Spring 5-2020

Predicting Disease Progression Using Deep Recurrent Neural Networks and Longitudinal Electronic Health Record Data

Seunghwan Kim
Washington University in St. Louis

Washington University in St. Louis

McKelvey School of Engineering

Department of Computer Science and Engineering

Thesis Examination Committee:

Philip R.O. Payne, Advisor

Chenyang Lu

Yixin Chen

Predicting Disease Progression Using Deep Recurrent Neural Networks and Longitudinal Electronic Health Record Data

By

Seunghwan Kim

A thesis presented to the McKelvey School of Engineering of Washington University in St. Louis in partial fulfillment of the requirements for the degree of Master of Science

May 2020

St. Louis, Missouri

Dedication

This thesis is dedicated to my parents, Kyung-Ho Kim and Seong-Hee Yoon, for their unconditional love and support.

Acknowledgements

I would like to thank my thesis advisor Dr. Philip R.O. Payne of the Institute for Informatics at Washington University School of Medicine. Professor Payne was always supportive of my research direction and provided me with valuable advice, and constantly steered me in the right direction throughout the research.

I would also like to thank the professors who were willing to serve as committee members for this research study: Dr. Chenyang Lu and Dr. Yixin Chen. Without them, the examination survey could not have been successfully conducted.

I would also like to thank clinical experts at the Department of Neurology and my colleagues from Payne lab at Washington University School of Medicine for their valuable discussions and guidance in various research studies.

Finally, I wish to express my immense gratitude to my family and to my friends for their support and encouragement throughout my years of study and in writing this thesis.

Seunghwan Kim

Table of Contents

List of Tables iii

List of Figures iv

Abstract v

Chapter 1: Introduction 1

Background 1

Outline of Thesis 3

Chapter 2: Disease Progression Management 4

Research Objective 5

Chapter 3: Recurrent Neural Networks 6

Long Short-Term Memory 7

Gated Recurrent Unit 9

Chapter 4: Dataset 11

Data Collection 11

Data Preprocessing 12

Data Imputation 13

Sequence Generation 14

Chapter 5: Implementation 17

Deep Recurrent Neural Networks Architecture 17

Model Hyperparameters 18

Chapter 6: Methods 19

Experiment Design 19

Results and Performances 20

Chapter 7: Conclusion 26

Discussion 26

References 28

List of Tables

Table 1 DRNN Performance on NF1 Clinical Registry Data 21
Table 2 DRNN Performance on EHR-merged NF1 Clinical Registry Data 24
Table 3 DRNN Performance on NF1 Clinical Registry Data 25
Table 4 DRNN Performance on EHR-merged NF1 Clinical Registry Data 25

List of Figures

Figure 1. Diagram of the RNN architecture..... 6
Figure 2. Diagram of the LSTM unit architecture 7
Figure 3. Diagram of GRU unit architecture 9
Figure 4. The input format to the RNN..... 14
Figure 5. Simple diagram of 2-layer DRNN architecture with LSTM and GRU..... 17
Figure 6. AUC of DRNN output at time step 36, trained on sequences up to age of 9. 22
Figure 7. F1 Score of DRNN output at time step 36, trained on sequences up to age of 9. 22
Figure 8. Recall of DRNN output at time step 36, trained on sequences up to age of 9. 23
Figure 9. Precision of DRNN output at time step 36, trained on sequences up to age of 9..... 23

Abstract

Predicting Disease Progression Using Deep Recurrent Neural Networks and Longitudinal Electronic Health Record Data

By

Seunghwan Kim

Master of Science in Computer Science

Washington University in St. Louis, 2020

Research Advisor: Professor Philip R.O. Payne

Electronic Health Records (EHR) are widely adopted and used throughout healthcare systems and are able to collect and store longitudinal information data that can be used to describe patient phenotypes. From the underlying data structures used in the EHR, discrete data can be extracted and analyzed to improve patient care and outcomes via tasks such as risk stratification and prospective disease management. Temporality in EHR is innately present given the nature of these data, however, and traditional classification models are limited in this context by the cross-sectional nature of training and prediction processes. Finding temporal patterns in EHR is especially important as it encodes temporal concepts such as event trends, episodes, cycles, and abnormalities. Previously, there have been attempts to utilize temporal neural network models to predict clinical intervention time and mortality in the intensive care unit (ICU) and recurrent neural network (RNN) models to predict multiple types of medical conditions as well as medication use. However, such work has been limited in scope and generalizability beyond the immediate use cases that have been focused upon. In order to extend the relevant knowledge-base, this study demonstrates a predictive modeling pipeline that can extract and integrate

