present the background on Recurrent Neural Networks (RNN), Densely connected RNN, and Batch Normalization.

2.2.1 RNN Layer:

A variety of RNN layers exist such as LSTM, GRU, Vanilla, and other variants. The output recurrent step $\boldsymbol{h}_{<t>}$, irrespective of the type of the cell, may be written as follows:

$$\boldsymbol{h}_{<t>} = f(\boldsymbol{x}_{<t>}, \boldsymbol{h}_{<t-1>}) \tag{2.1}$$

where $\boldsymbol{x}_{<t>}$ is the vector input at the current time step t whereas $\boldsymbol{h}_{<t-1>}$ is the vector output of the cell state of the same hidden layer from the previous time step. The function $f(.)$ depends upon the type of the cell in that layer viz. LSTM, GRU, or Vanilla. The number of hidden RNN cells in

the RNN layer is an important hyperparameter. A representation of a simple RNN layer maybe given by Figure 2-2 below:

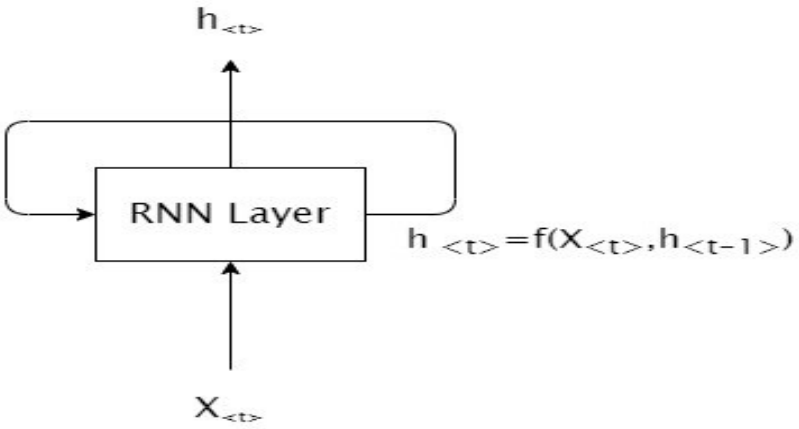


Figure 2-2: A Simple RNN Layer with hidden RNN cells. The f(.) of recurrent step depends upon the type of cells in the RNN layer. Can be a GRU, LSTM or Vanilla.

2.2.2 Stacked RNN layers without skip/residual connections:

Multiple RNN layers may be stacked upon each other such that the output state of a layer *l* is given as input to the $(l+1)^{th}$ layer. The output recurrent step ($\boldsymbol{h}^{l}_{<t>}$) for the l^{th} layer at a time step *t* can be given as follows:

$$\boldsymbol{h}^{l}_{<t>} = f(\boldsymbol{x}^{l-1}_{<t>}, \boldsymbol{h}^{l}_{<t-1>}) \tag{2.2}$$

where $$\boldsymbol{x}^{l-1}_{<t>} = \boldsymbol{h}^{l-1}_{<t>} \tag{2.3}$$

here $\boldsymbol{x}^{l-1}_{<t>}$ is the vector input to the l^{th} layer from the $(l-1)^{th}$ layer at time *t* which is equal to $\boldsymbol{h}^{l-1}_{<t>}$. The $\boldsymbol{h}^{l}_{<t-1>}$ is the output of the cell state of the same hidden layer from the previous time step. An example of a two-layered RNN network is given below:

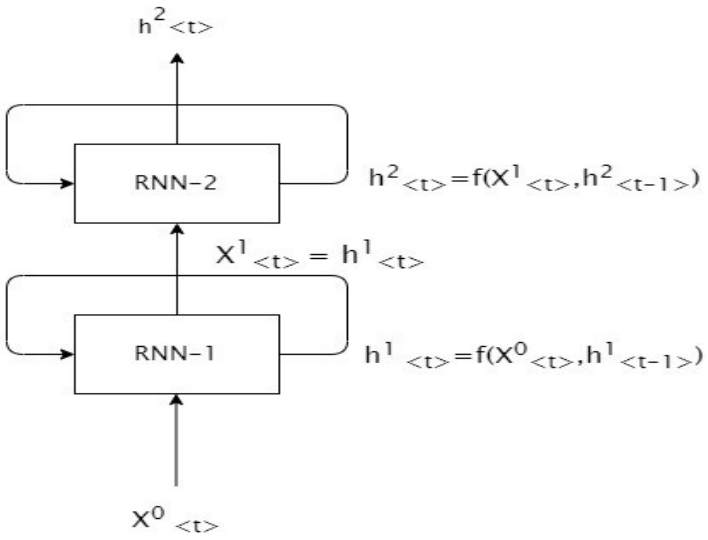


Figure 2-3: Plain Configuration of Two Stacked RNN layers without skip connections. At time step *t*, the input vector to the 2nd layer RNN is represented by $\boldsymbol{x}^{1}_{<t>}$ which is the output from the 1st layer represented as $\boldsymbol{h}^{1}_{<t>}$. The output of the 2nd layer is given by $\boldsymbol{h}^{2}_{<t>}$ whereas the embedding layer is represented by $\boldsymbol{x}^{0}_{<t>}$

2.2.3 Stacked RNN layers with dense connections:

Multiple layers of RNN are stacked on top of each other and the input of every layer is connected to the output of every other layer in a feed-forward fashion to give a densely connected RNN [27]. Such dense connections were introduced into RNNs by [27] to improve the performance of models in the field of language modeling.

At time step *t*, the input to the layer *l* is formed by the concatenation of the outputs of the layer *(l-1)* and all the lower layers including the embedding layer $\boldsymbol{x}^{0}_{<t>}$. The output recurrent step ($\boldsymbol{h}^{l}_{<t>}$) for the l^{th} layer in a densely connected RNN at a time step *t* can still be given as follows:

$$\boldsymbol{h}^{l}_{<t>} = f(\boldsymbol{x}^{l-1}_{<t>}, \boldsymbol{h}^{l}_{<t-1>}) \tag{2.4}$$

However, the vector input to the layer *l* at time step *t* is now written as follows:

$$\boldsymbol{x}^{l-1}_{<t>} = [\boldsymbol{h}^{l-1}_{<t>}; \boldsymbol{h}^{l-2}_{<t>}; \boldsymbol{h}^{l-3}_{<t>}; \ldots\ldots; \boldsymbol{x}^{0}_{<t>}] \tag{2.5}$$

where $\boldsymbol{h}^{l-1}{}_{<t>}$ is the output of the *(l-1)th* layer, $\boldsymbol{h}^{l-2}{}_{<t>}$ is the output of the *(l-2)th* layer and so on. The $\boldsymbol{x}^{0}{}_{<t>}$ is the embedding layer. All such inputs concatenated and given as input to the *lth* layer. An example of a three-layered densely connected RNN network is illustrated in Figure 2-4:

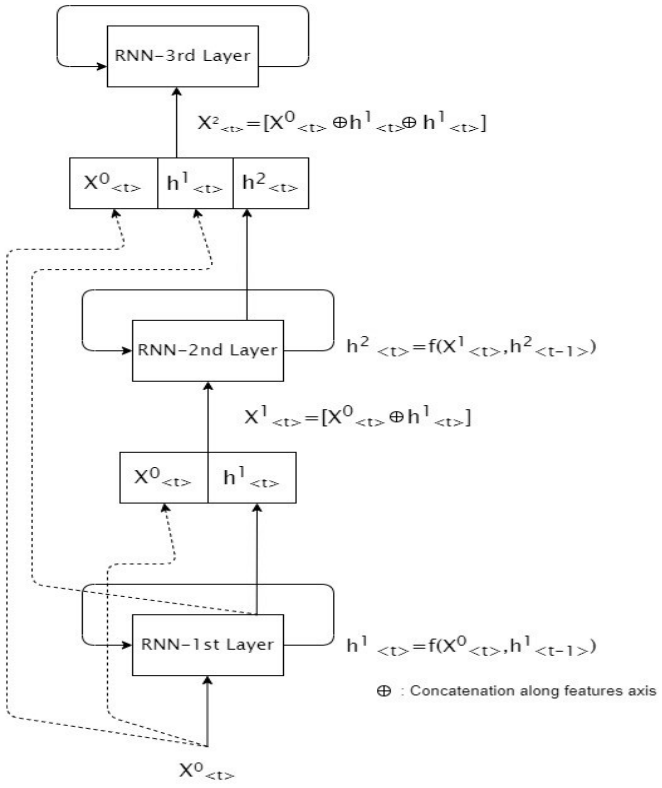


Figure 2-4: Densely Connected RNN Architecture. The input to RNN 2nd layer, is concatenation of the output from RNN-1st layer and the input to RNN-1st layer. Similarly, the input to RNN-3rd layer is concatenation of the output of RNN-2nd layer, output of RNN-1st layer and the input to RNN-1st layer.

2.2.4 Batch Normalization:

Batch Normalization [28] is a technique that is used to speed up the training and allow the networks to generalize better by reducing the internal covariate shift in CNN models. The activations of a layer are standardized using the statistics of mean and variance on the mini-batch of the training sample. This has an effect of reducing the dependency between the parameters of one layer from the other and helps in faster convergence. The batch normalizing transform is given as follows:

$$BN_{\gamma,\beta}(\boldsymbol{h}) = \beta + \gamma \odot \frac{\boldsymbol{h} - \hat{E}[\boldsymbol{h}]}{\sqrt{\hat{Var}[\boldsymbol{h}] + \varepsilon}} \tag{2.6}$$

where $\boldsymbol{h}$ is the activation vector from a layer. $\gamma \; and \; \beta$ are model parameters. $\hat{E}[\boldsymbol{h}]$ and $\hat{Var}[\boldsymbol{h}]$ are the sample mean and variance that is estimated on the current mini-batch.

Though batch normalization was primarily introduced in CNNs, it was also employed into RNNs in the context of language modeling, question answering [29], sentiment classification [30]. A variety of batch normalized variants of RNN exists. The batch normalization can be applied "vertically" i.e. in between two stacked RNN layers and also "horizontally" i.e. between consecutive time steps in the same RNN layer.

2.3 Related Work

To the best of our knowledge, the work that has studied omics data to explain the MDS-UPDRS score for the purpose of PD progression is of Mansu Kim et al [14]. The authors used the combination of DNA genotyping and neuroimaging data from the PPMI database to predict the MDS-UPDRS score. DNA genotyping belongs to the class of genomics and is the study of the genetic variants such as single nucleotide polymorphisms(SNPs) that are associated with the disease phenotype. Neuroimaging approaches such as diffusional MRI allows investigation of neuronal degradation and thus help to understand the underlying mechanism of PD.

Using the imaging genetics approach, the authors identified genetic variants associated with PD. Three SNPs related to the PARK2 gene and one SNP for each of PARK7, SNCA, HtrA2 and GIGYF2 gene were identified. The genetic variant features along with the neuroimaging features were used to construct a linear regression model to explain the MDS-UPDRS score. While performing the imaging genetics and also while constructing the regression model, the data of only baseline visit were included. The linear model thus focuses on the question of how much variance in MDS-UPDRS is explained by genetic variant features and neuroimaging features. The authors reported that about 72.5% of the variance in MDS-UPDRS is explained with an RMSE of 7.82 and a correlation of 0.788 with the actual and predicted value of the MDS-UPDRS score.

The work utilizes the genomics information of the baseline visit of a subject to predict the disease status score for the baseline visit. It will be interesting to investigate how transcriptomic data like RNA-Seq will explain the disease status as compared to genomics data. Moreover, the genomics information of multiple visits is not studied. There is an unmet need to use this omics information on past time points to predict the disease status of future time steps so that we stay ahead of the disease and plan the treatment accordingly.

To the best of our knowledge, there is no study that constructed predictive models using input information of multiple past visits to predict the disease status score of future visits for Parkinson's disease progression. However, some works have been done on Alzheimer's disease. The authors in [16] have proposed a single step ahead predictive model for the purpose of AD disease progression using Recurrent Neural Networks. The input features include longitudinal historical data including clinical information such as demographics, health history, physical examination to predict the disease status score for AD represented by Global CDR score. The number of input features is 78, the target variable is a categorical variable with 5 categories. The number of subjects for the study was 5432 and the average number of visits for a subject was 4.98 while the maximum number reaches as high as 12. The number of samples is greater than the number of input features. The RNN based predictive model managed to learn from the longitudinal information of patient's historical visits to predict AD disease progression in the classification framework.

It will be quite interesting to investigate what kind of RNN can learn from longitudinal information of the transcriptome of a patient for PD disease progression in the regression framework. The transcriptomic dataset in PPMI for our study has only 423 subjects with input features in the range of 35,000. Here, the number of samples is lesser than the number of input features. The average number of visits per subject is 2.8 while the maximum number reaches 5. Moreover, the target variable MDS-UPDRS is a continuous variable.

Chapter 3 Dataset Description

In this chapter, we provide a background of the different longitudinal studies available in the literature that aim to provide a comprehensive resource for Parkinson's Disease. We provide a description of our dataset source -*PPMI* and why we selected it. Followed by a detailed description of the input features and target values for our analysis.

3.1 Overview of Databases for Parkinson's Disease

3.1.1 Resources for Parkinson's Disease Study

Several programs exist that focus on the study of Parkinson's disease and offer a comprehensive resource for researchers to help find biomarkers of PD to help etiology, progression, stratification, treatment, and clinical trials of Parkinson's Disease. There are longitudinal studies where the subjects were recruited and followed clinically for 4 to 5 years to provide data such as biofluid samples, DNA, RNA samples, DAT SPECT imaging, MRI scans, medical history, clinical exams, motor, and nonmotor disease rating scales longitudinally. A brief overview of the past and ongoing longitudinal studies is given in the table below:

Table 3-1: Comprehensive Databases available for Parkinson's Disease

Program	**Web Link**	**Subjects**	**Significance of subjects**
Parkinson's Disease Biomarkers Program (PDBP)	Pdbp.ninds.nih.gov	Subjects (>1000 individuals), PD subjects (>600)	Patients have PD for an average of 5 years or more while they were recruited. Patients have overt PD symptoms with dopaminergic medications
BioFIND	biofind.loni.usc.edu	PD subjects (~119), Healthy controls (~96)	Samples were collected with both "off" and "on" dopaminergic medication. Patients while recruited had an average of 5 years of PD.

Parkinson's Progression Marker Initiative (PPMI)	www.ppmi.info.org	PD subjects (~400), Healthy control (~200)	PD patients must be within 2 years of diagnosis and naïve to dopaminergic medication when recruited. The cohort is seen as "early stage PD" or "*de novo* PD"
Parkinson's Associated Risk Study (PARS)	www.parsinfosource.com	Screened >10,000 PD risk subjects, at least 50 subjects were identified who may receive PD diagnosis	Subjects who didn't develop PD but who are at high risk of developing PD in future.

To get more information about other important databases for Parkinson's Disease, the reader is referred to the article "Finding useful biomarkers for Parkinson's disease" by Plotkin et al [2].

3.1.2 Rationale to choose PPMI database for our study:

For our study of modeling of disease progression in PD, we have used the datasets provided by Parkinson's Progression Marker Initiative PPMI [17]. The PPMI has done a longitudinal study on the early diagnosed Parkinson's Disease patients called "*de novo*" PD cohort for the purpose of improving PD therapeutics and identifying disease progression biomarkers. As part of our problem statement, we would like to predict how the disease would progress in an early diagnosed PD so that it will help the clinicians to administer the treatment accordingly at an early stage. Thus, we hypothesize that PPMI data would be beneficial to study as we aim to identify if transcriptomic data is a potential progression biomarker in PD.

3.2 Description of Datasets in PPMI database

3.2.1 Description of the subjects under study in PPMI:

The PPMI study has enrolled 423 PD subjects till now who were drug naïve (i.e. not much treated with dopaminergic medications) and who have been diagnosed with PD for a period of less than 2 years at the time of enrollment into the study. This is one of the most important differences in PPMI study when compared to other studies such as PDBP or BioFind that the subjects in PPMI are all 'early diagnosed' or 'de novo'. The study has about 196 healthy subjects and plans to recruit more five types of cohorts, viz: prodromal (100 individuals), patients who have scans without evidence of dopaminergic degeneration (i.e., SWEDD, 70 individuals), a genetic cohort of PD patients (600 individuals), a genetic registry (600 individuals).

3.2.2 Description of the Input Features and Target Variable for our Study:

The subjects are followed up for a period of 4 years from the date of enrollment and the clinical data is collected every (3 months, 6 months, 9, months, 12 months, 18 months, 24 months, 30 months, 36 months, 42 months, 48 months) after the date of baseline visit during the follow-up period. The data collected includes a series of assessments (motor and non-motor assessments), Omics data (including DNA Genotyping, RNA Sequencing), and biofluids (plasma, serum, whole blood, urine, and saliva), general neurological and physical examination, etc. However, for our analysis, we only use a subset of the PPMI dataset that includes transcriptomic data, motor assessment questionnaire, MDS-UPDRS score, demographics, etc. We also created a few variables that are derived from primary variables. The complete list of features used for our analysis is presented in Table 3-2 below. For ease of understanding, we grouped the input features into 5 groups.

Table 3-2: Description of Dataset of our Study

Group of features	Description of the feature/ group of features	Number of features/variables	Type of variable

Predictor Variables				
RNA Sequencing	Quantification file of Transcripts	Each file contains 34571 gene names. For each gene, following information is available after RNA-Seq: -gene length -effective length -TPM -NumReads For our analysis, we use Transcripts per Million (TPM), as it is the recommended abundance estimate of the transcripts of the genes. Other estimates are discarded.	34571 features (expressed in units of TPMs)	Continuous
Motor Assessment	MDS-UPDRS-I Questionnaire	This part has questions related to non-motor experiences of daily living and is self-administered by the patient	13	Categorical
	MDS-UPDRS-II Questionnaire	This part has questions related to motor experiences of daily living is self-	13	Categorical

		administered by the patient		
	MDS-UPDRS-III Questionnaire	This part has questions related to motor experiences and is completed by a healthcare professional after examining the patient	33	Categorical
	MDS-UPDRS-IV Questionnaire	This part has questions to be answered by a healthcare professional to assess two motor complications, dyskinesias and motor fluctuations that include OFF-state dystonia	6	Categorical
	Hoehn and Yahr Stage	The "Hoehn and Yahr stage "is now a part of MDS-UPDRS-III and has 5 classes. However, was originally published in 1967 to measure the Parkinson symptoms and disability	1	Categorical
Subject Characteristics	Demographics	Features such as birthdate, gender, and race	11	Categorical

at time of screening	Family status	Features to answer if close family relatives had PD	18	Categorical
	Socio Economics	Features regarding the educational level of the patient	2	Categorical
General Exam	Physical Exam		11	Categorical
	Vital Signs		10	Categorical and Continuous
Derived features	Time Interval between two visits	The difference in time between two successive visits	1	Continuous
	Age	The Age variable is derived from "BIRTHDT" variable of Demographics page of subject. The difference between the birthdate and the visit year.	1	Continuous
Total number of input features			**34,682**	
Target Variable				
Group of features		Description of the feature/ group of features	Number of features/variables	Type of variable
MDS-UPDRS Final Score		The target variable (MDS-UPDRS) reflects the motor symptoms and	1	Continuous

	clinometric properties of PD on a scale of 0 to 272 with 0 being normal and 272 being severe motor and non-motor decline		

Table 3-2 gives the information of the subset of features from PPMI datasets that we use for our analysis. The target variable for our analysis is MDS-UPDRS final score which is a continuous variable whose value ranges from 0 to 272 with 0 being normal and 272 being severe motor and non-motor decline. The primary predictor variables are the gene expression levels i.e. RNA transcripts expressed in Transcripts per Million. In addition to gene expression values, we also use the basic general information of patients that was collected at screening such as demographics, family history, and socioeconomic status. we have also created two variables called "Time interval" and "Age" that are derived from the primary features. The time interval for a visit is created by the difference in time between that current visit and the immediate next visit available. The "Age" variable for a visit is the age of the patient on that visit and is created by the difference in the birth year and the year of that visit.