clinical information from the EHR, construct a feature set, and apply a deep recurrent neural network (DRNN) to model complex time stamped longitudinal data for monitoring and managing the progression of a disease condition. It utilizes longitudinal data of pediatric patient cohort diagnosed with Neurofibromatosis Type 1 (NF1), which is one of the most common neurogenetic disorders and occurs in 1 of every 3,000 births, without predilection for race, sex, or ethnicity. The prediction pipeline is differentiable from other efforts to-date that have sought to model NF1 progression in that it involves the analysis of multi-dimensional phenotypes wherein the DRNN is able to model complex non-linear relationships between event points in the longitudinal data both temporally and also within the cross-sectional observation. Such an approach is critical when seeking to transition from traditional evidence-based care models to precision medicine paradigms. Furthermore, our predictive modeling pipeline can be generalized and applied to manage the progression and stratify the risks in other similar complex diseases, as it can predict multiple set of sub-phenotypical features from training on longitudinal event sequences.

Chapter 1: Introduction

Electronic Health Records (EHR) are widely adopted and used throughout healthcare systems and are able to collect and store longitudinal information data that can be used to describe patient phenotypes. From the underlying data structures used in the EHR, discrete data can be extracted and analyzed to improve patient care and outcomes via tasks such as risk stratification and prospective disease management. Previously, there have been various attempts to utilize machine learning models to predict clinical conditions, monitor risk progression, and identify phenotype complexes (Wei et al., 2017; Miled et al., 2020; Dodek et al., 1998; Goldstein et al., 2017). This chapter gives a brief summary of the background of temporality in EHR, clinical predictions using EHR, and the clinical background of Neurofibromatosis Type 1 as a use case.

Background

EHR was developed in 1960s, but it was not until the early 2000s when EHR was widely adopted throughout the healthcare systems. Since EHR collects longitudinal information data of patients throughout the course of visits, temporality in EHR is innately present given the nature of these data. Temporality can be present across different scales of time depending on the clinical importance of observation, from frequently measured lab tests to visits to annual visits for a broader progression monitoring.

Before the development of EHR, clinicians managed the progression of diseases through evidence-based decision from empirical experience. However, many diseases have complex multi-dimensional phenotypes and clinicians often lack a standard model for risk management because of that clinical variability. Furthermore, incorporating temporal trends across multiple dimensions of clinical features was a challenge in traditional medicine.

Recently, statistical models have been employed for discovering meaningful insights from the complex EHR data (Miled et al., 2020; Dodek et al., 1998; Goldstein et al., 2017). Traditional supervised learning approaches were applied to the patient data in EHR, and they were successful in disease classification and phenotype comorbidity analyses based on the direct observation of discrete set feature vectors (Wei et al., 2017). While they were able to learn from high dimensional feature vectors, often the temporal dimension was ignored and multiple feature vectors collected over timesteps for a sample were merged to form a discrete feature vector that reflects the current observation, thereby limiting for the model to learning only from a cross-sectional observation of each sample.

However, finding the temporal patterns is important for progression management in a disease with a diverse clinical variability, since the rate of progression for each complex multi-phenotype may vary tremendously. More recent works have attempted to address the temporality by introducing artificial neural networks such as temporal convolutional networks (Cartling et al., 2020) and variants of recurrent neural networks (Laksana et al., 2019; Wang et al., 2019). In these models, information at each time step is propagated through time in the learning process, and the prediction model learns from the entire course of patient's past states to predict a future state.

Among the diseases that have complex multi-dimensional phenotypes, Neurofibromatosis Type 1 exhibits extreme clinical variability. NF1 is one of the most common neurogenetic disorders and occurs in 1 of every 3,000 births, without predilection for across race, sex, or ethnicity (Friedman, 2002; Friedman, 1999). The clinical variability exists not only in the presence of multiple sub-phenotype combinations but also in the progression of such phenotypes throughout the pediatric ages. Therefore, it is not possible to have one standardized treatment

plan that can fit the entire pediatric NF1 patient cohort, which calls for the use of temporal prediction models.

Outline of Thesis

Chapter 2 explains the importance of disease progression management modeling, describes statistical approaches that are currently used in research, and present our research objective. Chapter 3 explains the mechanism of RNN and two gating mechanisms in RNN, which are used in this study's neural network architecture. Then, Chapter 4 describes the data preparation methods for the NF1-confirmed pediatric patient cohort. Chapter 5 describes the neural network architecture of the model and its implementation details including hyperparameter tuning schemes. Chapter 6 details the experiment design and evaluation results for this research. Finally, Chapter 7 summarizes the research and discusses its importance in clinical research applications.

Chapter 2: Disease Progression Management

Disease progression management is critical in providing precision care for the individuals with chronic diseases. Since there are various chronic conditions which can be non-life-threatening as long as a scheduled treatment is provided throughout the life stages, disease progression management is often much more useful for treating those conditions than a simple diagnostic system based on a cross-sectional observation. For those conditions, long-term progression needs to be monitored and therefore a scalable management model across a large range of timesteps is needed. Also, by having such a long-term disease progression management plan, clinicians can stratify the risk of each patient and schedule care accordingly.