3.2.3 Number of Samples in our study:

This section describes the sample size for our subset of the dataset. We discuss the frequency of information collection in PPMI.

The data for features described in the table (except the subject characteristics), was conducted by PPMI for all the subjects- *longitudinally* i.e. for various "visits" or "time points" during the study period of 4 years starting from screening(SC), baseline visit(BL), 3-months(V01), 6-months(V02), 9-months(V03), 12-months(V04), 18-months(V05), 24-months(V06), 30-months(V07), 36-months(V08), 42-months(V09), 48-months(V10). However, not all of the feature's data was collected for all the visits. For example, RNA-Seq is performed for only selected visits such as Baseline visit, 6 months visit, 12 months, 36 months, and 48 months whereas Motor Assessment questionnaires were performed twice for all the visits. Hence, we need to perform a merge operation of these datasets and select the samples for whom all the clinical data is available. Table

3-3 below describes in detail the frequency of collection of data for each feature subset in PPMI. This information will help us to understand the data selection criteria we provide in the next subsection Data Selection Criteria

Table 3-3: Frequency of Information Collection in PPMI

RNA-Seq Data	Performed once for annual visits: Baseline (BL), 12-months (V04), 24-months(V06), 36-months(V08).
MDS-UPDRS-I, II	Performed once for all visits
MDS-UPDRS-III	Performed for all visits. But twice on annual visits (Bl, V04, V06, V08, V10) one with "ON" state medication and other with "OFF" state dopaminergic medication [add footnote]. **ON state:** The subject was given dopaminergic medication **OFF state:** at-least 6 hours after the last dose of dopaminergic medication
MDS-UPDRS-IV	Performed once for all visits
General Exam	Performed once for all visits

3.3 Data Selection Criteria

This section provides the criteria for data selection from PPMI to build the subset of the dataset for our analysis:

1) **Data Download Date:** The datasets were downloaded from PPMI on the date: 2018-06-05 and the latest update of all datasets are downloaded on 2020-01-28. The analysis and the experiments were updated to include the changes in the dataset.
2) **Subject Categories:** The subjects belonging to Parkinson's Disease (PD) category were only considered for the analysis. The number of PD subjects is 423. The subjects belonging to other categories such as HC, SWEDD, Genetic PD, Genetic Registry PD were discarded from our study.
3) **Transcriptomic data samples:** For our analysis, we included all the data samples for which RNA-Seq was performed. The RNA-Seq analysis yields a couple of outputs. We chose only a subset of output from RNA-Seq and created the RNA-Seq "Subset-1". The details are explained in the next chapter.
4) **Disease status data samples:** For our analysis, only those samples for whom MDS-UPDRS-III performed in the "OFF" medication state were selected to form the disease status target variable. For a disease progression prediction problem, we need to have the motor and nonmotor condition of a subject while he/she was NOT on any dopaminergic medication. As per PPMI protocol, a delay of "6 hours" with no medicine is enough to withdraw the effect of the last dose of medicine on the patient.
5) **Data Sample ID:** Each data sample is represented by a unique key which is a combination of two fields - "PATNO" and "EVENT_ID". The field "PATNO" defines the subject's name that the data belongs to and the field "EVENT _ID" defines the time point or visit id of the data sample. This applies to all the datasets. We mapped all the data samples from the predictor variables, to the data samples in the MDS-UPDRS target variable with the help of the primary key on "PATNO" and "EVENT_ID". A more detailed explanation is given in the chapter Data Pre-Processing
6) For our analysis, we need 3-Dimensional data, the first dimension is the number of PD patients i.e. 423, the second dimension is the number of visits and the third dimension is the number of predictor variables. To this end, the final dataset has 423 3-D samples with a variable number of visits or 1709 2-D samples. A more detailed explanation is given in the chapter Data Pre-Processing.

Chapter 4 Data Pre-Processing

In this chapter, we will discuss in detail the various data preprocessing steps involved to prepare the dataset for our study. The chapter is organized into the following major sections:

1) **Preparation of Predictor Variables:** The RNA resource in the PPMI database is quite rich with a variety of data variables. We point to the data variables in this RNA resource that were used as predictor variables in our study. Furthermore, we present a discussion on a few derived input variables that were created by us.
2) **Preparation of Target Variable:** We will discuss the preprocessing steps on the target variable dataset of the MDS-UPDRS assessment.
3) **Merging of Predictor Variables and Target Variable:** The structure of the dataset that is finalized for our study.
4) **Missing Value Imputation:** We present the analysis of missing values and the imputation technique used.
5) **Feature Selection:** We present the feature selection scheme employed for our study.
6) **Other Important Pre-Processing Steps:** We present a brief discussion on how we dealt with the variable number of visits, performed data splits and normalization of data

4.1 Preparation of Predictor Variables

4.1.1 Preparation of subset of RNA-Seq dataset:

In the PPMI RNA-Sequence resource, the abundance estimates of RNA transcripts are available under the head of the "quants" folder. The quants resource has quantization files per subject and per visit. Each quantization file has rows of 34571 target genes with the following 5 columns of information per gene:

Name: The name of the target gene

Length: The length of the target gene in nucleotide base pairs

Effective Length: The computed length of the target gene by considering all biases

TPM: The estimate for Transcripts per Million for the target gene

NumReads: The estimate of the number of raw reads mapping to each transcript

For our analysis, we need only the **TPM** column from the file for 34571 target genes per patient per visit.

After analyzing the RNA resource, we know that PPMI subjects have undergone RNA-Sequencing for selected visits only i.e. for the first visit at Baseline (BL) and then subsequent visits scheduled at 6 months(V02), 12-months (V04), 24-months (V06) and 36-months (V08). Thus, each PPMI subject had 5 samples sequenced for transcriptomic studies and therefore there are 5 quantification files per subject. The number of PD subjects is (PD=423) and thus 423*5=2115 files are expected, however, there are 1709 such files for PD patients belonging to the visits (BL, V02, V04, V06, V08) implying missing RNA-Seq for some visits for few PD patients.

To this end, we extracted the **TPM** column from each file for 34571 target genes from all available files and merged into one file and named as “RNA-Seq subset”. Figure 4-1 below illustrates the merging performed.

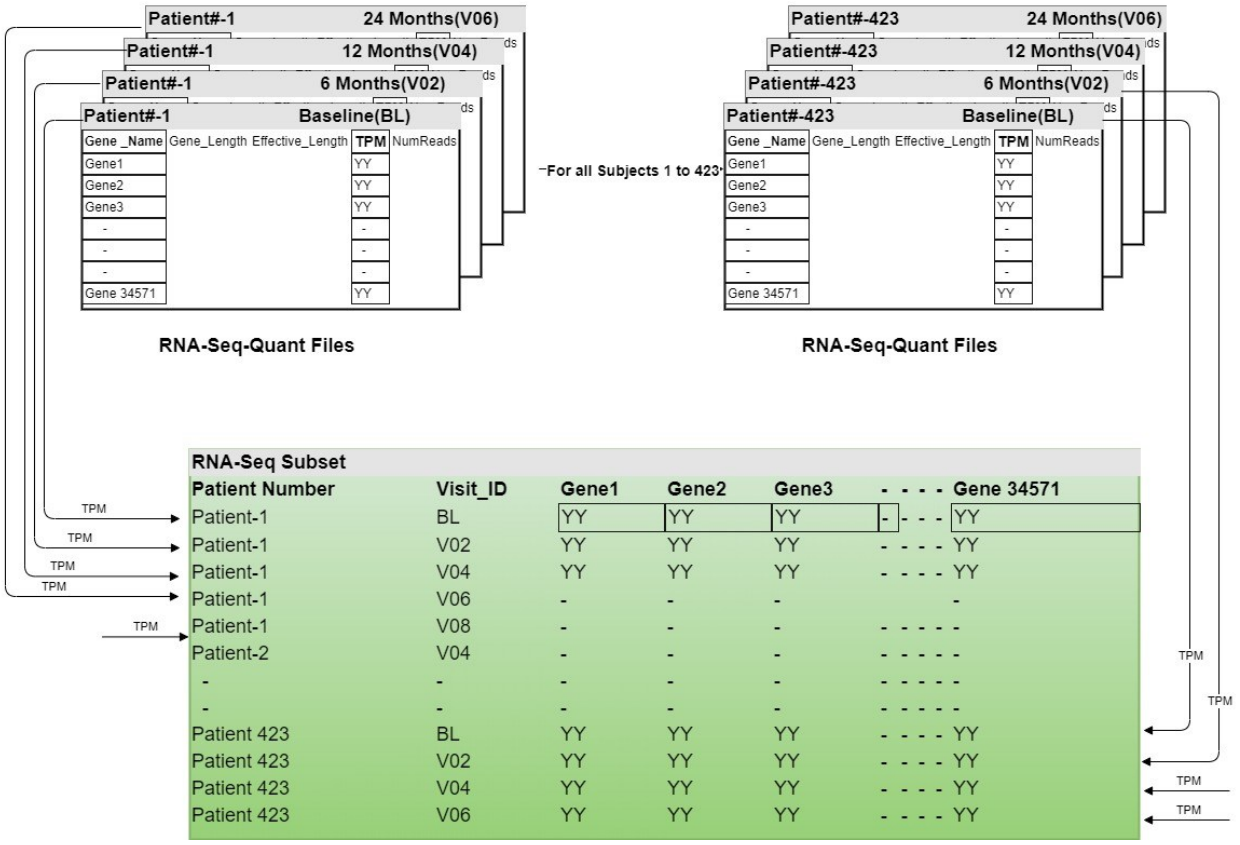


Figure 4-1: Preparation of "Subset of RNA-Seq" dataset from PPMI RNA-Resource. Each quant file is available per visit per subject. The TPM estimate for all 34571 genes per subject per visit is copied and merged into the data subset RNA-Seq Subset.

The resultant subset of RNA-Seq has 34571 columns and about ~1700 samples. Each sample in the subset dataset is mapped to a unique combination of Patient ID and Visit ID. Each of 34571 columns represents the expression of a target gene in Transcripts per Million.

4.1.2 Derived Variables:

We created two variables "Age" and "Time Interval" by using the information from the existing variables.

Age:

The age of the subject plays an important role in PD as studies confirm that advancing age is a major risk for developing PD. Thus, we decided to include such a variable into our dataset to help

our predictive model. The "Age" variable for a visit is the age of the patient on that visit and is created by the difference in the birth year and the year of that visit.

Time Interval:

This variable is the difference in time between the baseline visit of the subject and the future visits of the subject. For example, if $X_{<n>}$ represents the vector of baseline features to predict the disease status at future visits where the immediate next year disease status is denoted by $\mathbf{y}_{<n+1>}$. The second-year disease status by $\mathbf{y}_{<n+2>}$ and the third year by $\mathbf{y}_{<n+3>}$ and so on. Then the time interval between first year and baseline visit is denoted by $\Delta t_{<n+1>} = t_{<n+1>} - t_{<n>}$.Similarly, the time interval between second-year and baseline visit is $\Delta t_{<n+2>} = t_{<n+2>} - t_{<n>}$ and so on.

4.2 Preparation of Target Variable

4.2.1 MDS-UPDRS:

The Movement Disorder Society Unified Parkinson Disease Rate Scale (MDS-UPDRS) reflects the motor and nonmotor symptoms and clinometric properties of PD on a scale of 0 to 272 with 0 being normal and 272 being severe motor and non-motor decline. The MDS-UPDRS assessment comprises 4 parts, Part-I (**13** questions related to non-motor experiences of the daily living, which is self-answered by the patient/ caregiver), Part-II (**13** questions related to motor experiences of the daily living, again self-answered by patient/caregiver), Part-III (**34** questions related to motor experiences of daily living which are answered by a healthcare professional on examining the subject) and Part-IV(**6** questions to be answered by a healthcare professional to assess two motor complications, dyskinesias and motor fluctuations that include OFF-state dystonia). Each item is rated on a 5-point scale (ranging from 0 to 4) with higher scores representing severe impairment.

MDS-UPDRS Final Score Calculation

We derived the final score by adding all the answers of Part-I, II, III, IV for a subject's visit. However, only the studies that were conducted in the "OFF" medicine state were considered.

4.2.2 Disease Progression Curves:

The disease progression curve for an individual is the variation of the MDS-UPDRS final score with respect to the visits or duration of disease. In this section, we performed a quick analysis of the general trend in the progression curves of PD subjects. We choose randomly few subjects and plotted how the MDS-UPDRS final score varies during the period of study, starting from the Baseline (BL) visit to the last visit available for each of the subjects. Figure 4-2 shows the variation of the MDS-UPDRS final score for a sample of 8 PD subjects.

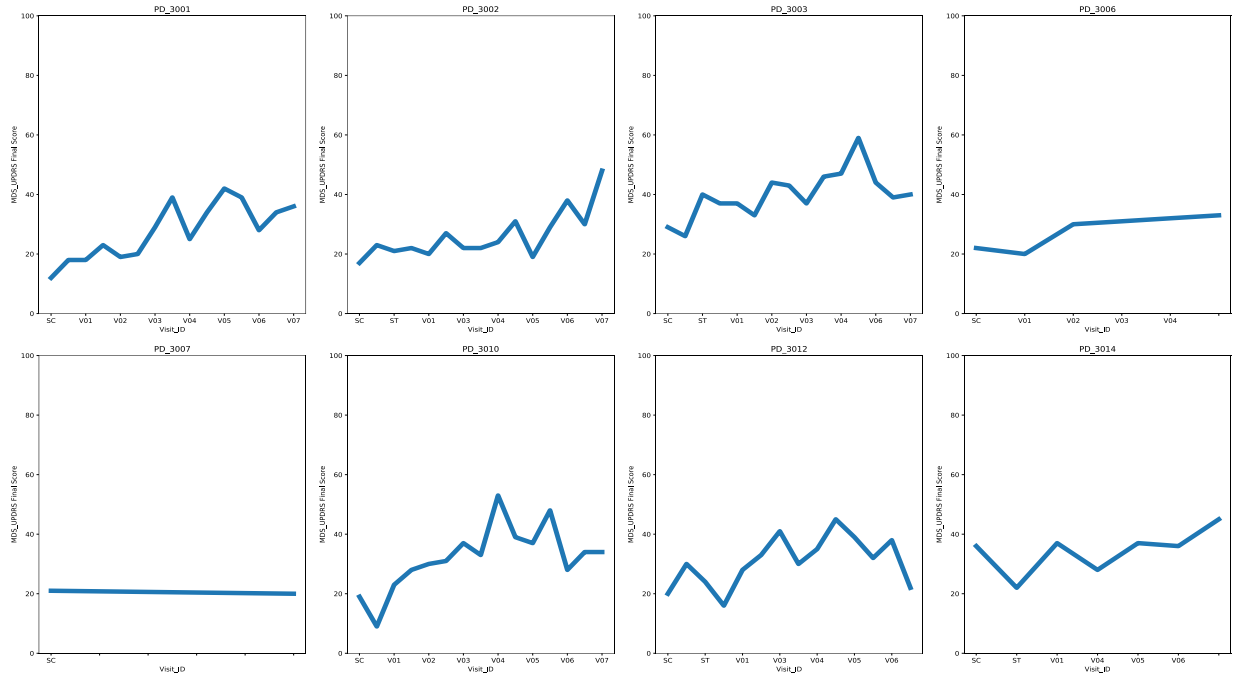


Figure 4-2: Variation of MDS-UPDRS final score for a sample of Parkinson Disease subjects for the period of study starting from Baseline visit to last visit. MDS-UPDRS final score is plotted on the y-axis for the corresponding visits on the x-axis.

From the figure, we observe that there is an increasing trend in the progression of PD subjects. We know that on average the MDS-UPDRS final score for PD patients in the PPMI cohort increases by 4.7 points per year [31].

Denoising of Disease Progression Curves:

We know that PD is progressive in nature, hence the disease progression curve for PD subjects is expected to have an increasing trend [31]. However, such as any time series curve, the progression curve also has noise in it. We tried to remove the noise by applying the data smoothing

techniques. We know that one of the popular smoothing techniques used in time series analysis and forecasting is moving averages [32]. As such, we employed the trailing moving average method with a window size of 3 as explained below:

$$\text{Moving\_Average}(t) = \text{Average}(\text{observation}(t-2), \text{observation}(t-1), \text{observation}(t)) \qquad 4.1$$

After we performed data smoothing on all PD subjects as per equation 4.1, we plotted the progression curves before and after the data smoothing for a sample of subjects and presented in Figure 4-3.

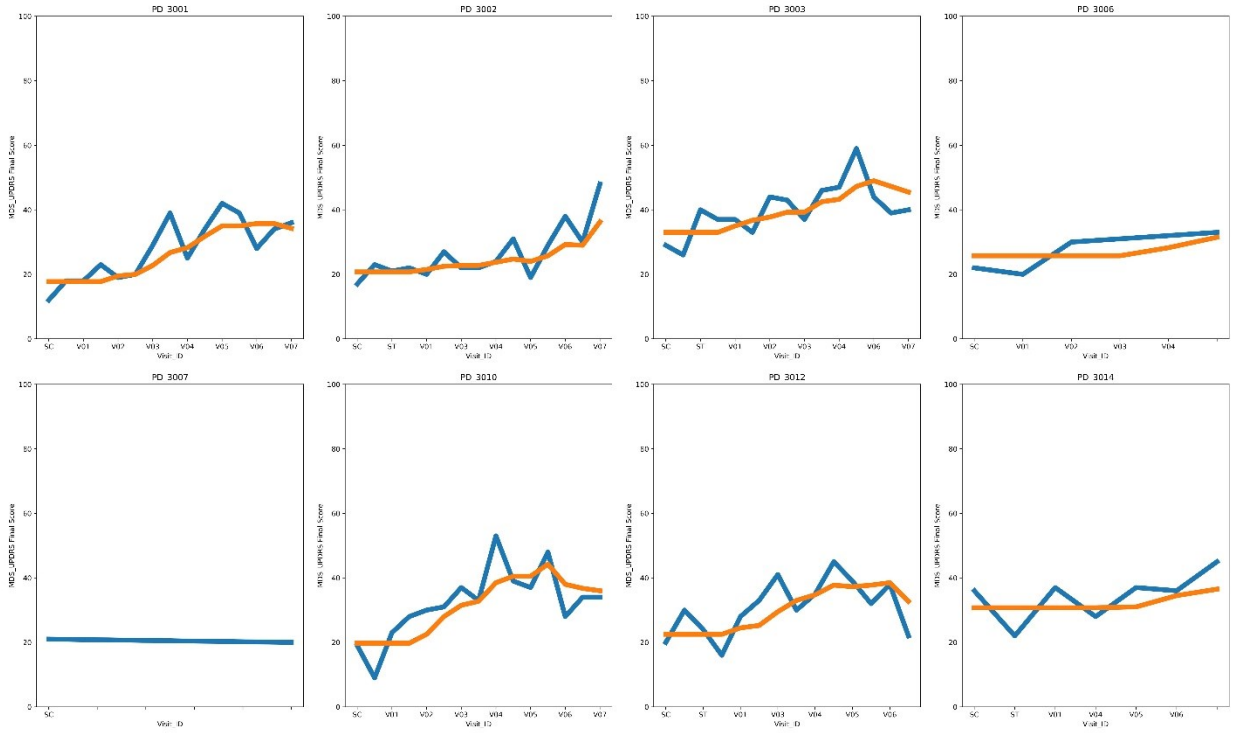
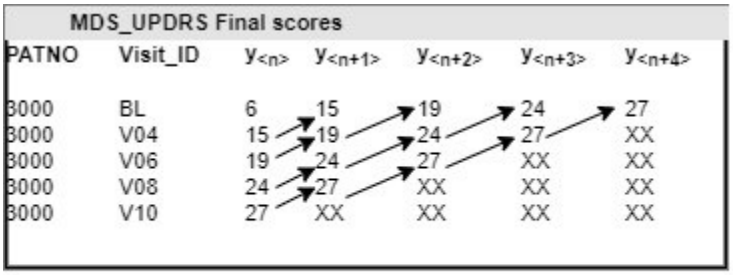


Figure 4-3: Data smoothing on the disease progression curve. A sample of 8 subjects are plotted such that MDS-UPDRS final score plotted on y-axis versus "point of visit" on x-axis. Trailing moving average method with 3 window size is applied to denoise the raw curves. The curve in blue is raw curve whereas the curve in orange is the noised version of the raw curve.

4.2.3 Time Window Shift:

Given a time series prediction problem, if the goal is to predict the observation at the next time step then it is called a single time step prediction problem. If the goal is to predict the observation for multiple time steps ahead then it is called multiple time step prediction problems. In this section, we prepare our MDS-UPDRS dataset for a single time step prediction problem of predicting disease status annually. Let the observation of MDS-UPDRS final score at n^{th} visit be denoted as $\mathbf{y_{<n>}}$ then the value of MDS-UPDRS final score at the next annual visit is denoted as $\mathbf{y_{<n+1>}}$ and further rolled in time/ future annual visits is denoted by $\mathbf{y_{<n+2>}}$, $\mathbf{y_{<n+3>}}$, $\mathbf{y_{<n+4>}}$ and so on. The time window shift is performed on all annual visits of the MDS-UPDRS final score. The time window shift for one of the PD individuals is shown in Figure 4-4.



Notation	Meaning
$y_{<n>}$	Target value at the current time step
$y_{<n+1>}$	value next year
$y_{<n+2>}$	value in next 2 years
$y_{<n+3>}$	value in next 3 years
$y_{<n+4>}$	value in next 4 years

Figure 4-4: Time Window shifts performed on annual visits of MDS-UPDRS final score for a subject identified by number 3000. For the value of MDS-UPDRS at the Baseline(BL) visit if denoted by $y_{<n>}$, then the value of MDS-UPDRs for the immediate future visit will be of at V04 as denoted by $y_{<n+1>}$.

4.3 Datasets Merging

The datasets for our analysis as described in Table 3-2 that includes the predictor variables (RNA-Seq subset, motor assessments, general exam etc.) and the target variable (MDS-UPDRs final score) were all merged into a single dataset. Each data sample is mapped to a combination of Patient ID (PATNO) and Visit (Visit _ID). We performed the merge in such a way that all data samples of RNA-Seq subset were included i.e. a left join was performed with RNA-Seq as the *left* dataset and the other datasets to the *right*. Since RNA-Seq was performed for only selected number of visits (Baseline(BL) , 12 months(V04), 24 months(V06), 36 months(V08), 48 months(V10)) ,

We have 5 data samples for each subject. There are instances where a subject during a visit had RNA-Seq data available but had missing clinical data (either one of motor assessment or general exam) and thus resulted in a few missing values which were imputed as per Section 4.4. The figure below illustrates the method for the preparation of the final dataset.

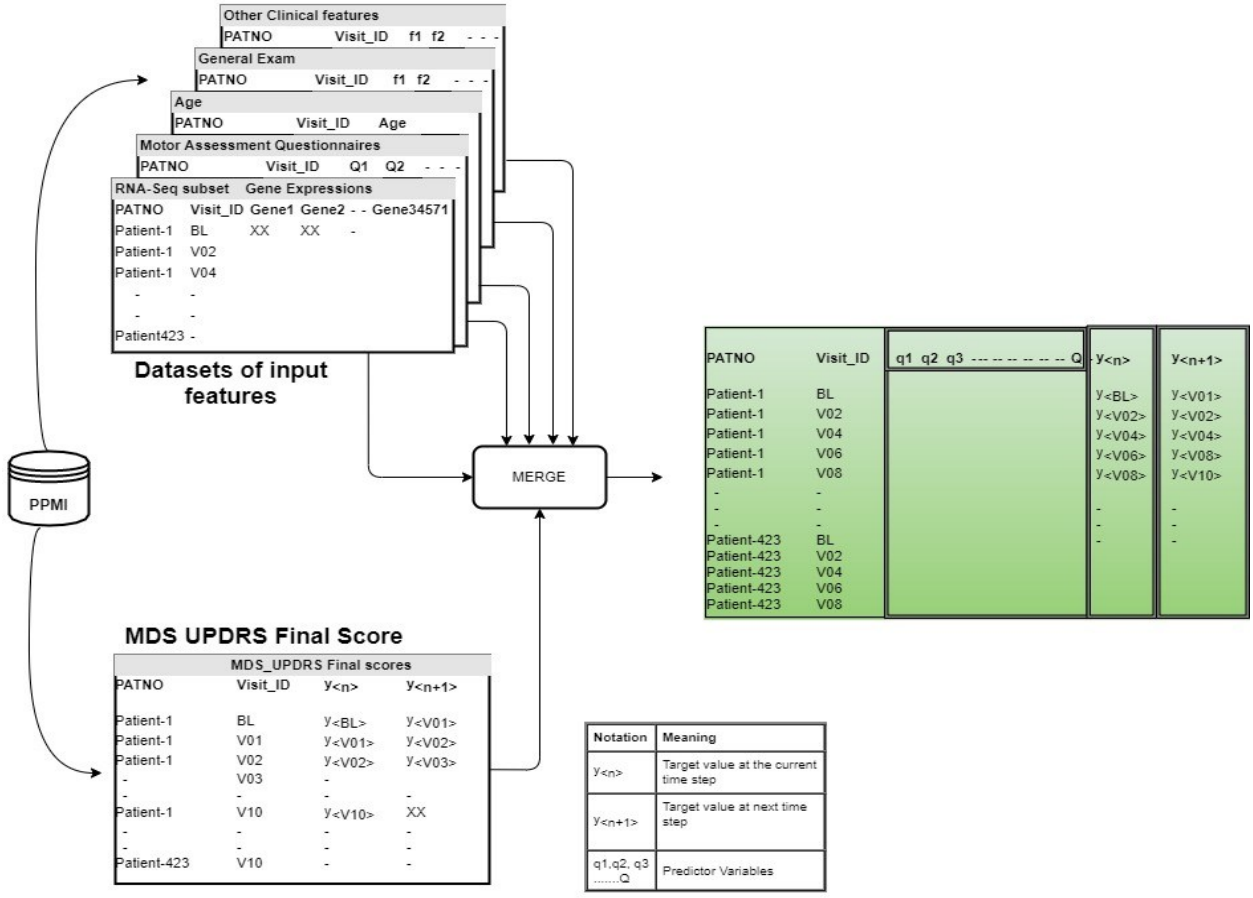


Figure 4-5: Overview of Datasets Merging

4.4 Missing Value Imputation

We performed an analysis of missing values in the datasets to find the percentage of missing values as described below. It is calculated by dividing the number of cells in the subset that have missing values by the total number of cells in the subset.

Table 4-1: Analysis of Missing Values for various subsets of data of our finalised dataset

Dataset	**% of Missing Values**
RNA- Seq Subset	0
Motor Assessments subset	25
Subject Characteristics at time of screening subset	0
General Exam subset	12

To impute the missing value of a variable (say qth variable) for a visit (say V06th visit) belonging to a patient-X we impute the missing value by considering the values of the variable (qth) from all the visits of only that patient. The rationale to not consider all the other subject data is that there is heterogeneity within the group. For example, a PD subject who might have a high rate of disease progression might have a value for a variable far ahead than that the PD subject who has a slow or moderate rate of progression. So, to minimize the bias, we will use the longitudinal data of that subject only and not of all the subjects. Such a strategy was used by authors of [16] for Alzheimer's Disease progression.

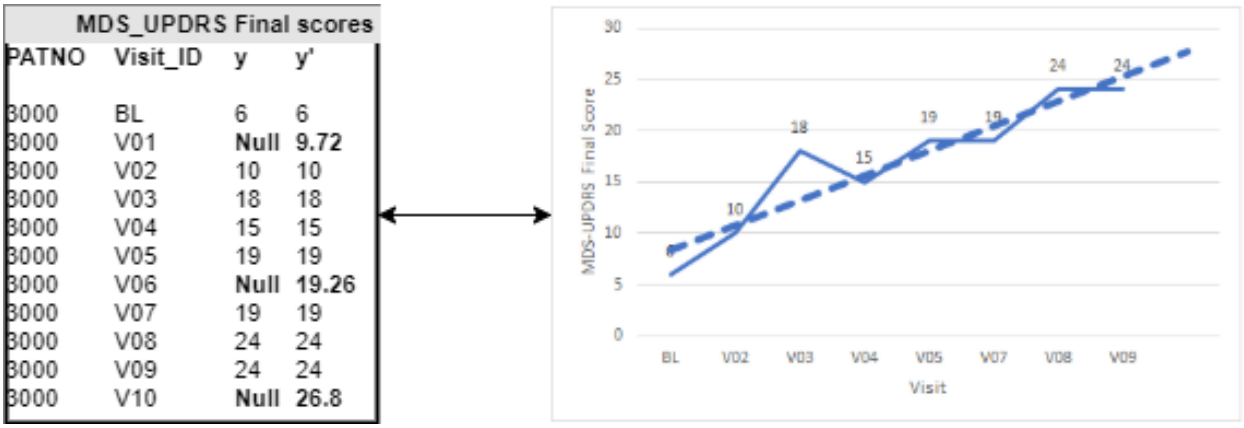
Further, the imputation scheme also depends upon the data type of the variable, i.e. whether the variable is a continuous type or categorical type(ordinal or nominal). In the next few sections, we describe the imputation scheme employed in our study for each of the data types of variables.

4.4.1 Imputation for Continuous Data type variable:

While it is a common practice to impute the value of a continuous type with the mean value of the other visits, we propose to impute the value by performing "linear interpolation". Let us understand with an example of imputing missing values for the continuous variable MDS-UPDRS score.

The score changes with different visits of the subject and is dependent on that subject only, so we use the technique of linear interpolation by considering the MDS-UPDRS score for all the visits for that subject and then impute the value for a visit that has the score missing with the help

of a line of best fit as shown in Figure 4-6. We know PD is a progressive disease and the rate of progression is usually positive or zero i.e. the disease status will either degrade with time or will remain the same. Using the technique of interpolation instead of mean value will help to follow the disease status trend.



MDS_UPDRS Final scores			
PATNO	Visit_ID	y	y'
3000	BL	6	6
3000	V01	Null	9.72
3000	V02	10	10
3000	V03	18	18
3000	V04	15	15
3000	V05	19	19
3000	V06	Null	19.26
3000	V07	19	19
3000	V08	24	24
3000	V09	24	24
3000	V10	Null	26.8

Figure 4-6: Missing value imputation technique for continuous data type variable. Shown is the technique of interpolation for imputing missing visit values for a continuous data type variable say MDS-UPDRS score. The y and y' in the table represents the values before and after the imputation. The values in y are used to fit a line of best fit as shown by dashed line in graph. The time points that have missing values (V01, V06, V10) are imputed from the line of best fit accordingly.

4.4.2 Imputation for Categorical Data type variable:

The missing value of a categorical variable of a subject at a current visit is imputed with the value of the categorical variable from the previous visit of the same subject. We know PD is a progressive disease and the rate of progression is usually positive or zero i.e. the disease status will either degrade with time or will remain the same. Thus, the scheme of imputing the missing categorical value with the value of the previous visits is justified.

4.5 Feature Selection

We performed feature selection on the gene expressions to select the most relevant RNAs that contribute to the disease status variable of MDS-UPDRS. There are studies that suggest that out of thousands of the genes in an experiment, only a few of them contribute to the phenotype [33]–

[35] . For example, in the binary classification of cancer subtype, usually, 50 informative genes are enough to make the prediction [33]. Selecting a more representative feature set not only helps to increase the prediction accuracy of the targeted disease but also reduces computational power. The feature selection technique used for our analysis is Minimum Redundancy Maximum Relevancy (MRMR) [36]. The method tends to select a set of features such that the features have a maximum correlation with the output variable and at the same time have the least correlation among the features themselves.

4.5.1 Minimum Redundancy Maximum Relevancy (MRMR) Method:

MRMR [36] is a supervised technique that has been extensively used for gene selection in DNA microarray gene expression profiles. The algorithm optimizes the combined criterion of maximum relevance and minimum redundancy using a greedy search and ranks the genes. The technique can be applied to feature set that are either all continuous features or all discretized features. For the features that are continuous, the relevancy between each feature and the output variable is calculated by F-statistics and the redundancy between the continuous features is calculated by Pearson's correlation coefficient. On the other hand, if the features are all discretized, then relevance between each feature and the output variable is calculated using mutual information between the feature and output variable. The redundancy for discretized features is also calculated by the mutual information between each discretized feature. An objective function is made by combining the relevance and redundancy metrics and a greedy search is performed to select features one by one by maximizing the objective function. A summary of the MRMR schemes for different data types is provided in Table 4-2.

4.5.2 Notations:

Let N be the number of features in the whole gene set Ω where the i^{th} and j^{th} gene expression is denoted by i and j, respectively. The class variable or disease status variable is denoted by h. Let S be the subset of features to be selected and |S| be the number of features to be selected. With these notations, the MRMR schemes for different types of data may be summarised as follows in Table 4-2.

Table 4-2: MRMR Criterion Functions for Different Data Types

	Relevance Metric	**Redundancy metric**	**Criterion Function**
Continuous Data	F-statistics F(i,h)	Pearson's Correlation c(i,j)	$\max_{i \in \Omega_S}[F(i,h) - \frac{1}{\|S\|}\sum_{j \in S} \| c(i,j) \|]$
Categorical Data	Mutual Information I(i,h)	Mutual Information I(i,j)	$\max_{i \in \Omega_S}[I(i,h) - \frac{1}{\|S\|}\sum_{j \in S} I(i,j)]$

4.5.3 Data Pre-Process for Feature Selection:

The dataset contains 34571 RNA-Seq features and the output variable MDS-UPDRS final score. The features as well as the output variable are continuous in nature in the original dataset. We preprocessed the data such that each feature has zero mean and unit variance. We also performed the discretization of the dataset to perform the discretized analysis of MRMR. In the process of discretization, the observations of each feature were discretized into one of the three states as an under-expressed, baseline, and overexpressed states depending upon the mean (μ) and standard deviation (σ) of that feature. The scheme to discretize the RNA-Seq dataset is given below.

Table 4-3: Scheme to discretize the RNA-Sequence dataset for the purpose of feature selection

Under Expressed RNA transcript	**Baseline RNA transcript**	**Over Expressed transcript**
If the value of the observation in a feature is lesser than (μ-σ/2) then the observation is discretized to state -1	If the value of the observation in a feature lies between (μ-σ/2) and then (μ+ σ/2) then the observation is discretized to state 0	If the value of the observation in a feature is greater than (μ+ σ/2) then the observation is discretized to state 1

4.5.4 Implementation of Feature Selection Scheme:

The feature selection scheme was implemented in R with the help of mRmRe library [37] using the classical MRMR high-level APIs. The analysis ran on a high-performance cloud computing cluster at Compute Canada with the following configuration.

Table 4-4: Computing Configuration for Feature Selection

No of CPU Cores	RAM per CPU core	Time for Job
4	4*65 G= 260GB	48 Hours

We performed both continuous and discretized analysis of MRMR on our RNA-Seq dataset. Out of the 34571 RNA Transcripts, the algorithm searched for the top 5000 RNA transcripts that maximized the criterion of maximum relevance with the disease status variable MDS-UPDRS and minimum redundancy among the RNA transcripts. As two analyses were carried out separately, one for continuous data and another for discretized data, we have two subsets of RNA transcripts each with 5000 top-ranked RNA transcripts. An intersection was performed between these two subsets to yield the final subset of RNA transcripts. The final subset has a total of 3736 transcripts.

4.6 Other Important Pre-Processing Steps

4.6.1 Handle Variable Number of Visits:

The total number of subjects for the study are 423 subjects. Each subject has a variable number of visits with an average number of 3.2 visits per subject. The maximum number of visits available is 5 visits and the minimum number of visits available is 1 visit. Thus, while training the RNN, we list the subjects in the increasing order of the number of visits and 5 batches are created. The first batch has the data from the subjects that have at least 1 visit. The second batch has the data from the subjects that have at least two visits and so on and so forth. Furthermore, each batch is further divided into mini-batches of 16 subjects.

4.6.2 Train, Validation, and Test Split:

For initial model training and hyperparameter tuning, we divided the subjects into training, validation, and testing subjects randomly as 60%, 20%, 20% respectively.

4.6.3 Normalization:

We also performed data normalization after splitting the datasets. We employed min-max scaling for the continuous data variables and one-hot encoding on the categorical variables.

Chapter 5 Proposed Methods

In this chapter, we present a discussion on how we modeled the prediction problem for the disease progression in PD. We will present the mathematical formulation of the prediction problem using Recurrent Neural Networks. Further, we will discuss the background and related work for our proposed RNN architecture. We will then study in detail the mechanics of our proposed Densely connected RNN architecture. Also, we will present the training and hyperparameter tuning pipeline for our proposed work.

5.1 Modelling of Problem Statement

5.1.1 Problem Statement:

We investigate whether artificial neural networks especially Recurrent Neural Networks (RNN) can predict the rate of disease progression in Parkinson's disease by leveraging longitudinal temporal information of RNA sequence data. Particularly, we aim to predict the disease status variable (MDS-UDRS score) of a patient for the immediate future hospital visit by considering the RNA sequence data of multiple previous visits.

In other words, the problem is modeled in a way that when a patient's Baseline year (1^{st} year) RNA-seq data is given to the predictive model, it predicts the next year's MDS-UPDRS score of that patient. When baseline year and 2^{nd} year's RNA-seq data is given to the model, it predicts the 3^{rd} year's MDS-UPDRS score and likewise, when Baseline year's, 2^{nd} year's and 3^{rd} year's data is given as input, then the model predicts the 4^{th} year's MDS-UPDRS score. Thus, as we move forward in time sequences, the model leverages the temporal patterns in the historical RNA sequence data and tries to predict the immediate future time disease status. This strategy has been previously used for disease progression in Alzheimer's Disease by the authors of [16].