Furthermore, disease progression management can be extended to not only provide scheduled care for the patients but also identify the temporal development of complex phenotypical representations associated with the chronic disease. Especially because modeling phenotype progression is a challenge for conditions with extreme clinical variability, a scalable management model for high dimensional phenotypical information needs to be developed.

Fortunately, there are an increasing number of machine learning models that can provide investigators and care providers with the tools needed to quickly generate hypotheses concerning the relationships between entities found in heterogeneous collections of scientific data — for example, exploring potential linkages by a gene, phenotype, and disease management protocols, thus enabling the forward engineering of prognostic and therapeutic strategies based on knowledge generated via basic science studies (Hood et al., 2004; Hood et al., 2004; Ahn et al., 2006; Payne et al., 2009; Payne et al., 2005; Schadt et al., 2012). The primary benefit of such multi-scale modelling is the ability to create an ensemble of features that maximizes predictive power, but is not constrained by data source type or scale.

Research Objective

To handle the limitations of the traditional statistical prediction models on longitudinal EHR data, this study demonstrates a predictive modeling pipeline that can predict a complex multi-phenotypical representation of the future state by learning temporal patterns across multiple dimensions. This predictive pipeline can extract and integrate clinical information from the EHR, construct a feature set, and apply a deep recurrent neural network architecture to model complex time stamped longitudinal data for monitoring and managing the progression of phenotype complexes that are associated with a disease condition. This study was conducted on a pediatric patient cohort with confirmed diagnosis of NF1, a chronic disease with extreme clinical variability that requires detailed disease progression management in the early life. In particular, this study focuses on the scalability of the predictive model for learning temporal clinical trends from high dimensional phenotypical feature vectors. For that purpose, long short-term memory units and gated recurrent units were chosen in my DRNN architecture to build a scalable prediction network. The prediction pipeline is differentiable from other efforts to-date that have sought to model NF1 progression, in that it involves the analysis of multi-dimensional phenotypes wherein the DRNN is able to model complex non-linear relationships between event points in the longitudinal data both temporally and cross-sectionally. Such an approach is critical when seeking to transition from traditional evidence-based care models to precision medicine paradigms.

Chapter 3: Recurrent Neural Networks

Traditional static neural networks are able to effectively model non-linear relationships with large dimension. However, static neural networks are not able to learn from the temporal information as it trains. Therefore, in order for the neural network to have the contextual information when making a prediction, information from the past states must persist. That led to the development of RNN (insert reference), which remembers the information from prior inputs throughout the time steps in the training process.

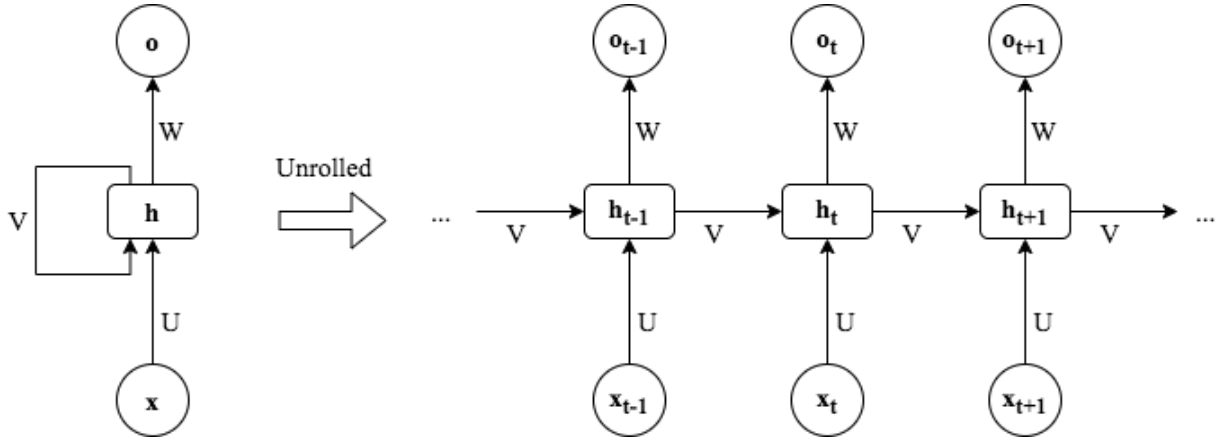

Figure 1. Diagram of the RNN architecture

As shown in Figure 1, the temporal information is passed through the hidden states across the time steps in the hidden layer. In other words, each output of the hidden state of the previous time step is used as the input to the hidden state of the next time step. The output of the hidden state is calculated using the equation:

$$h_t = \sigma(U * x_{t-1} + V * h_{t-1})$$

This process simply takes in the input sequence x_{t-1} and the output of the previous hidden state h_{t-1} and uses a sigmoid activation function to produce the output. We can also see that the weights U, V, and W are shared across time steps in order for the architecture to perform the exact same task at each step recurrently, thereby requiring much fewer parameters to train. In

such vanilla RNN architecture, we need to backpropagate through time in the weight update process using the gradient descent algorithm because all recurrent neurons in the past time steps are used to calculate the output. However, because weight V needs repeatedly multiplied to a time step in the far past, the gradient of the loss function decays exponentially with time, and we may face the vanishing gradient problem. When training a model on large sequences of temporal information, this will be problematic because if the difference in gradients is too small, the network will not learn anything and will not be able to perform meaningful weight updates. This motivated the development of a recurrent unit that can handle long-term dependencies, which is called long short-term memory unit (Hochreiter et al., 1997).

Long Short-Term Memory

Long short-term memory addresses the vanishing gradient problem by introducing the concept of internal memory units. In the RNN shown in Figure 1, weight V, which connects the hidden states, was causing the vanishing gradient problem. In LSTM, weight V can essentially be fixed to 1, which will not cause the gradient to vanish or explode even after it is multiplied across long time steps in the back-propagation process. Instead, a memory cell is present in the unit and it controls the propagation of the past information.

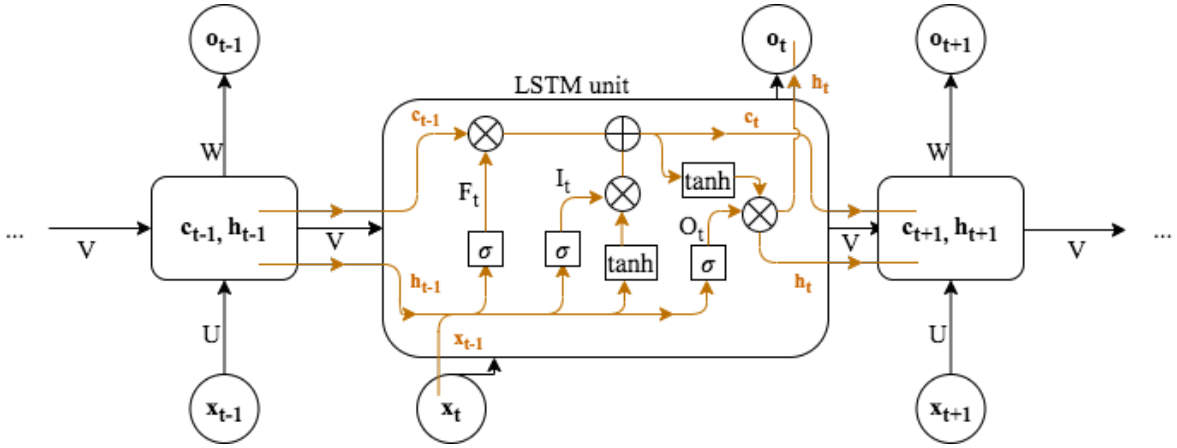

Figure 2. Diagram of the LSTM unit architecture

As shown in Figure 2, LSTM unit consists of a memory cell c and three gates, which are forget gate, input gate, and output gate, which control the memory cell state. It has three inputs, c_{t-1}, h_{t-1}, and x_t. h_{t-1} is the output of the previous hidden state, and x_t is the input vector at time step t. These two input vectors are combined and processed through the sigmoid activation function, which produces Boolean output matrix F_t that decides if the forget gate should be opened or closed based on the input. Based on the decision, c_{t-1}, the output of the previous memory cell state, is passed through. In other words, the forget gate controls how much of the value can remain in the memory cell c. At the same time, the combined input vector goes through the tanh activation function and also through another sigmoid function. The outputs of these two activation functions decide the input gate. The sigmoid output I_t decides for which of the input values the input gate should be opened or closed and to what extent. The tanh output decides the candidate values $\tilde{c}_t$that we want to feed into the memory cell. Using the two outputs I_t and $\tilde{c}_t$, we can now update the memory cell state. We take the sum of the leftover information of previous memory cell state and the candidate values that are passed through the input gate to calculate the new memory cell state c_t. The memory cell update process can be mathematically expressed as:

$$c_t = F_t * c_{t-1} + I_t * \tilde{c}_t$$

Finally, the combined input vector goes through a sigmoid activation function again. The sigmoid output O_t decides for which values of the new memory cell state we want to output. Then, the new memory cell state c_t goes through a tanh activation function to produce cell state values to output. We then multiply outputs O_t and $\tanh(c_t)$ to produce an output h_t, which also becomes the input to the next hidden state:

$$h_t = O_t * \tanh(c_t)$$

We can observe that LSTM units are very powerful in handling long-term dependencies using their internal memory units, especially in a large-scale time series input data.