5.1.2 Mathematical Notation for the Problem Statement:

Given there are N annual visits for a subject, the input at time $t_{<n>}$ where n=1, 2, 3 . . . N is a feature vector denoted by $X_{<n>}$. The feature vector $X_{<n>}$ has a Q number of features that combine both RNA-Seq features and non-RNA features represented at the nth visit of the patient. For example, $X_{<1>}$ is a feature vector with Q features at the first visit of the patient. Further, if the "<N>" visit represents the current visit of the subject, then the "<N+1>" visit represents the immediate next year's visit. The MDS-UPDRS final score at the N^{th} is represented by $\mathbf{y_{<n>}}$ and the future value of MDS-UPDRS is represented by $\mathbf{y_{<n+1>}}$, then the proposed predictive model in its most basic form can be written as follows:

$$\hat{y}_{<n+1>} = f(X_{<1>}, X_{<2>}, X_{<3>}, \ldots X_{<n>}) \tag{5.1}$$

Where the output $\hat{y}_{<n+1>}$ is the predicted value for the MDS-UPDRS score of the next visit of the patient and the function f(.) represents the core block of our proposed model and what follows next is added on top of this core model. We go further to add time intervals between two consecutive visits in our predictive model to help the model deal with the non-uniform time intervals between two consecutive visits of the patients. If the time interval between the first visit and the immediate second visit is denoted by $\Delta t_{<1>}$ then the time interval between N^{th} visit and $(N+1)^{th}$ can be denoted by $\Delta t_{<n>}$. The proposed model in equation (5.1) can now be written as:

$$\hat{y}_{<n+1>} = f(X_{<1>}, X_{<2>}, X_{<3>}, \ldots X_{<n>} ; \Delta t_{<1>}, \Delta t_{<2>}, \Delta t_{<3>}, \ldots\ldots \Delta t_{<n>}) \tag{5.2}$$

The model in equation (5.2) is modified further to include the ground-truth value of MDS-UPDRS scores of the current and previous visits. For example, to predict the MDS-UPDRS score of $(N+1)^{th}$ visit, we input the model with MDS-UPDRS score of N^{th} visit and all the previous visits $(N-1)^{th}$,$(N-2)^{th}$,……..1^{st}.The final version of our proposed model is as follows:

$$\begin{aligned} \hat{y}_{<n+1>} = f(&X_{<1>}, X_{<2>}, X_{<3>}, \ldots X_{<n>} ; \\ &\Delta t_{<1>}, \Delta t_{<2>}, \Delta t_{<3>}, \ldots\ldots \Delta t_{<n>} ; \\ &y_{<1>}, y_{<2>}, y_{<3>}, \ldots\ldots\ldots y_{<n>}) \end{aligned} \tag{5.3}$$

The model in equation (5.3) can be represented pictorially in Figure 5-1.

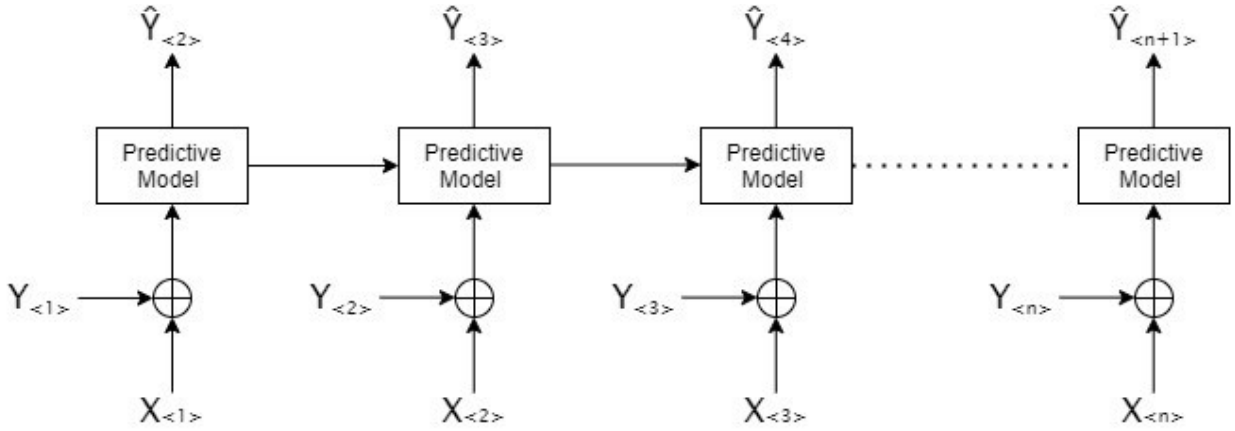


Figure 5-1: Block diagram of the problem statement. The predictive model takes the RNA seq input at 1st time step ($X_{<1>}$) along with the MDS-UPDRS value of 1st time step i.e. $y_{<1>}$ to predict the value of MDS-UPDRS for the 2nd year i.e. $\hat{\boldsymbol{y}}_{<2>}$. When the time step moves to next year i.e. at t=2, we now have the actual value of MDS-UPDRS that was predicted last year. This actual value of MDS-UPDRS at time step 2 along with RNA-Seq data for 2nd year is given to a predictive model so it predicts the value of MDS-UPDRs for the 3rd year. At time step t=2, the model has the memory of RNA-seq patterns from the first visit. The model keeps on updating its memory as it moves horizontally, and which helps it in predicting the next year's score by understanding the patterns from RNA seq data from previous visits.

5.2 Proposed RNN Model: Densely Connected Recurrent Neural Networks for PD Progression

In this section, we present the details of our proposed RNN architecture that is inspired by DenseNet [38]. The novelty of our work lies in combining RNNs with the recent advancements done on Convolution Neural Networks (CNN) such as dense connections, batch normalization, etc. The detailed explanation to provide the necessary mathematical background on the concepts of densely connected RNNs, and batch normalization is presented in Section 2.2.

The idea of skip connections and residual connections were introduced primarily in computer vision problems to help train deep CNN in the field of image recognition [39]–[41]. Followed by the introduction of dense connections in CNN where the input of every layer was connected to the output of every other layer in a feed-forward fashion using skip connections to give a state of the

art performance in image recognition and object detection tasks [38] . Such advancements showed several advantages in CNNs such as strengthening feature propagation, reduction in the number of parameters in the model, facilitating training with fewer data samples, encouraging feature reuse, etc.

As the improvements were significant in CNNs, the skip connections were also introduced into RNNs by [27] to improve the performance of models in the field of language modeling. When multiple layers of RNN are stacked on top of each other and the input of every layer is connected to the output of every other layer in a feed-forward fashion, it is called as densely connected RNN [27]. The detailed mathematical background on densely connected RNNs is provided in the Section 2.2.3

To the best of our knowledge, the densely connected RNNs are not employed in the disease progression of PD. Motivated by the improvements provided by dense skip connections in CNNs and considering the nature of genetics data which is characterized by fewer data samples, huge feature size, and variable-length visits of patients, we hypothesize that the densely connected RNNs may help in the PD disease progression.

Batch Normalization [28] is a technique that is used to speed up the training and allow the networks to generalize better by reducing the internal covariate shift in CNN models. The mathematical background on batch normalization is provided in Section 2.2.4. Though batch normalization was primarily introduced in CNNs, it was also employed into RNNs in the context of language modeling, question answering [29], sentiment classification [30]. A variety of batch normalized variants of RNN exists. The batch normalization can be applied “vertically” i.e. in between two stacked RNN layers and also “horizontally” i.e. between consecutive time steps in the same RNN layer.

Motivated by the effectiveness of batch normalization layers in CNNs and RNNs to consistently speed up the training and generalize better, we integrated batch normalization into our proposed densely connected RNN model architecture.

5.2.1 Proposed Densely Connected RNN Model:

Motivated by DenseNet [38], we propose a densely connected RNN architecture for PD progression modeling. The architecture is composed of multiple Dense Blocks (DBs) stacked on top of each other such that the input of every DB is connected to the output of every other DB in a feed-forward fashion. The output of the last DB is connected to a fully connected normal neuron. Furthermore, each of the DB is composed of multiple Composite Blocks (CBs) stacked on top of each other such that the input of every CB is connected to the output of every other CB in a feed-forward fashion. To this end, a CB is a composite layer that contains an RNN layer followed by a Batch normalization layer. Let us understand each of the above segments in detail in the following subsections:

Composite RNN Blocks(CB):

Motivated by [42], we define $\boldsymbol{H}_{<t>}(.)$ as a composite block containing an RNN layer followed by a batch normalization layer. We apply batch normalization to the output activations of the RNN layer at time step *t*. The output of the composite block ($\boldsymbol{H}_{<t>}$) at time step *t* can be written as follows:

$$\boldsymbol{H}_{<t>} = BN_{\beta,\gamma}(\boldsymbol{h}_{<t>}) \tag{5.4}$$

where

$$BN_{\gamma,\beta}(\boldsymbol{h}) = \beta + \gamma \odot \frac{\boldsymbol{h} - \hat{E}[\boldsymbol{h}]}{\sqrt{\hat{Var}[\boldsymbol{h}] + \varepsilon}} \quad \text{from (2.6)}$$

$$\boldsymbol{h}_{<t>} = f(\boldsymbol{x}_{<t>}, \boldsymbol{h}_{<t-1>}) \quad \text{from (2.1)}$$

Here, $\gamma \; and \; \beta$ are the model parameters. $\hat{E}[\boldsymbol{h}]$ and $\hat{Var}[\boldsymbol{h}]$ are the sample mean and variance that is estimated on the current mini-batch. $\boldsymbol{x}_{<t>}$ is the vector input at the current time step *t* whereas $\boldsymbol{h}_{<t-1>}$ is the vector output of the cell state of the same RNN hidden layer from the previous time step. The function $f(.)$ depends upon the type of the cell in the RNN layer viz. LSTM, GRU, or Vanilla. A simple CB may be represented in the figure below:

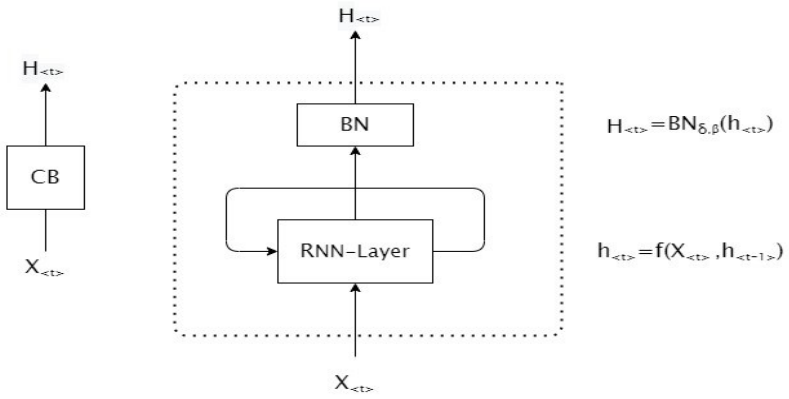


Figure 5-2: Composite Block(CB). The block comprises of an RNN layer followed by a Batch Normalization (BN) layer. At time step *t*, the input to CB block is $\boldsymbol{x_{<t>}}$ and the output is $\boldsymbol{H_{<t>}}$. The output of RNN layer ($\boldsymbol{h_{<t>}}$) is processed with a BN layer as per the statistics of the current mini batch.

The RNN layer in the CB block has several hidden RNN cells, we identify the number of hidden RNN cells in the RNN layer as an important hyperparameter that is needed to be tuned. We call this hyperparameter as *"nb_cells"*

Dense Connectivity of several CBs: Dense Block (DB)

Multiple CBs are stacked on top of each other such that the input of every CB is connected to the output of every other CB in a feed-forward fashion to form one Dense Block. The connection is defined by a concatenation operator. A densely connected CB can be shown in Figure 5-3:

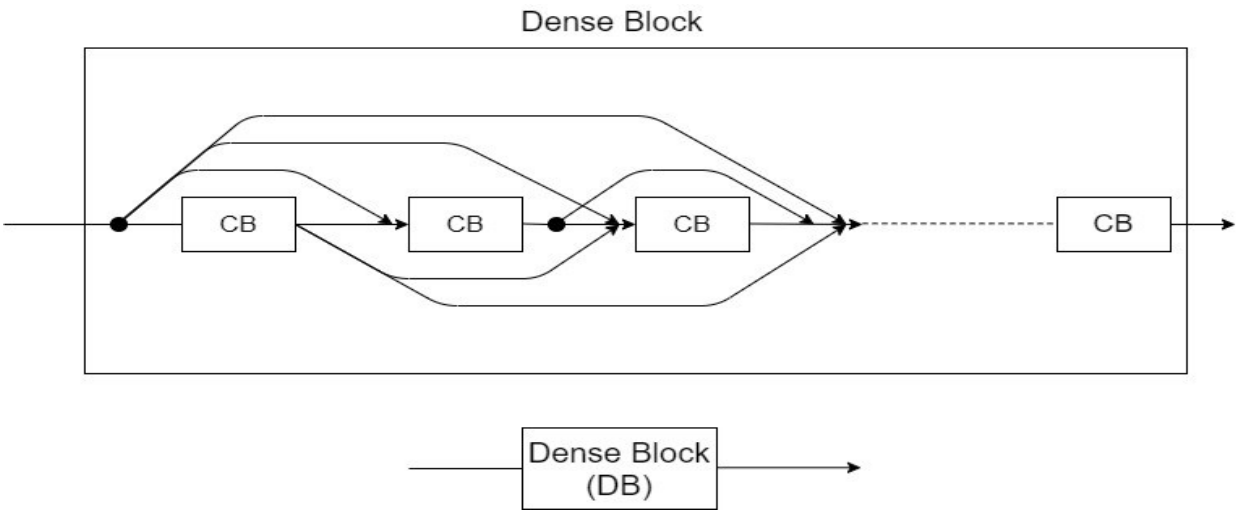


Figure 5-3: Dense Block(DB). Multiple composite blocks are stacked on top of each other and are connected in dense fashion.

The number of CBs in a dense block is identified to be an important hyperparameter and named as "*nb_CBs"*. It is not necessary that all the composite blocks have an equal number of RNN cells.

Dense Connectivity of several DBs: Our Proposed Architecture

Multiple DBs are stacked on top of each other such that the input of every DB is connected to the output of every other DB in a feed-forward fashion to form our proposed Densely connected RNN architecture for PD progression using RNA-Sequence data. An architecture with 3 Dense Blocks is shown in Figure 5-4 below:

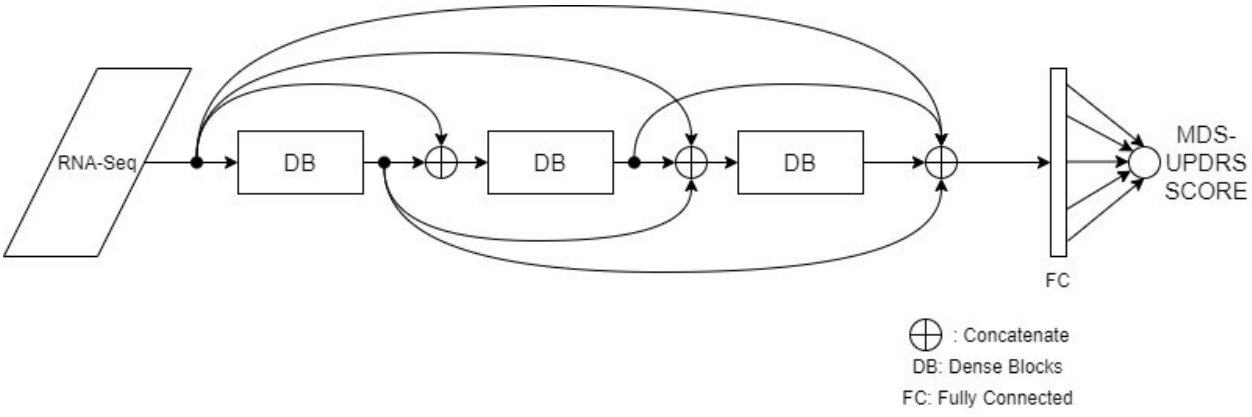


Figure 5-4 : Proposed Densely Connected RNN Architecture. Multiple Dense blocks are stacked on top of each other and are connected in dense fashion. The output of the last dense block flattened

and connected to a single neuron. Note that the number of dense blocks is an important hyper parameter and may be tuned.

The number of DBs is identified to be an important hyperparameter and named as "*nb_DBs"*. It is not necessary that all the dense blocks have an equal number of CBs. However, we tried a variety of experiments by keeping the *nb_CBs* both constant and varied across various dense blocks. The exact configuration i.e. the number of RNN cells in a composite block, the number of composite blocks, and the number of dense blocks of the architecture after the hyper-parameter tuning is described in detail in Section 7.1.

5.3 Training Pipeline and Hyperparameter Tuning

During the training and hyperparameter tuning phase of RNN models, a train-validation-test split was performed on the dataset to reserve 60% subjects as training, 20% subjects as validation, and the remaining 20% subjects as hold outset. The training and hyperparameter tuning of the RNN models were performed using the training and validation datasets and the performance of the tuned model was tested on the test dataset.

We used the validation dataset to tune the hyperparameters of our RNN models using a grid search strategy on the hyperparameter search space. We categorize the tuning experiments into three categories: *Structural*, *Optimization,* and *Regularisation*. The Structural category has the hyperparameters such as the type of RNN cell, number of RNN cells, number of Densely Connected Blocks(DBs), number of Composite Function Blocks (CBs). The Optimization category has the hyperparameters such as learning rate, loss optimizers, type of loss, number of epochs and batch sizes, etc. whereas the Regularisation category has the hyperparameters that help to combat overfittings such as selection between L1, L2 and L1_L2 regularization [43], the regularization constant, dropout layer [44] factors, etc. The table below summarizes the three categories of tuning experiments.

Table 5-1: Hyper Parameter Tuning Search Space

Purpose	**Hyper parameter**	**Grid/Sample Space**
Structural	Type of RNN Cell	[Vanilla RNN]
	Number of RNN cells(*nb_cells*)	[32,64,128,256,512]
	Number of Composite Function Blocks (*nb_CBs*)	[2,4,8,12,16,32]
	Number of Densely Connected Blocks (*nb_DBs*)	[1,2,4,8,16]
No of Experiments in Structural Category=		150
Optimization	Learning rate(lr)	lr ∈ [0.0001, 1] with 5 steps
	Loss	[Mean_square_error, Mean_absolute_error, Mean square_logarithmic_error]
	Loss Optimizer	[Adam[45], Nadam[46], RMSProp[47], Adadelta[48], Adagrad[49], Adamax[45], SGD]
	Number of epochs	[250]
No of Experiments in Optimization Category =		105
Regularisation	Dropout factor	[0,0.2,0.5]
	Type of Regularization	[L1, L2, L1_L2]
	Regularization Applied on	[Weights, Bias, Recurrent, Activity]
	Regularisation constant(λ)	$\lambda \in$ [0,0.1] with 5 steps
No of Experiments in Regularisation Category		50

While performing the hyperparameter tuning, we first tuned the RNN models for the Structural hyperparameters (No of CBs, DBs, Cells) keeping the hyperparameters of the remaining Optimization and regularisation initialized to default values as mentioned below in Table 5-2. we found the combination of Structural hyperparameters that gives the best performance, we then moved to tune the hyperparameters of the optimization category keeping the regularization aspects as constant and the structural hyperparameters initialized to the best combination. Lastly, the hyper parameters of regularization were tuned by initializing the hyperparameters of structure and optimization to the best combination. The pipeline for model training and hyperparameter tuning is detailed in the next Section 5.3.1.

In this way, the total number of experiments run as part of our study was the "addition" of the number of experiments in Structural, the number of experiments in Optimization and the number of experiments in Regularization category. From the

Table 5-1 above, the number of experiments run as part of the Structural category are the total combinations in the structural grid that is 150 experiments. Similarly, the number of experiments in Optimization and Regularization Categories are 105 and 50**,** respectively. As such, the total number of experiments totals to (150+105+50)=305 experiments.

Table 5-2: Default Initialization values of Hyperparameters

	Hyper parameter	**Default Values**
Optimization Category	Learning rate(lr)	Lr=0.001
	Loss	Mean_square_error
	Loss Optimizer	Adam
	Number of epochs	250
Regularisation Category	Dropout factor	0
	Type of Regularization	L2
	Regularization Applied on	Weights
	Regularisation constant(λ)	$\lambda = 0.01$

5.3.1 Algorithm : Model Training and Hyper Parameter Tuning

Algorithm 1: Model Training and Hyper Parameter Tuning Pipeline

Input: Preprocessed Input RNA-Sequence Data
Split: Randomnly Split subjects into training, validation and testing
Normalization: Perform feature normalization by fit from training onto validation dataset
Grid: Create the vectors for each category of hyperparameter tuning
$\theta_{structural} \leftarrow [type\_cell, nb\_cells, nb\_CBs, nb\_DBs]$
$\theta_{optimization} \leftarrow [lr, loss, optimizer, nb\_epochs]$
$\theta_{regularization} \leftarrow [dropout, type\_reg, reg\_ON, reg\_constant]$
$Set_{Structural} \leftarrow$ Set of all Structural hyperparamters
$Set_{Optimization} \leftarrow$ Set of all Optimization hyperparamters
$Set_{Regularization} \leftarrow$ Set of all Regularization hyperparamters

Defaults: Initialize the default hyperparameters
$\theta_{optimization,default} \leftarrow [0.001, MSE, Adam, 500]$
$\theta_{regularization,default} \leftarrow [0, L2, Weights, 0.01]$

`/* Experiments to find best structural hyperparameters start */`

foreach $\theta_{structural,i}$ in $Set_{Structural}$ **do**
  $\theta_{optimization,i} \leftarrow \theta_{optimization,default}$;
  $\theta_{regularization,i} \leftarrow \theta_{regularization,default}$;
  Create and Train Model ;
  $Cost_i^{val} \leftarrow$ Cost on validation dataset
end foreach
$\theta_{structural,best} = \theta_{argmin}(Cost_1^{val},,,,Cost_i^{val})$;

`/* Experiments to find best Optimization hyperparameters start */`

foreach $\theta_{optimization,j}$ in $Set_{Optimization}$ **do**
  $\theta_{structural,j} \leftarrow \theta_{structural,best}$;
  $\theta_{regularization,j} \leftarrow \theta_{regularization,default}$;
  Create and Train Model ;
  $Cost_j^{val} \leftarrow$ Cost on validation dataset
end foreach
$\theta_{optimization,best} = \theta_{argmin}(Cost_1^{val},,,,Cost_j^{val})$;

`/* Experiments to find best Regularization hyperparameters start */`

foreach $\theta_{regularization,k}$ in $Set_{Regularization}$ **do**
  $\theta_{structural,k} \leftarrow \theta_{structural,best}$;
  $\theta_{optimization,k} \leftarrow \theta_{optimization,best}$;
  Create and Train Model ;
  $Cost_k^{val} \leftarrow$ Cost on validation dataset
end foreach
$\theta_{regularization,best} = \theta_{argmin}(Cost_1^{val},,,,Cost_k^{val})$;

5.4 Hardware and Software Environment

5.4.1 Cloud Computing Resource for Model Training and Tuning:

Each experiment was assigned to a SLURM job that ran onto Compute Canada High-Performance Computing Cluster-Cedar. Each experiment ran for about 8 hours with the configuration as described in Table 5-3.

Table 5-3: Computing Configuration for Model Training and Hyper Parameter Tuning

No of CPU Cores	**GPU**	**RAM per CPU core**	**Time for a single Job**
6	16 GB P100	40GB	~8 Hours

GNU parallel program was employed to run multiple jobs/experiments in parallel. The whole set of experiments ran for a range of 15-25 days on the cluster due to the waiting time for the shared resource.

5.4.2 Software Environment:

We used Keras [50] with TensorFlow [51] Backend to create all Neural Network Models. Python machine learning libraries were used to perform the tasks including data preprocessing, plotting, etc.

Chapter 6 Baseline Models and Evaluation Metrics

In this chapter, we will discuss the baseline models that are used to compare the performance of our proposed model. Further, we will define the evaluation metrics that were used to evaluate our models.

6.1 Baseline Methods

6.1.1 Machine Learning Baseline Methods:

We compare the performance of our proposed DL model with the performance of classical machine learning techniques including Linear Regression (LR), Support Vector Machines (SVM), Decision Trees (DT), and Random Forest (RF). The above machine learning baseline models were used to compare the performance of RNN models for predictive modeling of Alzheimer's Disease(AD) progression by the authors of [16]. The problem statement studied by Wang et al [16] is similar to ours except that the study was performed on AD and the data was of clinical features(non-genetic/transcriptomic).

6.1.2 Training Strategy for Machine Learning based Baseline Methods:

The baseline models cannot handle varied lengths of longitudinal data. Hence, we use the commonly used training strategy for using aggregated features of all patients' historical visits to train these baseline models [52]. This technique was also used by the authors of [16] for AD progression to train their baseline models.

Consider an individual patient who has N number of visits then assuming Q number of features were collected for every visit, the patient's data can be represented in the form of a 3D vector as follows:

$$[1, N, X_{1xQ}] \tag{6.1}$$

The data presented in equation-(6.1) is a longitudinal temporal representation of a single patient's data. The same is used to train the Deep learning RNN models, however, to train the non-DL baseline models, this data is converted into an aggregated form. The aggregated data vector also has Q features, where each value of the q^{th} feature (q=1,2,3,4,……., Q) is aggregated over N visits depending upon the data type of the feature as summarized below in Table 6-1.

Table 6-1: Aggregated Value for Different Data Types

Data type of q^{th} feature	**Aggregated Value for q^{th} feature**
Continuous type feature	$q^{th}{}_{aggregated} = \frac{q_{<1>} + q_{<2>} + q_{<3>} + \ldots . q_{<N>}}{N}$
Ordinal type feature	$q^{th}{}_{aggregated} = Median(q_{<1>}, q_{<2>}, q_{<3>}, \ldots . q_{<N>})$
Nominal type feature	$q^{th}{}_{aggregated} = Mode(q_{<1>}, q_{<2>}, q_{<3>}, \ldots . q_{<N>})$

If the q^{th} feature is a continuous type variable, then the aggregated value of that feature is the mean of the value of that feature over the past N visits. If the q^{th} feature is an ordinal data type, then the aggregated value is the median of the values of past N visits. Similarly, if the q^{th} feature is a nominal variable then the aggregated value of that feature is the mode of the values of that feature for all historical N visits. The scheme of data aggregation over historical visits for a single patient can be depicted in Figure 6-1.

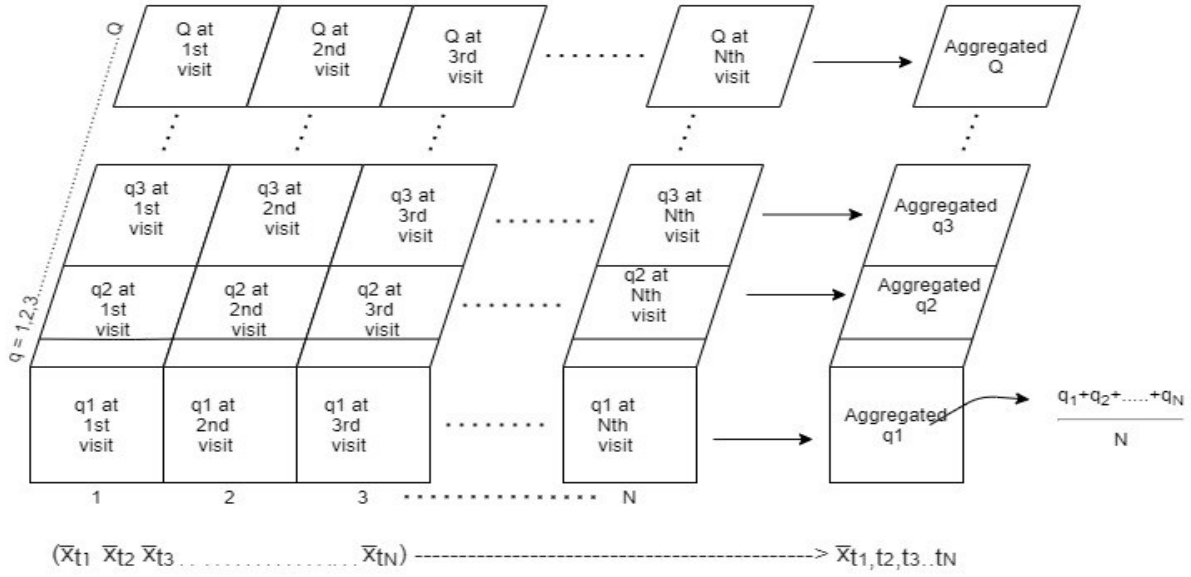


Figure 6-1: Data Aggregation for a single patient with N visits where at visit the RNA sequence features(q1 to Q) were collected.

Based on the aggregated values, the equation for the predictive model in equation-(5.1), can now be written as follows:

$$\hat{y}_{<n+1>} = f(\tilde{X}_{<1,2,\ldots,n>}) \tag{6.2}$$

where $\tilde{X}_{<1,2,\ldots,n>}$ is a feature vector with Q dimensions (q=1,2,3,……., Q), where each of the q^{th} dimension is an aggregated value of the q^{th} feature for all the historical N visits.

6.1.3 Experimental Setup for Machine Learning based Baseline Methods:

The PPMI dataset was preprocessed using the procedures elaborated in Chapter 4 that included steps such as data merging, missing value imputation, noise removal and feature selection followed by the process of data aggregation described in Section 6.1.2. 5-fold cross-validation on the complete dataset was performed to train, tune, and test on the baseline models. The results of the performance of the baseline methods are described in detail in Chapter 7.

We use the package Scikit-learn [53] to employ Linear Regression, Support Vector Machines, Decision Trees, and Random Forests. The LR fits a linear model with no regularisation and is an implementation of Ordinary Least Squares. The sklearn parameters for LR with default initialization (fit_intercept=True, normalize=False) is used. The fit_intercept is set to True so that the intercepts can be included in calculations while normalize is set to False as the data is already in normalized form.

The SVM regressor is implemented with a radial basis function kernel in epsilon-insensitive loss with a squared L2 penalty. The regularization parameter (C) is set to 1 and the parameter epsilon which controls the width of the insensitive zone is initialized to 0.2.

The DT Regressor is implemented with a mean square error criterion. There is no limit imposed on the maximum depth of the trees and the nodes are expanded until all leaves are pure. The minimum number of samples required to split an internal node is 2. The minimum number of samples required to be at a leaf node is 1.

The RF Regressor is implemented with the mean square error criterion. The number of trees in the forest is 100. The maximum depth of each tree is set to 3. The maximum number of features considered for splitting a node is equal to the total number of features in the dataset. The minimum number of samples required to be at a leaf node is 1.

6.1.4 Other RNN based Baseline Models:

RNN models have shown great potential in healthcare applications. We reviewed the literature to find various previous studies that employed RNNs. To the best of our knowledge, the study closest to our problem statement is Alzheimer's Disease(AD) Progression using RNN by [16]. They implemented a two-layered LSTM Neural Network for the purpose of AD progression and showed its potential for the use of predictive modeling in other chronic diseases. Motivated by [16], we also implemented their proposed architecture for the purpose of PD disease progression. These sets of experiments form RNN based baselines for us.

The RNN architecture proposed by authors of [16] is given below in Figure 6-2. It consists of two RNN layers (LSTM layers) stacked upon each other. The architecture showed a promising result

for the purpose of AD Progression with longitudinal variable visits patient data. We would like to see if such an architecture has potential in PD progression as well. We conducted experiments with various combinations of the number of hidden neurons, the number of hidden RNN layers, type of RNN cells to study their application in PD progression.

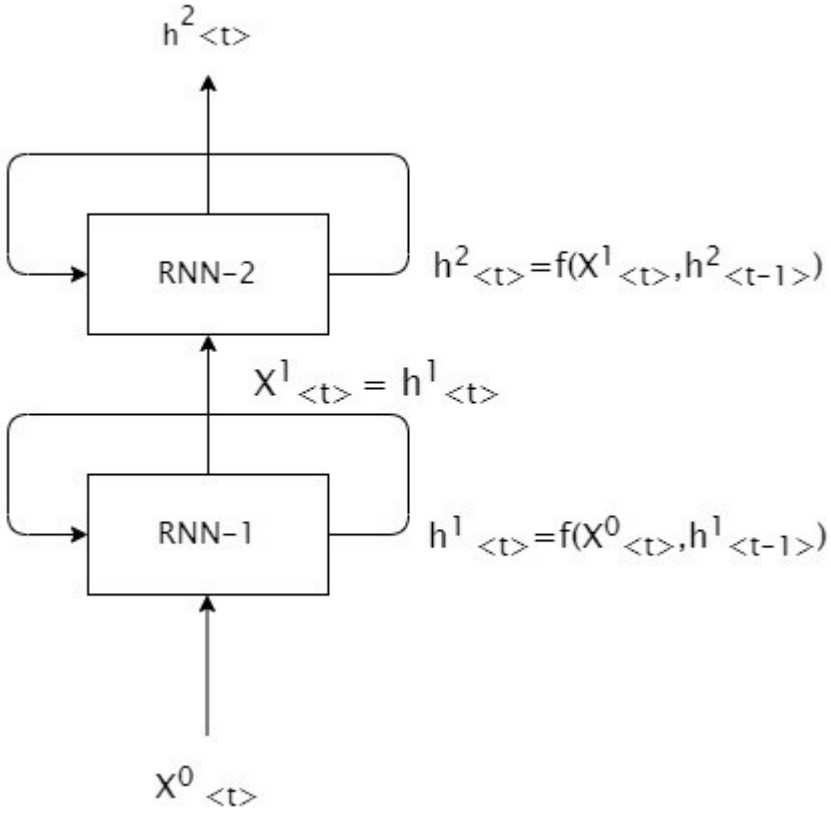


Figure 6-2: Two RNN layers stacked on each other

The various configurations of simple multi-layered RNN network that we trained are listed in Table 6-2.

Table 6-2: Other Baseline RNN Model Configurations that were trained

RNN Cell	**Model Architecture**	**Configurations**
LSTM	No of Hidden Layers:3 No of Hidden Neurons: 4000 in 1st and 2nd layer, 1000 in 3rd layer	Configuration:1 3 layered LSTM
		Configuration:2

		With Batch normalization layers
		Configuration:3 With Drop out Layers
		Configuration:4 With Multi-Task Learning
GRU	No of Hidden Layers:3 No of Hidden Neurons: 4000 in 1st and 2nd layer, 1000 in 3rd layer	Configuration:1 3 layered GRU
		Configuration:2 With Batch normalization layers
		Configuration:3 With Drop out Layers
		Configuration:4 With Multi-Task Learning
Vanilla RNN	No of Hidden Layers:3 No of Hidden Neurons: 4000 in 1st and 2nd layer, 1000 in 3rd layer	Configuration:1 3 layered Vanilla RNN
		Configuration:2 With Batch normalization layers
		Configuration:3 With Drop out Layers
		Configuration:4 With Multi-Task Learning

6.1.5 Experimental Setup for Other RNN based Baseline Models:

The PPMI dataset was preprocessed using the procedures elaborated in Chapter -4 Data Pre-Processing that included steps such as data merging, missing value imputation, noise removal, and feature selection. A train-validation split was performed on the training dataset to reserve 80%

subjects as training, 20% subjects as validation subjects for the purpose of model training, and hyperparameter tuning.

All the RNN models were created using the Keras framework with TensorFlow backend. The RNN model parameters were initialized with He_Normal [54] and the loss is mean square error with L2 regularization applied to penalize the weights with a regularization constant of $\lambda = 0.01$. The loss optimizer Adam is used with the default learning rate initialized at 0.001. A learning rate schedule is employed to reduce the learning rate by a factor of 1/5 for every 10 epochs if there is no improvement in the validation loss. The training happened in a mini batch fashion with a batch size of 16 subjects where each subject had the same number of time sequence/visits data. The total number of epochs in the case of each of RNN model created was 1000.

Each experiment was assigned to a SLURM job that ran onto Compute Canada High-Performance Computing Cluster-Cedar. Each experiment ran for about 5 hours with the following configuration.

Table 6-3: Computing Configuration for Other RNN based Baseline Methods

No of CPU Cores	**GPU**	**RAM per CPU core**	**Time for single Job**
6	32 GB V100	40 GB	~5 Hours

6.2 Evaluation Metrics

In this section, we define the evaluation metrics to assess the performance of our proposed predictive model and baseline models.

6.2.1 Prerequisites:

The target variable (MDS-UPDRS) to be predicted by the model is a continuous type variable. Hence the evaluation metrics for our analysis are based on the following two standard regression metrics:

1) Root Mean Square Error (**RMSE**): It measures the error of a model in predicting quantitative data. The difference between the ground truth value and the predicted value is called as residual/error [55]. RMSE is the square root of the mean of squares of all such residuals and can be mathematically defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{L} \left(y_i - \hat{y_i}\right)^2}{L}} \tag{6.3}$$

where y_i is the ground truth value of MDS-UPDRS, $\hat{y_i}$ is the predicted value of MDS-UPDRS by the model and L is the number of samples/patients.

2) Spearman's rank Correlation ($\boldsymbol{\rho}$): It measures the strength and direction (negative or positive) of a relationship between the predicted and ground truth variable. Mathematically it is defined as follows:

$$\rho(y_i, \hat{y_i}) = 1 - \frac{6\sum_{i=1}^{L} {d_i}^2}{L(L^2 - 1)} \tag{6.4}$$

where d_i is the difference in the rank of ith predicted and ground-truth value of MDS-UPDRS score and L is the number of samples/patients

Based on the above two regression metrics, let us propose the evaluation metrics for our study. We know that PD is a progressive disease and usually the disease status of the patient will either worsen by next year's visits or would remain the same. In rare cases, the disease status would become better, but we know this will not happen until an effective treatment is given to the patient which is currently not available. To a patient, he/she might be concerned with how accurate the predicted value of the future year's MDS-UPDRS score is. We, therefore, provide the following criterion to evaluate the performance of our predictive models.

6.2.2 Mathematical Definition of our Evaluation Metrics:

The aim is to evaluate the skill of the model in predicting the MDS-UPDRS score. We know that the model can be utilized such that when RNA-Seq of the 1st visit is given as input to the model, it predicts the 2nd year's disease status score. And when the first year's and second's RNA-Seq information is given as inputs, it predicts the 3rd year's visit disease status score. Likewise, to predict the 4th year's visit disease status, the model takes the past information of 1st, 2nd, and 3rd year and the 5th year prediction follows the same pattern. The RMSE produced by the model in predicting the MDS-UPDRS score for all the prediction time points is averaged to give the metric called Progression Identification Error (PIE). In addition, the rank order correlation between the actual and predicted MDS-UPDRS is averaged for all prediction time points and is defined as a Progression Identification Correlation (PIC). A mathematical definition of the metrics PIE and PIC that is based on the equations (6.3) and (6.4) can be provided as follows:

Recall the prediction model for our study from equation (5.1) that is $\hat{y}_{<N+1>} = f(X_{<1>}, X_{<2>}, X_{<3>}, \ldots X_{<N>})$. The maximum number of visits available is N=5. The first prediction point starts at n=2 and the last prediction point is n=6. Let p be the number of patients with 5 visits data availability.