Gated Recurrent Unit

Gated recurrent unit (GRU) is a variant of an LSTM unit with simpler gating mechanism (Cho et al., 2014). It was created to overcome the vanishing gradient problem of vanilla RNN, just like LSTM. However, GRU is different from LSTM in that it does not have a separate memory cell. Instead, it uses gating mechanisms to control the flow of memory within the unit.

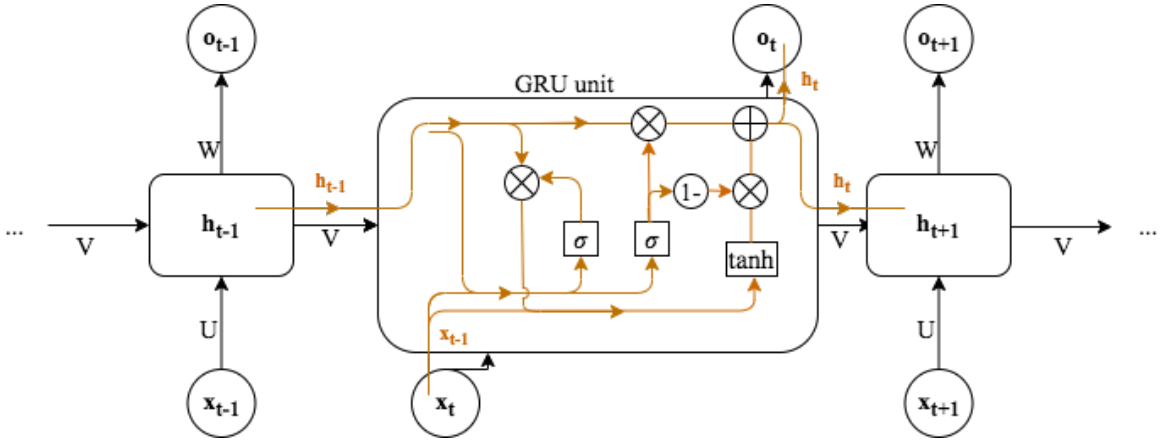

Figure 3. Diagram of GRU unit architecture

As shown in Figure 3, GRU unit consists of two gates: update gate and reset gate. It has two inputs, h_{t-1} and x_t. h_{t-1} is the output of the previous hidden state, and x_t is the input vector at time step t. These two input vectors are combined and processed through the sigmoid activation function, which produces Boolean output matrix Z_t that decides whether the update gate should be opened or closed. Based on the decision, h_{t-1}, the output of the previous hidden state, is passed through the memory. At the same time, a similar mechanism is used to produce a Boolean matrix R_t, with only different weights for the gate. R_t decides whether the reset gate should be opened or closed and which information from the past time steps should be removed. Furthermore, we can use the reset gate to calculate the current memory information. R_t is

multiplied with h_{t-1} to remove unnecessary information from the past time steps. Then, it is combined with the input vector x_t and passed through tanh activation function. It can be mathematically expressed as:

$$h'_t = \tanh(Ux_t + R_t * Vh_{t-1})$$

where U and V are weights of the two input vectors. Using this current memory information h'_t and the memory information from past time steps h_{t-1}, we can produce the final memory information to output, which will be the input to the next hidden state. We can multiply the update gate output Z_t with h_{t-1} to control the flow of past information. We can multiply the inverse of the update gate output, $1 - Z_t$, with h'_t to control the flow of current information. It can be mathematically expressed as:

$$h_t = Z_t * h_{t-1} + (1 - Z_t) * h'_t$$

We can observe that GRU units are also powerful in handling long-term dependencies using their gating mechanism, especially in a large-scale time series input data. The major difference between GRU and LSTM is that GRU units don't have output gates and they pass on the entire memory information to the future hidden state whereas LSTM units have output gates that control the memory information that flows to the future hidden state (Chung et al., 2014).

Chapter 4: Dataset

Neurofibromatosis Type 1 (NF1) exhibits extreme clinical variability, not only among unrelated individuals and among affected individuals within a single family but also even within a single person at different times in life (Jett et al., 2010). NF1 is one of the most common neurogenetic disorders, and occurs in 1 of every 3,000 births, without predilection for race, sex, or ethnicity (Friedman, 2002; Friedman, 1999). Therefore, it is not possible to have one standardized treatment plan that can fit the entire pediatric NF1 patient cohort. Since the dataset of the pediatric NF1 patient cohort has high dimensional phenotypical features and exhibits longitudinal visits of the patients, it naturally became our disease cohort of choice.