Progression identification Error (PIE):

PIE= Average of RMSE in predicting MDS-UPDRS for various prediction time points

$$\textbf{PIE} = \frac{RMSE\ for\ 2^{nd} + RMSE\ for\ 3^{rd} + RMSE\ for\ 4^{th} + RMSE\ for\ 5^{th} + RMSE\ for\ 6^{th}\ time\ point}{5}$$

where, $RMSE\ for\ 2^{nd} time\ point = \sqrt{\frac{\sum_p `(y_{<2>} - f(X_{<1>}))^2}{p}}$,

$$RMSE\ for\ 3^{rd} time\ point = \sqrt{\frac{\sum_p `(y_{<3>} - f(X_{<1>}, X_{<2>}))^2}{p}},$$

$$RMSE\ for\ 4^{th} time\ point = \sqrt{\frac{\sum_p `(y_{<4>} - f(X_{<1>}, X_{<2>}, X_{<3>}))^2}{p}},$$

$$RMSE\ for\ 5^{th} time\ point = \sqrt{\frac{\sum_p `(y_{<5>} - f(X_{<1>}, X_{<2>}, X_{<3>}, X_{<4>}))^2}{p}},$$

$RMSE\ for\ 6^{th} time\ point = \sqrt{\frac{\sum_p `(y_{<6>} - f(X_{<1>}, X_{<2>}, X_{<3>}, X_{<4>}, X_{<5>}))^2}{p}}$ and

$y_{<2>}, y_{<3>}, y_{<4>}, y_{<5>}, y_{<6>}$ are ground truth values,

$f(.)$ represents the predicted values,

$X_{<1>}, X_{<2>}, X_{<3>}, X_{<4>}, X_{<5>}$ are the longitudinal temporal RNA-Seq data.

In general form, the definition of PIE can be written as follows:

$$PIE = \frac{1}{N}\sum_{k=1}^{N} `(RMSE_{<k+1>}) \tag{6.5}$$

where

$$RMSE_{<k+1>} = \sqrt{\frac{\sum_p `(y_{<k+1>} - f(X_{<1>}, X_{<2>}, , , , , X_{<k>}))^2}{p}} \tag{6.6}$$

here p is the number of patients in test data and N is the number of visits/time intervals of data for patients.

PIE is an unbounded continuous scale metric the same as RMSE and hence helps in relative performance comparison. In the simplest form, PIE is the RMSE averages over all the time sequence predictions of the model. The lesser the value of PIE, the lesser is the error in the average error between predicted and ground-truth values of MDS-UPDRS score.

Progression identification Correlation (PIC):

PIC= Average of spearman's rank order correlation (ρ) between predicted and ground truth values of MDS-UPDRS for prediction time points

$$\mathbf{PIC} = \frac{\rho\ for\ 2^{nd} + \rho\ for\ 3^{rd} + \rho\ for\ 4^{th} + \rho\ for\ 5^{th} + \rho\ for\ 6^{th}\ time\ point}{5}$$

where, $\rho\ for\ 2^{nd}\ timepoint = \rho\big(y_{<2>}, f(X_{<1>})\big)$,

$\rho\ for\ 3^{rd} timepoint = \rho\big(y_{<3>}, f(X_{<1>}, X_{<2>})\big)$,

$\rho\ for\ 4^{th} timepoint = \rho\big(y_{<4>}, f(X_{<1>}, X_{<2>}, X_{<3>})\big)$,

$\rho\ for\ 5^{th} timepoint = \rho\big(y_{<5>}, f(X_{<1>}, X_{<2>}, X_{<3>}, X_{<4>})\big)$,

$\rho\ for\ 6^{th}\ time\ point = \rho(y_{<6>}, f(X_{<1>}, X_{<2>}, X_{<3>}, X_{<4>}, X_{<5>}))$ and

$y_{<2>}, y_{<3>}, y_{<4>}, y_{<5>}, y_{<6>}$ are ground truth values,

$f(.)$ represents the predicted values,

$X_{<1>,}X_{<2>}, X_{<3>}, X_{<4>}, X_{<5>}$ are the longitudinal temporal RNA-Seq data.

In general form, the definition of PIC can be written as follows:

$$PIC = \frac{1}{N}\sum_{k=1}^{N} `(Rho_{<k+1>}) \tag{6.7}$$

where $$Rho_{<k+1>} = \rho(y_{<k+1>}, f(X_{<1>}, X_{<2>}, , , , , X_{<k>})) \tag{6.8}$$

here p is the number of patients in test data and N is the number of visits/time intervals of data for patients.

PIC is also accompanied by the p-value to comment if the relationship is statistically significant. As part of model evaluation, we use both PIE and PIC to evaluate the performance of our proposed model in predicting the MDS-UPDRS score. PIE is an unbounded continuous scale metric the same as RMSE and hence helps in relative performance comparison. The lesser the value of PIE, the lesser is the error in the average error between predicted and ground-truth values of MDS-UPDRS score. On the other hand, PIC is a continuous metric but bounded between -1 and +1. A PIC of -1 indicates a perfect negative correlation and a PIC of +1 indicates a perfect positive correlation between predicted and ground-truth values of the MDS-UPDRS score. A PIC of 0 indicates there is no correlation between the predicted and ground-truth values. A small PIE with PIC close to +1 is desired in our case.

Note: For the machine learning in baseline models, the prediction model equation is given by equation (6.2), that is different from that of our proposed model equation (5.1) for RNNs, hence the metrics defined for the baseline models will use aggregated values of historical information instead of longitudinal data values.

Table 6-4 below summarises the criterion for model performance evaluation for both RNN-based models and machine learning-based models:

Table 6-4: Evaluation Metrics

Metrics	
	Progression identification Error $(PIE) = \frac{1}{N}\sum_{k=1}^{N}{}^{`}(RMSE_{<k+1>})$ where, $RMSE_{<k+1>} = \sqrt{\frac{\sum_p {}^{`}(y_{<k+1>} - f(X_{<1>}, X_{<2>}, ,,,, X_{<k>}))^2}{p}}$ for RNN models $RMSE_{<k+1>} = \sqrt{\frac{\sum_p {}^{`}(y_{<k+1>} - f(\tilde{X}_{<1,2,.....k>}))^2}{p}}$ for other baseline models
	Progression identification Correlation $(PIC) = \frac{1}{N}\sum_{k=1}^{N}{}^{`}(Rho_{<k+1>})$ where, $Rho_{<k+1>} = \rho\left(y_{<k+1>}, f(X_{<1>}, X_{<2>}, , , , , X_{<k>})\right)$ for RNN models $Rho_{<k+1>} = \rho\left(y_{<k+1>}, f\left(\tilde{X}_{<1,2,.....k>}\right)\right)$ for other baseline models

Metrics to Evaluate Temporal Capability:

This section discuss the metrics to evaluate if the model can capture the patterns from the historical visits i.e. whether giving multiple historical visits data to the model has an effect on the model's skill to accurately predict MDS-UPDRS score. Let us understand this evaluation scheme with an example, consider a patient with a total number of visits be N, then for predicting MDS-UPDRS for the $(N+1)^{th}$ time point, the model may be provided with only most recent visit's data i.e. at N^{th} time point. In this case the data at $(N-1)^{th}$, $(N-2)^{th}$,…,2^{nd},1^{st} visit is not given to the model and the

RMSE say $RMSE_{<N>}$ for such a prediction is noted. In the second case, to predict the same $(N+1)^{th}$ disease status, we now provide data from the past two visits at N^{th} and $(N-1)^{th}$ and the $RMSE_{<N,N-1>}$ is noted. In the third case, to predict the disease status at $(N+1)^{th}$, we provide the data from last three visits i.e. at N^{th}, $(N-1)^{th}$ and $(N-2)^{th}$ and the $RMSE_{<N,N-1,N-2>}$ is noted. Similarly, we note $RMSE_{<N,N-1,N-2,\ N-3>}$ and proceed to find further RMSEs until we reach $RMSE_{<N,N-1,N-2,\ N-3,\ldots.1>}$ that is calculated by the model that is given the data from all the historical visits. If the model succeeds in learning the patterns from data of historical visits, then we will have the given scenario:

$$RMSE_{<N>} > RMSE_{<N,N-1>} > RMSE_{<N,N-1,N-2>} > RMSE_{<N,N-1,N-2,\ N-3>} > \ldots >$$

$$RMSE_{<N,N-1,N-2,\ldots\ldots\ldots N-k+1>} > \ldots > RMSE_{<N,N-1,N-2,\ N-3,\ldots.1>} \quad (6.9)$$

where,

$$RMSE_{<N,N-1,\ldots N-k+1>} = \sqrt{\frac{\sum_p `(y_{<N+1>} - f(X_{<N>}, X_{<N-1>}, \ldots\ldots\ldots X_{<N-k>}))^2}{p}} \quad (6.10)$$

for k=1,2,3,…,N in RNN models

$$RMSE_{<N,N-1,\ldots N-k+1>} = \sqrt{\frac{\sum_p `(y_{<N+1>} - f(\tilde{X}_{<N,N-1,\ldots\ldots N-k>}))^2}{p}} \quad (6.11)$$

for k=1,2,3,…,N in baseline models

6.2.3 Statistical Significance Test for Model Comparison:

We will perform 5-fold Cross validation (CV) in our experiments to do the comparison in performance between the proposed model and the baseline models. The metrics obtained will be reported in the from of mean and standard errors. Further, we will test if the improvements in the performance of our proposed model are statistically significant by employing the paired Student's t-test at the significance level of $\alpha = 0.05$ on the difference of means.

To Test Progression identification Error (RMSE):

Null Hypothesis(H_0): The PIE of proposed RNN model is greater or equal than the PIE of the Baseline model

$$\boldsymbol{H_0}: PIE(Proposed\ Model) \geq PIE(Baseline\ Method)$$

Alternate Hypothesis(H_1): The PIE of proposed RNN model is lesser than the PIE of the Baseline model

$$\boldsymbol{H_a}: PIE(Proposed\ Model) < PIE(Baseline\ Method)$$

Criteria:

If p<0.05 using the one-tailed t-test then we have sufficient evidence to reject the null hypothesis and infer that the error between the actual and predicted values of the MDS-UPDRS score produced by our predictive model is lesser than the error between actual and predicted values produced by the baseline method.

To Test Progression Identification Correlation (Correlation):

Null Hypothesis(H_0): The PIC of the proposed RNN model is lesser or equal to the PIC of the Baseline model

$$\boldsymbol{H_0}: PIC(Proposed\ Model) \leq PIC(Baseline\ Method)$$

Alternate Hypothesis(H_1): The PIC of proposed RNN model is greater than the PIC of the Baseline model

$$\boldsymbol{H_a}: PIC(Proposed\ Model) > PIC(Baseline\ Method)$$

Criteria:

If p<0.05 using the one-tailed t-test then we have sufficient evidence to reject the null hypothesis and infer that the correlation between actual and predicted values of the MDS-UPDRS score produced by our predictive model is greater than the correlation between actual and predicted values produced by the baseline methods.

Chapter 7 Experimental Results and Discussion

In this chapter, we present a summary of the various experiments that were conducted as part of our study and the relevant results and discussion.

7.1 Results Overview

The PPMI dataset was preprocessed using the procedures elaborated in Chapter 4 that included steps such as data merging, missing value imputation, noise removal and feature selection. A train-validation split was performed on the training dataset to reserve 80% subjects as training, 20% subjects as validation subjects for the purpose of model training, and hyperparameter tuning. The experiments were carried out to implement our proposed densely connected RNN architecture as described in Section 5.2.1 using the algorithm for model training and hyperparameter tuning as described in 5.3.1.

The experiments ran on Compute Canada HPC for a range of 15-25 days on the cluster. A learning rate schedule was employed to reduce the learning rate by a factor of 1/5 for every 10 epochs if there is no improvement in the validation loss. The training happened in a mini-batch fashion with a batch size of 16 subjects where each subject had the same number of time sequence/visits data. The model checkpoints with the best validation loss were saved at the end of every experiment. After spending close to 2000 GPU hours, the combination of best Structural, Optimization, and Regularization hyperparameters obtained for our proposed model is presented in the table below: The architecture is named "*Dense-Vanilla*"

Table 7-1: "Dense-Vanilla" Net Architecture and Optimized Hyper Parameters

Structural Parameters	**Optimization Hyper Parameters**	**Regularization Hyper Parameters**
Type of RNN Cell**:** Vanilla RNN Number of RNN cells: 256 Number of CBs : 4 Number of DBs : 3	Lr: 0.001 Loss: MSE Optimizer: Nadam Number of epochs=250	Dropout: 0 Regularization: L2 on weights with constant 0.025

A 5-fold cross-validation on the dataset was performed and the model Dense-Vanilla achieved an RMSE of (mean=6.01, standard deviation= 0.41) in predicting the MDS-UPDRS score and showed a rank order Correlation of (mean=0.83, standard deviation=0.02, $p<0.0001$) between the predicted MDS-UPDRS and the true MDS-UPDRS.

The following gives an outline on the various sections of this chapter:

1) The model "*Dense-Vanilla"* was employed to predict the MDS-UPDRS scores for the test subjects. The predicted and ground truth progression curves for test patients of one of the folds of 5-CV are presented in Section 7.2. We present the progression curves for the test patients that had 5 visits of data available. The progression curves are presented in two figures, with 12 test patients in each figure.
2) We compare the performance of our proposed Dense-Vanilla with the performance of classical baseline methods viz. Linear Regression, SVM, Decision Trees, and Random Forests. Furthermore, a statistical significance test was performed using a paired Student's t-test to test the significance of improvements. The results are presented in Section 7.3.
3) Other RNN-based baseline models: The set of experiments ran to investigate how two-layer or three-layer RNN models would perform on the PD Progression problem. The results are presented in Section 7.4.
4) We observed that dense connections played an important role and the model fails to learn without the dense connections. We present a comparison in performance between the RNN models that were densely connected and the RNN models with plain configurations i.e. with no dense connection. The results are presented in Section 7.5.
5) We observed Batch Normalization in the Composite Block plays an important role, without which it would be too difficult to make the model learn. We present a comparison in performance between the densely connected RNN models that have Batch Normalization in the composite block and the densely connected RNN models with no Batch Normalization in the composite block. The results are presented in Section 7.6.

7.2 Prediction of Disease Progression in PD Patients using our Predictive Model:

The model "Dense-Vanilla" was employed to predict the rate of disease progression for the subjects in the test set for a period of 4 years. The model predicts the disease status variable (MDS-UDRS score) of a patient for the immediate future hospital visit by considering the RNA sequence data of historical visits. As per the 5-CV model evaluation, we know that the predictive model has an RMSE of 6.01 in predicting the MDS-UPDRS score and a rank correlation of (PIC=0.83, p-value=0.0001).To give a visual representation of the results, the ground truth values of MDS-UPDRS score are plotted to compare how close the predicted MDS-UPDRS scores are to the true values of MDS-UPDRS. The disease progression curves are presented in Figure 7-1 and Figure 7-2, with 12 test patients in each figure.

The problem is modeled in a way that when a patient's Baseline year (1st year) RNA-seq data is given to the predictive model that predicts the next year's MDS-UPDRS score of that patient. When baseline year and 2nd year's RNA-seq data is given to the model, it predicts the 3rd year's MDS-UPDRS score and likewise, when Baseline year's, 2nd year's and 3rd year's data is given as input, then the model predicts the 4th year's MDS-UPDRS score. Thus, as we move forward in time sequences, the model leverages the temporal patterns in the historical RNA sequence data and tries to predict the immediate future time disease status.

In addition, to analyze the impact of various feature categories on the model performance, we conducted experiments with and without non-transcriptomic feature categories (Motor Assessment, Subject Characteristics, General Exam, Age, Time Interval) and the results are:

Table 7-2: Results of the model with and without non-transcriptomic input features

Feature Categories	**Evaluation Metrics**	
	PIE(RMSE)	**PIC(Correlation)**
RNA-Seq with non-transcriptomic features	6.01±0.185	0.83±0.01
RNA-Seq without non-transcriptomic features	6.87±0.265	0.71±0.03

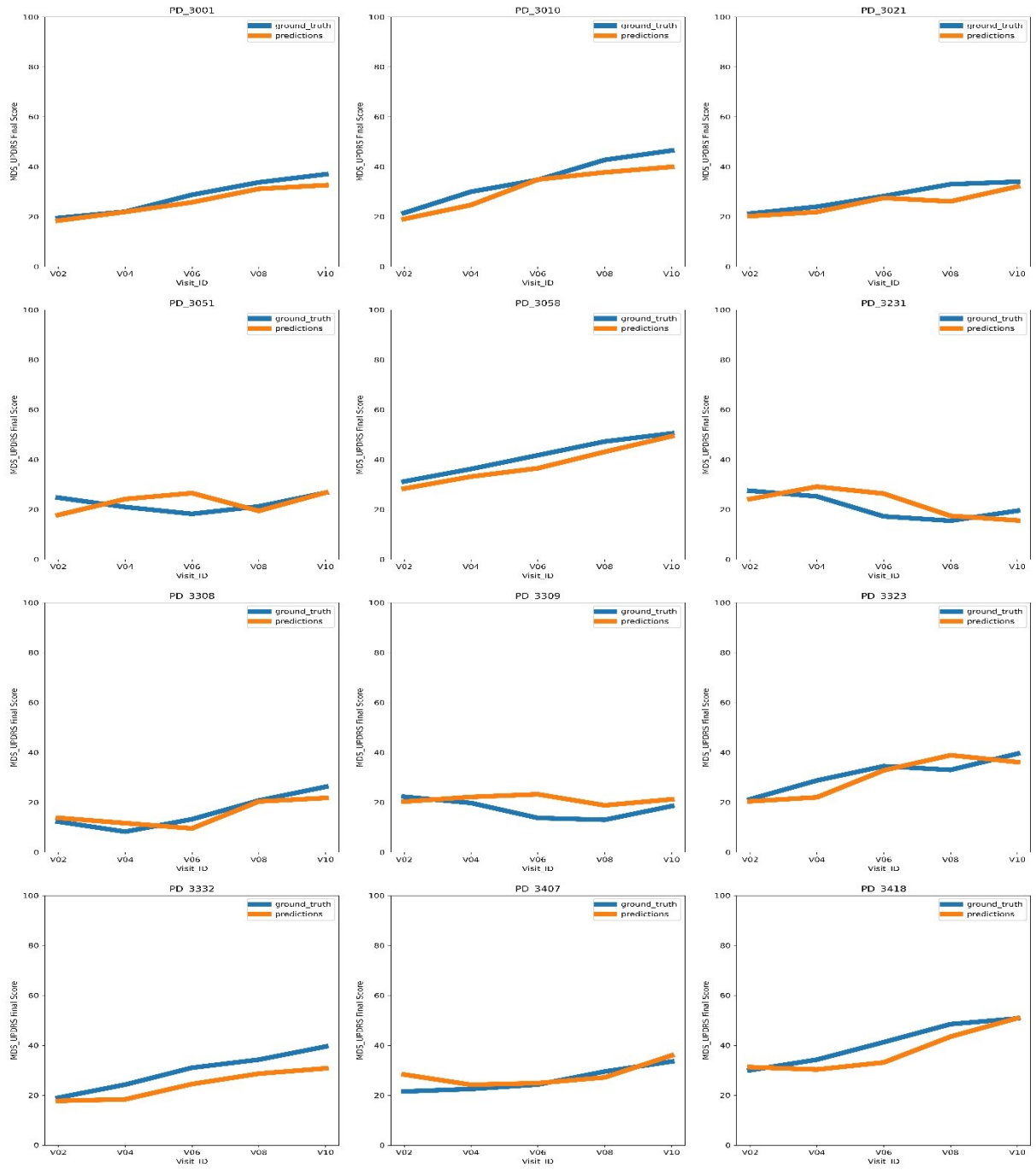


Figure 7-1: Predicted and Ground Truth disease progression curves for Test Patients. Each subplot belongs to one test patient. The curve in blue is the ground truth curve and the curve in orange is the one predicted by our Predictive Model. The x axis has discrete visit points(6months, 12 months, 24 months, 36 months, 48 months) and the y-axis has the MDS-UPDRS score for the patient.

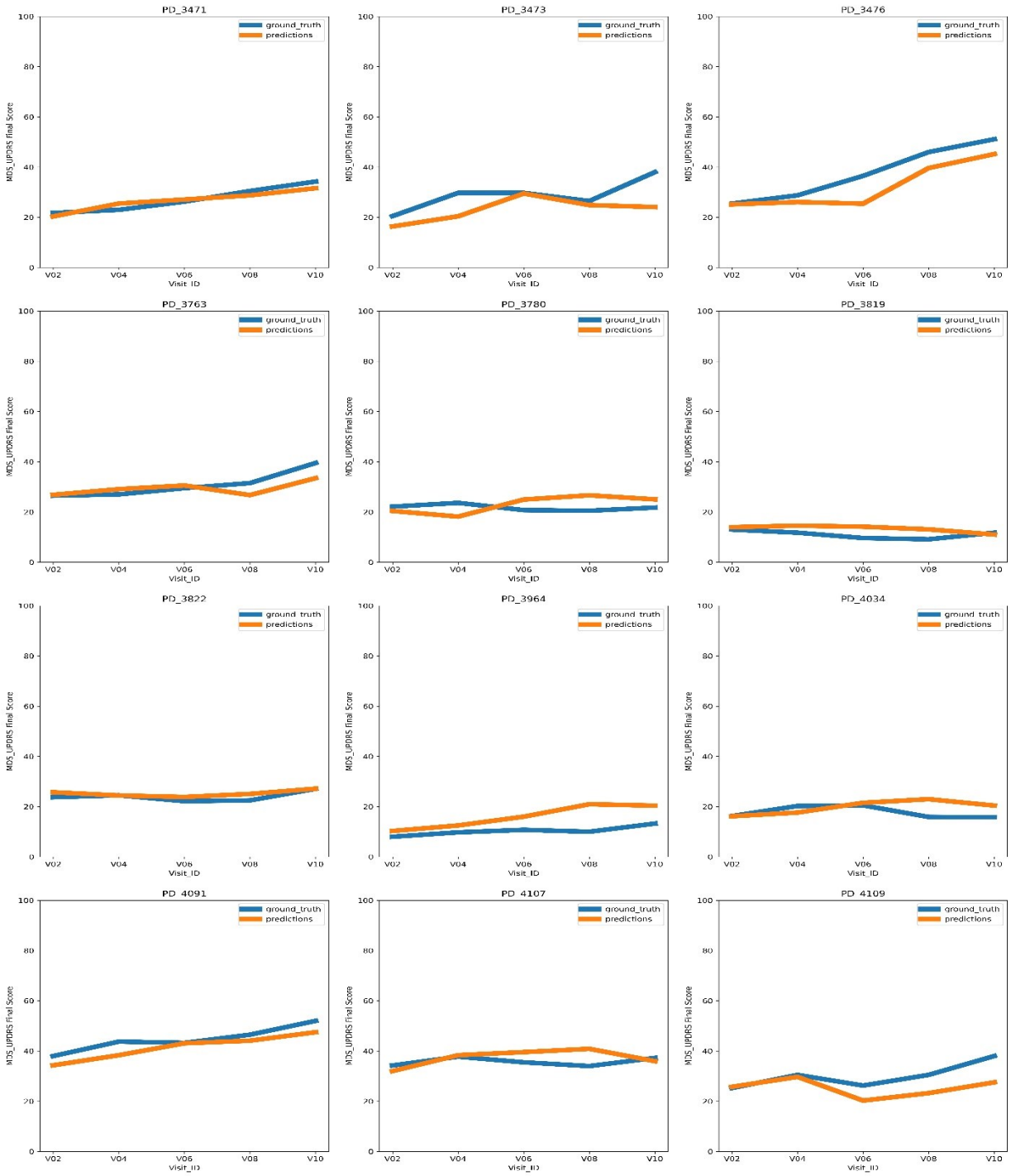


Figure 7-2: Predicted and Ground Truth disease progression curves for Remaining Test Patients

7.2.1 Discussion:

As illustrated in Figure 7-1 and Figure 7-2, our model is able to capture the trend and predict the MDS-UPDRS scores for a period of 4 years at discrete visit points of V02 (6 months), V04 (12 months), V06 (24 months), V08 (36 months), and V10 (48 months). For the sake of simplicity, we presented the disease progression graphs for the patients in the test dataset with 5 number of visits.

For almost all the test patients, the disease progression predicted by our model aligns with the ground truth disease progression values. This confirms that our proposed model was able to learn the medical patterns and granular features from the longitudinal data of the transcriptome of the patient. As per the 5-CV model evaluation, we know that the predictive model has an RMSE of 6.01 in predicting the MDS-UPDRS score and a rank correlation of (PIC=0.83, p-value=0.0001).We further evaluated the model with the temporal evaluation condition discussed in equation (6.9) and the model satisfies the condition confirming that the model benefitted from data of multiple historical visits.

As indicated in Table 7-2, we wanted to know the impact of adding non-transcriptomic input features such as Motor Assessment, Subject Characteristics, General Exam, Age, Time Interval to the RNA-Seq features while training the proposed model. The performance of the proposed model when trained on RNA-Seq input features, with and without non-transcriptomic data does not differ much.

It will be interesting to discuss the acceptability of RMSE of 6.01 in real-world applications. While MDS-UPDRS is a highly valid and reliable instrument to capture the true phenotype[56] [57], it is susceptible to “noise” [58]. The two major sources of noise are measurement error (inter and intra-rater variability) and short-term effects (mood, stress, climate, time of the day etc.) that are irrelevant to the overall progression of the disease[58]. It is reasonable to state that if the RMSE of 6.01 in predicting the MDS-UPDRS is lesser than or equal to the noise inherent in estimating the true MDS-UPDRS score then the predictive model has an acceptable error for real-world applications.

To this end, as per the study[58], the authors calculated the error variance because of noise in estimating the true value of MDS-UPDRS score and reported an error variance of 4.87 in a estimating score of part-I, 3.66 in part-II, and 15.52 in Part-III of MDS-UPDRS. The total noise

component of the MDS-UPDRS score(part-I, part-II, and part-III) is thus 8.059. As the RMSE of 6.01 in predicting the total MDS-UPDRS score by our predictive model is lesser than the inherent noise of 8.059 in the instrument itself, it is safe to assume that the performance of the predictive model is acceptable for real-world applications.

Moreover, as per the recent study[14] in predictive modeling of PD using imaging genetics on a combination of DNA genotyping and neuroimaging, the authors reported an RMSE of 7.82 in predicting MDS-UPDRS-Score. This value of RMSE is comparable to the RMSE of our predictive model that uses transcriptomic data.

7.3 Performance Comparison Between our Proposed Model and Baseline Methods

We compare the performance of our proposed model *Dense-Vanilla* with the classical machine learning baseline methods viz. Linear Regression (LR), Support Vector Machine (SVM), Decision Trees (DT) and Random Forest (RF). A 5-Fold CV was performed, and the metrics are reported in mean and standard errors of 5-CV in Table 7-3.

Table 7-3: Performance Comparison Between our Proposed Model and Baseline Methods

Models	**Evaluation Metrics**	
	PIE(RMSE)	**PIC(Correlation)**
Our Model (Dense-Vanilla)	6.01±0.185	0.83±0.01
Linear Regression	12.9± 0.279	0.38±0.02
SVM	9.87± 0.214	0.37±0.05
Decision Trees	8.39± 0.469	0.68±0.03
Random Forest	8.12±0.286	0.723±0.024

The values in Table 7-3 may be visualised by Figure 7-3 and Figure 7-4.

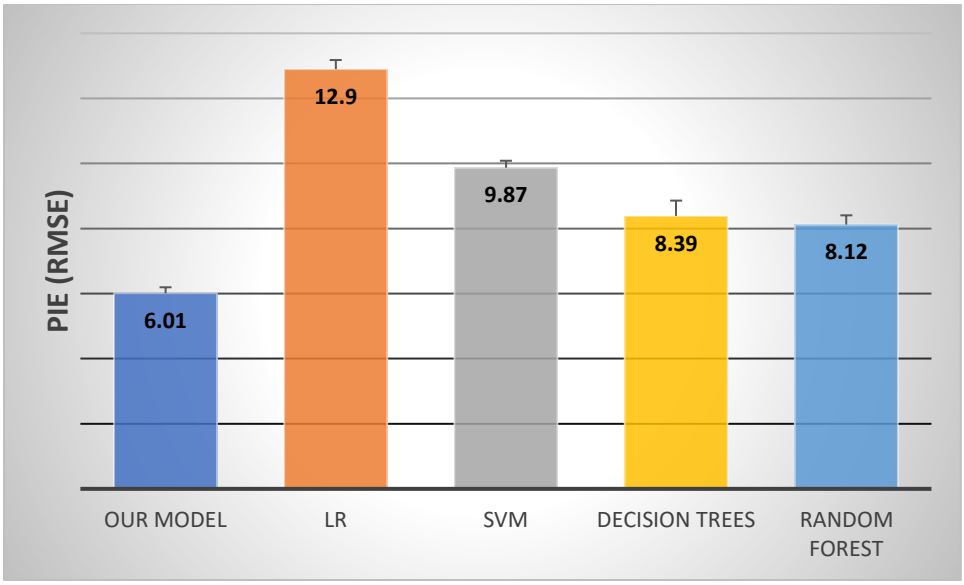


Figure 7-3: Comparison of RMSE produced by our proposed model and the baseline methods in predicting MDS-UPDRS score. The lesser the value of RMSE the better is the model in performance. All the values are mean and standard errors of 5-Fold CV.

As illustrated in Figure 7-3, there is a significant difference in the RMSE obtained by our proposed model as compared to the classical machine learning methods of Linear Regression, SVM, Decision Trees, and Random Forests. Thus, our model outperforms all the baseline methods in terms of RMSE produced in predicting the MDS-UPDRS score.

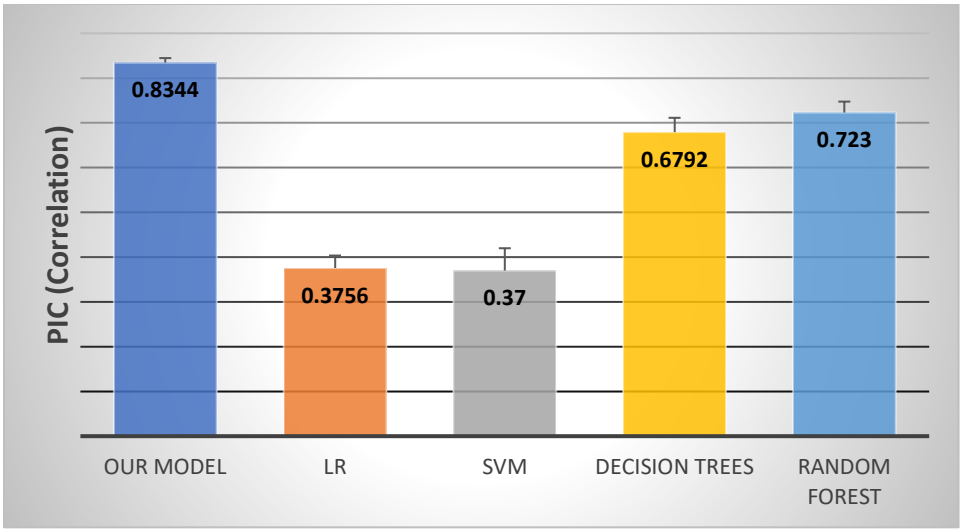


Figure 7-4: Comparison in rank correlation produced by our proposed model and the baseline methods in predicting MDS-UPDRS score. The greater the value of correlation the better is the model. All the values are mean and standard errors of 5-Fold CV.

As illustrated in Figure 7-4, there is a significant difference in the correlation obtained by our proposed model as compared to the classical machine learning methods of Linear Regression, SVM, Decision Trees, and Random Forests. Thus, our model outperforms all the baseline methods.

To test if the improvement in the performance of our proposed model was statistically significant as compared to the comparison methods, the metrics obtained from 5-Fold CV were given to the student's t-test with a significance level of alpha=0.05, degrees of freedom=4, as described in Section 6.2.3. The results are described in Table 7-4.

Table 7-4: Statistical Significance Test for Model Comparison

Models	**Hypothesis Testing**	
	t-Test on PIE(RMSE)	**t-Test on PIC(Correlation)**
Our Model and Linear Regression	t-stat= -20.58, p=0.000	t-stat=-18.9, p=0
Our Model and SVM	t-stat= -13.65, p=0.0001	t-stat=-9.07, p=0.0004
Our Model and Decision Trees	t-stat= -4.72, p=0.0046	t-stat=-4.5,p=0.0054
Our Model and Random Forests	t-stat= -6.19, p=0.0018	t-stat=-4.11,p=0.0074

As illustrated in Table 7-4, the improvements in the performance of our model as compared to the performance of baseline methods are statistically significant at a significance level of 0.05.

7.4 Results from Other RNN based Baseline Models

The architecture with two RNN layers stacked upon each other showed a promising result for the purpose of Alzheimer's Disease Progression in the study conducted by [16]. We investigated if such an architecture as described in Section 6.1.4 has potential in PD progression as well. We conducted experiments with various combinations of the number of hidden neurons, the number of hidden RNN layers, type of RNN cells to study their application in PD progression. The results of the baseline RNN models on a validation set are presented in Table 7-5.

Table 7-5: Results from Other RNN based Baseline Models

RNN Cell	**Model Architecture**	**Configurations**	**Evaluation Metrics**	
			PIE(RMSE)	PIC(Correlation)
LSTM	No of Hidden Layers:3 No of Hidden Neurons: 4000 in 1st and 2nd layer, 1000 in 3rd layer	Configuration:1 Simple 3 layered LSTM	10.24	0.22
		Configuration:2 With Batch normalization layers	10.18	0.32
		Configuration:3 With Drop out Layers	9.68	0.35
		Configuration:4 With Multi-Task Learning	12.04	0.25
GRU	No of Hidden Layers:3 No of Hidden Neurons: 4000 in 1st and 2nd layer, 1000 in 3rd layer	Configuration:1 Simple 3 layered GRU	10.24	0.259
		Configuration:2 With Batch normalization layers	10.17	0.40

		Configuration:3 With Drop out Layers	9.97	0.28
		Configuration:4 With Multi-Task Learning	12.16	0.25
Vanilla RNN	No of Hidden Layers:3 No of Hidden Neurons: 4000 in 1st and 2nd layer, 1000 in 3rd layer	Configuration:1 Simple 3 layered Vanilla RNN	11.13	0.20
		Configuration:2 With Batch normalization layers	8.12	0.73
		Configuration:3 With Drop out Layers	10.34	0.23
		Configuration:4 With Multi-Task Learning	11.81	0.35

7.5 Performance Comparison Between Densely Connected RNN Models and Plain RNN Configurations

The RNN model with the best performance is *Dense-Vanilla* that has a vanilla-RNN in its composite block with the number of composite blocks being 4 and the number of Dense Blocks being 3 as described in Table 7-1. We would like to see the performance of the architecture by changing the RNN cell to GRU and LSTM, keeping all the other hyperparameters the same.

Furthermore, we compare the performance of densely connected RNN models with that of models with plain configurations. Here plain configuration means that no dense connections were introduced either between Composite Blocks or between Dense Blocks. All the other hyperparameters (structural, optimization, and regularization) were kept the same to provide a fairground for comparison. A 5-Fold CV was performed, and the metrics are reported in mean and

standard errors of 5-CV. Table 7-6 summarises the performance of different configurations of neural network models.

Table 7-6: Performance Comparison Between Densely Connected RNN Models and Plain RNN Configuration

RNN Cell Type	**Model Configuration**	**Evaluation Metrics**	
		PIE(RMSE)	**PIC(Correlation)**
Vanilla RNN	With Dense Connections	6.01± 0.185	0.83±0.01
	Without Dense Connections	11.428±0.525	0.06±0.038
GRU	With Dense Connections	6.22±0.344	0.81±0.016
	Without Dense Connections	9.894±0.784	0.59±0.033
LSTM	With Dense Connections	6.22±0.124	0.84±0.010
	Without Dense Connections	13.251±0.603	0.53±0.032

The values in the table may be visualized by Figure 7-5 and Figure 7-6. We split the figures into two to accommodate both the evaluation metrics viz. RMSE and Rank order Correlation.

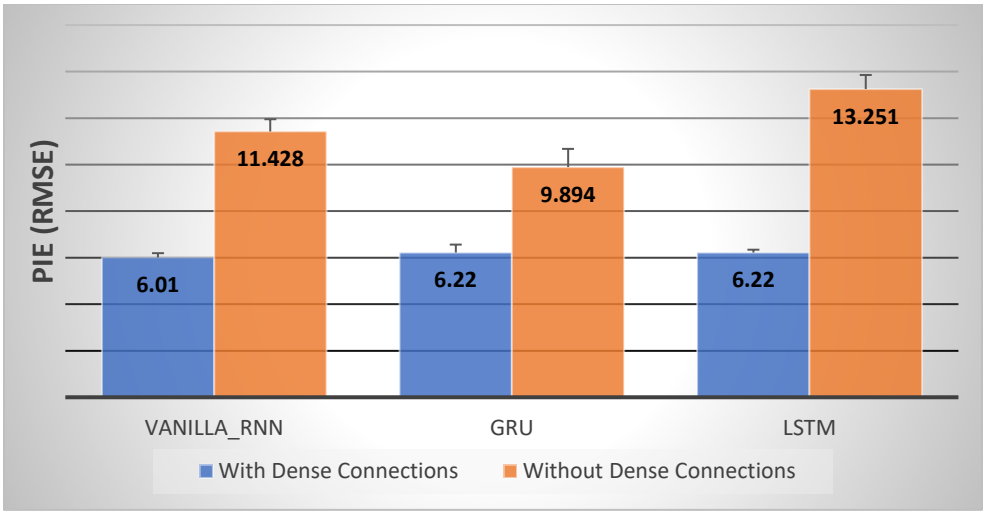


Figure 7-5: Comparison of RMSE produced by Densely Connected RNN Models and Plain RNN Configurations in predicting MDS-UPDRS score. The lesser the value of RMSE the better is the model. All the values are mean and standard errors of 5-Fold CV.