Data Collection

We extracted retrospective clinical data from two sources within the Washington University NF Center: the NF1 Clinical Registry and the EHR. From the NF1 Clinical Registry data, we extracted 29 unique clinical features in addition to longitudinal data on the development of NF1-related clinical features and associated diagnoses. From the EHR, multiple ICD 9/10 codes associated with NF1 – NF unspecified (ICD 9: 237.70; ICD 10: Q85.00); NF1-like syndrome (ICD 9:758.5; ICD 10: Q99.8); and NF1 (ICD 9: 237.71; ICD 10: Q85.01) – were extracted in order to account for clinical-level variability in usage of ICD9/10 codes and to capture a comprehensive patient cohort. All available diagnosis codes for each individual were extracted from the EHR-derived data set and were recorded as 15,890 unique ICD 9/10 codes. Given the high number of ICD 9/10 codes, a consistent, concept-level “roll up” of relevant codes to a single phenotype description was created by mapping the extracted ICD 9/10 values to phenome-wide association (PheWAS) codes called Phecodes (Denny et al., 2010; Denny et al., 2013), which have been demonstrated to better align with clinical disease compared to individual

ICD codes (Wei et al., 2017). Due to necessary privacy and confidentiality measures that were implemented during curation of the NF1 Clinical Registry, data sets were merged using first name, last name, and date of birth as keys. Full name plus date of birth as a unified linkage variable has previously been demonstrated to be an effective method of matching cross-database registries with sensitivity of over 87% (Kho et al., 2015).

Data Preprocessing

Diagnosis codes recorded in the EHR dataset can be merged onto the clinician-curated NF1 Clinical Registry to populate the records of the individuals with more information. Since EHR diagnosis codes represent all the encounters and sickness throughout the visits, it can also add some information of non-NF1-associated conditions that were present in the patient's history. While the 29 clinical features associated to NF1 from the NF1 Clinical Registry mark the most known sub-phenotypical representations of NF1 progression, EHR diagnosis also have the potential to capture complex non-linear relationships between the clinical features, which the clinicians have yet to discover. Before data from the EHR is merged onto the NF1 Clinical Registry data, the former must be transformed first. Data from NF1 Clinical Registry is formatted to have a feature vector for each unique visit of the individuals. However, data from the EHR is in the unrolled format and is formatted to have a unique diagnosis code for each unique visit of the individuals. Therefore, data from the EHR must be converted into a pivot table, with patient identification code and visit date as the keys. Upon completion, the transformed EHR data can be merged onto data from NF1 Clinical Registry, by an outer join on the two keys, patient identification code and visit date.

Data for 798 individuals were available in the NF1 Clinical Registry. All 798 individuals were confirmed patients with NF1. Also, records of 4631 unique individuals from the EHR were

extracted for the patient cohort with broader NF diagnosis codes present in the record. In order to merge the two datasets by a left join on the NF1 Clinical Registry's confirmed individuals, the overlapping individuals were identified using a map file that maps the unique key of NF1 Clinical Registry to the unique keys in the EHR data. A total of 734 mappable individuals were available in the map file. From the NF1 Clinical Registry, data for 732 individuals were mappable using the map file, and among those, 551 individuals had visit records under the age of 18. From the EHR dataset, data for all 734 individuals from the map file were mappable, and among those, 593 individuals had visit records under the age of 18. Because we are merging additional diagnosis information extracted from the EHR onto the patient cohort with NF1 confirmed from the NF1 Clinical Registry, we merge onto all the mappable individuals in the NF1 Clinical Registry. In particular, since our study is focusing on the disease progression during the pediatric ages, we only considered the visits under the age of 18, which reduces the number of mappable patients in the NF1 Clinical Registry to 551 individuals as described previously. Therefore, by filtering the data from the EHR to only include the 551 mappable individuals and performing an outer join on two keys, patient identification code and visit date, we are able to capture all visit information recorded in the hospital system for the 551 individuals.

Data Imputation

After the merge of the two extracted datasets, we obtain a very sparse comprehensive dataset. The addition of EHR data brought in 997 additional features, increasing the total size of the feature dimension to 1026. New visits added by the EHR dataset have empty values for the 29 NF1-associated feature values, and new visits observed by the EHR also have empty values for the 997 additional feature values. In order to fill the missing observations, forward imputation

method was used. Because a patient visits multiple times, we can propagate information from their last recorded visit to fill in the next visit. Since the 29 NF1-associated conditions have a slow progression rate, forward imputation from the closest past visit can sufficiently impute the missing observations without too many false assumptions. On the other hand, because we have no information on how the 997 additional features from the EHR progresses, we have no choice but to use the same method for imputing the feature values from the EHR. Missing visit information before the first recorded visit of the patient is filled with zeroes, assuming that the patient did not have the condition in the past and therefore did not visit the hospital.

Sequence Generation

In traditional prediction models, the current dataset is enough to be fed directly as an input, as it is already formatted in 2D, with visits as rows and features as columns. For the RNN, input is no longer a vector with a length of the feature dimension. Instead, it is a sequence of vectors, for which the sequence represents the timesteps and the vector represents the observed input vector at a time step.