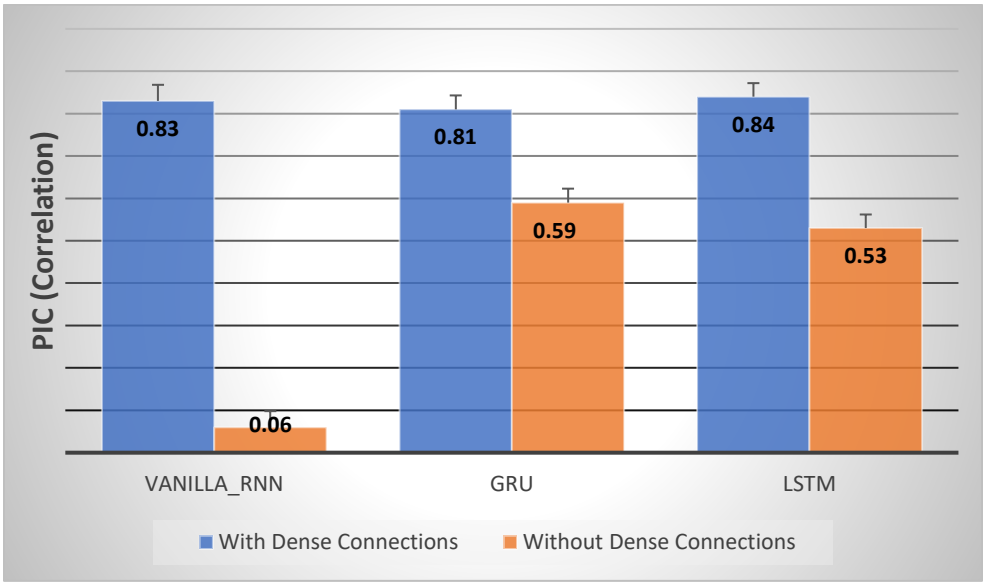


Figure 7-6: Comparison of Correlation scores obtained by Densely Connected RNN Models and Plain RNN Configurations in predicting MDS-UPDRS score. The greater the value of Correlation the better is the model. All the values are mean and standard errors of 5-Fold CV.

7.5.1 Discussion:

As illustrated in Figure 7-5, there is a significant difference in the RMSE obtained by models with Dense Connections and models without Dense Connections. We observe the behaviour for all three

categories of RNN cell types (Vanilla RNN, GRU, and LSTM). In each of the categories, the model with dense connections has lower RMSE in predicting the MDS-UPDRS as compared to the model without dense connections. Furthermore, as seen in Figure 7-6, a higher rank order correlation in predicting disease status is obtained when the models had dense connections as otherwise. Thus, the addition of dense connections between layers of RNN played an important role in our proposed architecture.

Also, we observe that the RMSE and Correlation scores obtained in each case of densely connected RNN models are similar, irrespective of the RNN cell type viz. Vanilla RNN, GRU, or LSTM. Generally, GRU and LSTM have a superior performance if the number of time steps/visits are large as they better handle the problem of vanishing gradient better than a Vanilla RNN. However, in our case, the maximum number of time-steps/visits available for a subject is only 5-time steps, and thus there is no vanishing gradient problem while backpropagating through time. Therefore, we observe a similar performance from a GRU, LSTM, or a Vanilla RNN cell in our proposed architecture.

It is worth mentioning that we choose to have Vanilla RNN as our final cell type in our proposed architecture because of the lesser number of model parameters.

7.6 Performance Comparison Between Densely Connected RNN Models with Batch Normalization and Densely Connected RNN Models without Batch Normalization Configurations

We observe that the proposed composite function block CB as described in Section 5.2.1, plays an important role in helping the densely connected RNN models to learn. The Batch Normalization layer on top of the RNN layer in a CB block is integral to the successful learning of our models. We conducted experiments where the normalization layer in the CB block was removed and the remaining model configurations remained the same. The table below shows the performance comparison between the models that are identical to each other in all aspects except the batch normalization layer in the CB block.

Table 7-7: Performance Comparison Between Densely Connected RNN Models with Batch Normalization and Densely Connected RNN Models without Batch Normalization:

Densely Connected RNN Models	**Model Configuration**	**Evaluation Metrics**	
		PIE(RMSE)	**PIC(Correlation)**
Vanilla RNN-Dense Connections	With Batch Normalization in CB	6.01 ±0.185	0.83±0.01
	Without Batch Normalization in CB	18.315±5.323	0.40±0.086
GRU-Dense Connections	With Batch Normalization in CB	6.22±0.344	0.81±0.016
	Without Batch Normalization in CB	11.491±1.801	0.40±0.096
LSTM-Dense Connections	With Batch Normalization in CB	6.22±0.124	0.84±0.010
	Without Batch Normalization in CB	11.253±1.755	0.38±0.099

The values in the table may be visualized by Figure 7-7 and Figure 7-8. We split the figures into two to accommodate both the evaluation metrics viz. RMSE and Rank order Correlation.

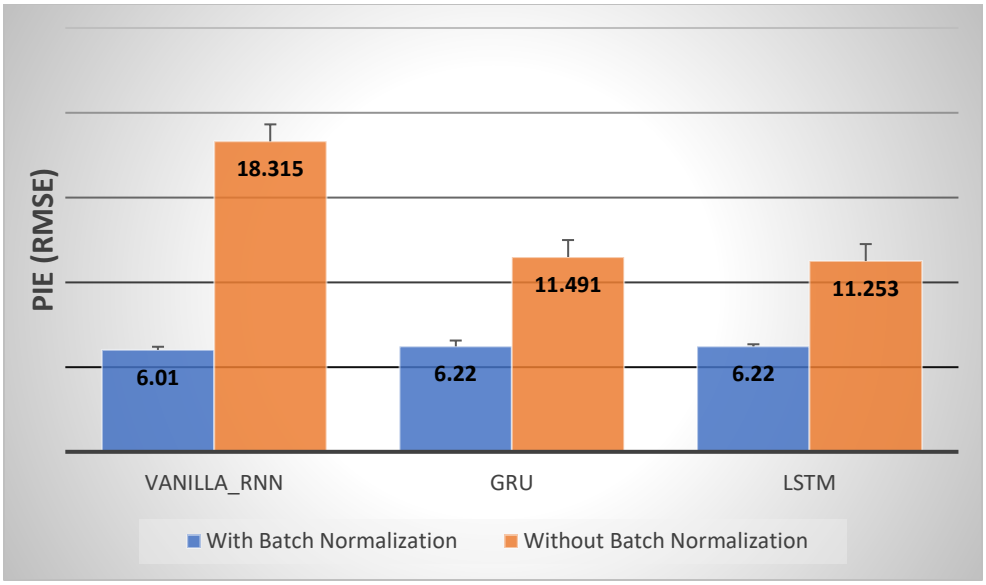


Figure 7-7: Comparison of RMSE produced by Densely Connected RNN Models with and without Batch Normalization layers. The lesser the value of RMSE the better is the model in performance. All the values are mean and standard errors of 5-Fold CV

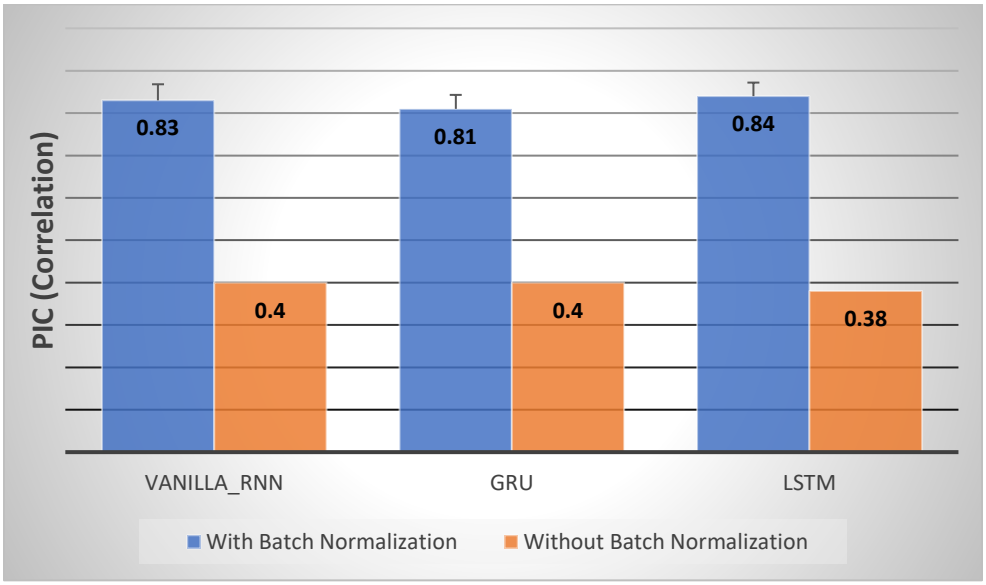


Figure 7-8: Comparison of Correlation scores obtained by Densely Connected RNN Models with and without Batch Normalization layers. The greater the value of correlation score, the better is the model in performance. All the values are mean and standard errors of 5-Fold CV

7.6.1 Discussion:

As illustrated in Figure 7-7, there is a significant difference in the RMSE obtained by models with Batch Normalization and the models without Batch Normalization. We observe the behaviour for all three categories of RNN cell types (Vanilla RNN, GRU, and LSTM). In each of the category, the model with batch normalization has lower RMSE in predicting the MDS-UPDRS as compared to the model without batch normalization. Furthermore, as seen in Figure 7-8, a higher rank order correlation in predicting disease status is obtained when the models have batch normalization. Thus, the Composite Block (CB) that is composed of an RNN Layer followed by a batch normalization layer is a component that helped in the successful learning of our predictive model.

Chapter 8 Conclusion and Future Work

8.1 Conclusion

To the best of our knowledge, this study is the first to investigate the usefulness of Omics data such as RNA-Seq in the predictive modeling of the PD Disease Progression using Deep-Learning. The contribution of this study can be summarized as follows:

1) We proposed a deep RNN structure that can predict the future year's MDS-UPDRS score of a patient by taking the inputs of previous year's RNA-Sequence data. The model can leverage the temporal patterns in the historical RNA sequence data.
2) The Predictive model is adaptive over time as the model was trained with irregular visit time intervals and the various number of visits. The model is able to predict the MDS-UPDRS score with an RMSE of 6.01 ad rank correlation of 0.83 between the predicted and true values of MDS-UPDRS
3) We observed that the introduction of Batch Normalization and Dense Connections play an important role in making the multi-layered RNN to learn the features from high dimensional gene expression data.

8.2 Future Work

1) In the near future, we aim to develop a multi-step predictive model. A model that takes the RNA-Sequence input of the only 1st visit and then predicts the disease progression of up to 5 years.
2) It is worth mentioning that transcriptome of the patient is an important progression biomarker, however, there is an unmet need to study the impact of other Omics data in the progression modeling. Therefore, we aim to extend our study to include genomics, proteomics, and metabolomics data of PD patients for the purpose of disease progression.
3) We aim to investigate the applicability of our proposed RNN architecture on other chronic disease progressions as such as Alzheimer's disease.

References

[1] F. Benetti, S. Gustincich, and G. Legname, "Gene expression profiling and therapeutic interventions in neurodegenerative diseases: a comprehensive study on potentiality and limits," *Expert Opin. Drug Discov.*, vol. 7, no. 3, pp. 245–259, 2012.

[2] A. S. Chen-Plotkin *et al.*, "Finding useful biomarkers for Parkinson's disease," *Sci. Transl. Med.*, vol. 10, no. 454, 2018.

[3] M. D. S. T. F. on R. S. for P. Disease, "The unified Parkinson's disease rating scale (UPDRS): status and recommendations," *Mov. Disord.*, vol. 18, no. 7, pp. 738–750, 2003.

[4] H. Kaur, A. K. Malhi, and H. S. Pannu, "Machine learning ensemble for neurological disorders," *Neural Comput. Appl.*, pp. 1–18, 2020.

[5] I. El Maachi, G.-A. Bilodeau, and W. Bouachir, "Deep 1D-Convnet for accurate Parkinson disease detection and severity prediction from gait," *Expert Syst. Appl.*, vol. 143, p. 113075, 2020.

[6] M. Diaz, M. A. Ferrer, D. Impedovo, G. Pirlo, and G. Vessio, "Dynamically enhanced static handwriting representation for Parkinson's disease detection," *Pattern Recognit. Lett.*, vol. 128, pp. 204–210, 2019.

[7] R. Prashanth, S. D. Roy, P. K. Mandal, and S. Ghosh, "High-accuracy detection of early Parkinson's disease through multimodal features and machine learning," *Int. J. Med. Inform.*, vol. 90, pp. 13–21, 2016.

[8] I. Kollia, A.-G. Stafylopatis, and S. Kollias, "Predicting Parkinson's disease using latent information extracted from deep neural networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[9] C. Xu, J.-G. Zhang, D. Lin, L. Zhang, H. Shen, and H.-W. Deng, "A systemic analysis of transcriptomic and epigenomic data to reveal regulation patterns for complex disease," *G3 Genes, Genomes, Genet.*, vol. 7, no. 7, pp. 2271–2279, 2017.

[10] Y. Hasin, M. Seldin, and A. Lusis, "Multi-omics approaches to disease," *Genome Biol.*, vol.

18, no. 1, pp. 1–15, 2017.

[11] E. Courtney, S. Kornfeld, K. Janitz, and M. Janitz, “Transcriptome profiling in neurodegenerative disease,” *J. Neurosci. Methods*, vol. 193, no. 2, pp. 189–202, 2010.

[12] N. Kumudini *et al.*, “Comparative analysis of four disease prediction models of Parkinson’s disease,” *Mol. Cell. Biochem.*, vol. 411, no. 1–2, pp. 127–134, 2016.

[13] G. S. Babu and S. Suresh, “Parkinson’s disease prediction using gene expression–A projection based learning meta-cognitive neural classifier approach,” *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1519–1529, 2013.

[14] M. Kim, J. Kim, S.-H. Lee, and H. Park, “Imaging genetics approach to Parkinson’s disease and its correlation with clinical score,” *Sci. Rep.*, vol. 7, p. 46700, 2017.

[15] X. Zhang *et al.*, “Data-driven subtyping of Parkinson’s disease using longitudinal clinical records: a cohort study,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, 2019.

[16] T. Wang, R. G. Qiu, and M. Yu, “Predictive modeling of the progression of Alzheimer’s disease with recurrent neural networks,” *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, 2018.

[17] “Parkinson’s Progression Markers Initiative,” 2018. https://www.ppmi-info.org/access-data-specimens/download-data/.

[18] R. G. Qiu, J. L. Qiu, and Y. Badr, “Predictive modeling of the severity/progression of Alzheimer’s diseases,” in *2017 International Conference on Grey Systems and Intelligent Services (GSIS)*, 2017, pp. 400–403.

[19] R. Lemos, J. Marôco, M. R. Simões, B. Santiago, J. Tomás, and I. Santana, “The free and cued selective reminding test for predicting progression to Alzheimer’s disease in patients with mild cognitive impairment: a prospective longitudinal study,” *J. Neuropsychol.*, vol. 11, no. 1, pp. 40–55, 2017.

[20] J. Yang, J. McAuley, J. Leskovec, P. LePendu, and N. Shah, “Finding progression stages in time-evolving event sequences,” in *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 783–794.

[21] M. Raff, B. Alberts, J. Lewis, A. Johnson, and K. Roberts, *Molecular Biology of the Cell*

4th edition. New York: National Center for Biotechnology InformationÕs Bookshelf, 2002.

[22] "What Are the Organ Systems of the Human Body?," 2016. https://study.com/academy/lesson/what-are-the-organ-systems-of-the-human-body.html.

[23] M. Nirenberg, "Historical review: Deciphering the genetic code–a personal account," *Trends Biochem. Sci.*, vol. 29, no. 1, pp. 46–54, 2004.

[24] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics," *Nat. Rev. Genet.*, vol. 10, no. 1, pp. 57–63, 2009.

[25] F. Ozsolak and P. M. Milos, "RNA sequencing: advances, challenges and opportunities," *Nat. Rev. Genet.*, vol. 12, no. 2, pp. 87–98, 2011.

[26] J. Stammer, "StatQuest: A gentle introduction to RNA-seq," 2017, [Online]. Available: https://statquest.org/video-index/.

[27] F. Godin, J. Dambre, and W. De Neve, "Improving language modeling using densely connected recurrent neural networks," *arXiv Prepr. arXiv1707.06130*, 2017.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv Prepr. arXiv1502.03167*, 2015.

[29] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," *arXiv Prepr. arXiv1603.09025*, 2016.

[30] H. Margarit and R. Subramaniam, "A batch-normalized recurrent network for sentiment classification," *Adv. Neural Inf. Process. Syst.*, pp. 2–8, 2016.

[31] S. K. Holden, T. Finseth, S. H. Sillau, and B. D. Berman, "Progression of MDS-UPDRS scores over five years in de novo Parkinson disease from the Parkinson's progression markers initiative cohort," *Mov. Disord. Clin. Pract.*, vol. 5, no. 1, pp. 47–53, 2018.

[32] J. Brownlee, "moving-average-smoothing-for-time-series-forecasting-python," 2016. https://machinelearningmastery.com/moving-average-smoothing-for-time-series-forecasting-python/.

[33] T. R. Golub *et al.*, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science (80-.).*, vol. 286, no. 5439, pp. 531–537, 1999.

[34] W. Li and Y. Yang, “How many genes are needed for a discriminant microarray data analysis,” in *Methods of microarray data analysis*, Springer, 2002, pp. 137–149.

[35] M. Xiong, X. Fang, and J. Zhao, “Biomarker identification by feature wrappers,” *Genome Res.*, vol. 11, no. 11, pp. 1878–1887, 2001.

[36] C. Ding and H. Peng, “Minimum redundancy feature selection from microarray gene expression data,” *J. Bioinform. Comput. Biol.*, vol. 3, no. 02, pp. 185–205, 2005.

[37] N. De Jay, S. Papillon-Cavanagh, C. Olsen, N. El-Hachem, G. Bontempi, and B. Haibe-Kains, “mRMRe: an R package for parallelized mRMR ensemble feature selection,” *Bioinformatics*, vol. 29, no. 18, pp. 2365–2368, 2013.

[38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[39] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[40] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway networks,” *arXiv Prepr. arXiv1505.00387*, 2015.

[41] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training very deep networks,” in *Advances in neural information processing systems*, 2015, pp. 2377–2385.

[42] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9908 LNCS, pp. 630–645, 2016, doi: 10.1007/978-3-319-46493-0_38.

[43] S. Merity, N. S. Keskar, and R. Socher, “Regularizing and optimizing LSTM language models,” *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, 2018.

[44] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[45] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv Prepr. arXiv1412.6980*, 2014.

[46] T. Dozat, “Incorporating nesterov momentum into adam,” 2016.

[47] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA Neural networks Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[48] M. D. Zeiler, “Adadelta: an adaptive learning rate method,” *arXiv Prepr. arXiv1212.5701*, 2012.

[49] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization.,” *J. Mach. Learn. Res.*, vol. 12, no. 7, 2011.

[50] F. and others Chollet, “Keras.” GitHub, 2015, [Online]. Available: https://github.com/fchollet/keras.

[51] M. Abadi *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[52] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, “Using recurrent neural network models for early detection of heart failure onset,” *J. Am. Med. Informatics Assoc.*, vol. 24, no. 2, pp. 361–370, 2017.

[53] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[54] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[55] J. F. Kenney, *Mathematics of statistics*, 3rd ed. D. Van Nostrand, 1939.

[56] C. G. Goetz *et al.*, “Movement Disorder Society-sponsored revision of the Unified Parkinson’s Disease Rating Scale (MDS-UPDRS): scale presentation and clinimetric testing results,” *Mov. Disord. Off. J. Mov. Disord. Soc.*, vol. 23, no. 15, pp. 2129–2170,

2008.

[57] P. Martinez-Martin *et al.*, “Expanded and independent validation of the Movement Disorder Society–Unified Parkinson’s disease rating scale (MDS-UPDRS),” *J. Neurol.*, vol. 260, no. 1, pp. 228–236, 2013.

[58] L. J. W. Evers, J. H. Krijthe, M. J. Meinders, B. R. Bloem, and T. M. Heskes, “Measuring Parkinson’s disease over time: The real-world within-subject reliability of the MDS-UPDRS,” *Mov. Disord.*, vol. 34, no. 10, pp. 1480–1487, 2019.