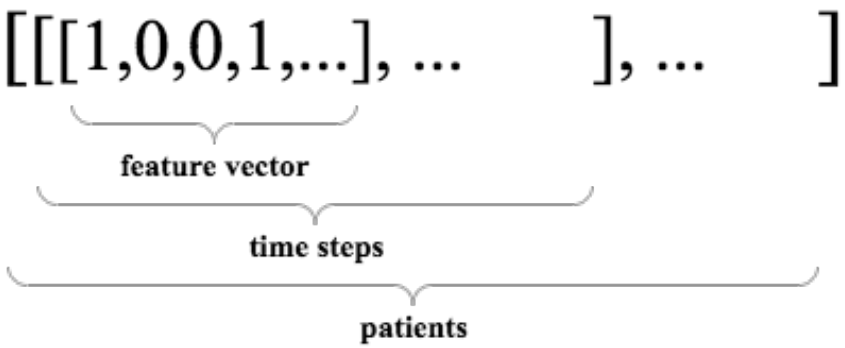

Figure 4. The input format to the RNN

As shown in Figure 4, the input the RNN is a 3D matrix, in which each dimension represents patients, timesteps, and features. It is quite obvious that each individual will have different number of visit records, and that can produce visit sequences with varying lengths.

This problem can be addressed by applying a padding across the temporal dimension of the input matrix, which pads the patient records to match the patient with the longest visit record history. Similarly, the sequences can be trimmed to match the patient with the shortest visit record history. While these methods have shown to be effective and have a very small impact on the performance of the model, they have less value when it comes to clinical application. In the disease progression management of NF1, standard practice is to observe the patient annually, because the important NF1-associated features do not have fast progression rates that needs to be frequently monitored. Therefore, discretizing the raw visit records into pre-fixed age interval bins is much more useful, as the clinicians have the freedom to choose the age intervals from which they want to get insight, depending on the particular disease's standard management practice of traditional care. Also, by allowing such freedom, we are able to address the gap between machine learning research from the laboratory and clinical practice. A careful design of predictive model that incorporates the standard disease management practice is a step closer to connecting the gap between two entities and can be extended to enable the forward engineering of prognostic and therapeutic strategies based on knowledge generated via basic science studies (Hood et al., 2004; Hood et al., 2004; Ahn et al., 2006; Payne et al., 2009; Payne et al., 2005; Schadt et al., 2012).

Using the knowledge from traditional NF1 disease management, we discretized and mapped each patient's visit records into 1 of the 36 age interval bins. Each age bin represents 6 months of age, and therefore 36 age bins represents the observation of the patient from age 0 to 18, by every six months. This effectively mimics the traditional management timeline of NF1, and by varying the length of training sequence or by varying the target prediction age, we can produce a disease progression prediction model with high interpretability and practicality. Any

empty bins were imputed by forward propagating the patient's last recorded visit before the bin. If no previous visit record to propagate from is filled with zeroes, assuming that the patient did not have the condition in the past and therefore did not visit the hospital.

This same process was repeated to produce a 3D input matrix to the RNN with just the data from the NF1 Clinical Registry. We constructed two sequence matrices, one from EHR-merged NF1 Clinical Registry data and the other solely from the NF1 Clinical Registry data. Both data are temporal, but they differ in the sense that merged dataset has extremely high dimensional feature vectors that includes all conditions from the patients' histories, which often may be irrelevant to NF1. On the other hand, the NF1 Clinical Registry only keeps track of 29 features that are directly related to NF1. Hence, both datasets will be used to compare the network's performance in learning from high-dimensional feature vectors.

Chapter 5: Implementation

To design an RNN architecture that is generalizable to predict the future onset of events in longitudinal EHR data, we need a network that can handle various scales of long-term dependencies. If the disease condition requires frequent measurement of a certain set of clinical features, the temporal dimension will be very large. On the other hand, if the disease condition requires a large number of clinical features that are measured sparsely, the feature dimension will also be very large. Therefore, our network also needs to be scalable enough to compute high dimensional feature vectors and capture the pattern across the feature vector, thereby being able to identify progression of complex multi-phenotypical patterns.

Deep Recurrent Neural Networks Architecture

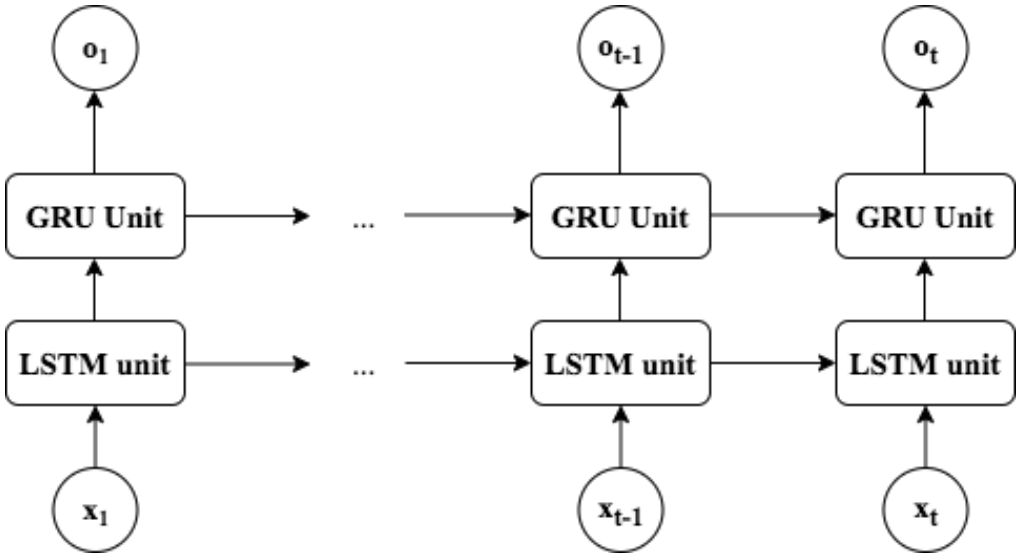

Figure 5. Simple diagram of 2-layer DRNN architecture with LSTM and GRU

Figure 5 shows a simplified diagram of the unrolled DRNN. Essentially, there are two hidden layers, each constructed with LSTM cells and GRU cells. These two particular RNN units were used because of their ability to generalize in scale to capture long-term dependencies across the time steps (Hochreiter et al., 1997; Cho et al., 2014). The structures of the two units are very similar – they use gating mechanisms to control the flow of memory information between the

hidden states. They differ slightly in the number of gates used and by the presence of a memory cell unit, and therefore, GRU can achieve a slightly better training time. However, in achieving the scalability across the temporal dimension, both architectures have been proven to be effective with similar performances (Choi et al., 2016). Even though for our NF1 dataset we have fixed the size of the temporal dimension to be 36, it can still exhibit a vanishing gradient problem depending on the size of our training sequences. For these reasons, a DRNN with two hidden layers consisting of two of the most popular RNN units is created.

Model Hyperparameters

Different number of units for each RNN cell, from a single unit to ten and one hundred, were tried and tested and the performances were compared to determine that 100 units is the optimal number. Similarly, the network was trained with various numbers of epochs: (a) 1, (b) 10, (c) 100 and the results were compared to decide that 100 training epochs are required for the network to train to find the optimal weights. Batch size in the training process was determined by comparing the performances of three types of batch gradient descent algorithms: stochastic batch gradient descent, mini-batch gradient descent, and batch gradient descent. Mini-batch gradient descent with a batch size of 32 was chosen to update the weights relatively frequently while avoiding the computational tradeoff of using a stochastic gradient descent. Binary Cross Entropy was the loss function of our choice because we are doing a multilabel binary prediction task. Adam optimizer was used to tune the learning rate of the network.

Chapter 6: Methods

In the past, we have explored the ability of much simpler classification models in predicting a NF1-associate disease condition. Simple logistic regression, support vector classifier, random forest algorithm, and gradient boosting algorithms were used to predict the presence of a target feature such as optic pathway glioma (OPG), attention deficit hyperactivity disorder (ADHD), or plexiform neurofibromas.

Since the classification models train on each sample's cross-sectional observation at the time of analyses, we obtained poor performance results. For example, a simple logistic regression model that predicts OPG had the best F1 score of 0.79 and the best recall of 0.57. The same model that predicts ADHD had the best F1 score of 0.71 and the best recall of only 0.44. Therefore, we moved on to use the RNN architecture to incorporate temporality and high dimensionality.

Two experiments were designed to directly tackle our goal of learning from the temporal progression of a disease cohort. The first experiment tests the performance of the DRNN model by varying the length of the training sequences. The next experiment trains the algorithm on the training sequences of fixed length, while trying to predict phenotypical representations in varying future time steps.

Experiment Design

All experiments were repeated on two different input datasets. NF1 Clinical Registry input and the EHR-merged input were each split to perform 10-fold cross validation on the dataset. The first experiment was designed to analyze the ability of the neural network to learn temporal patterns. The length of the training sequences was controlled up to varying age points, and the network predicts the sequence of patterns at age 18. By varying the length of the training

sequences, DRNN learns temporal patterns across multiple sizes of time steps. Furthermore, the traditional approach to NF1 progression management observes patients annually, so training sequences of length shorter than one year are not significant. For the network to be able to start observing patterns across time, the minimum length of training sequence was set to 3 years, which corresponds to 6 time steps. Then, the experiment was repeated with the length of the training sequence increased by 3 years until it reaches the age of 15. The algorithm's performance was compared across varying lengths of the training sequences.

The second experiment was designed to analyze the ability of the neural network to predict a state further in time. The length of the training sequences was fixed to be up to age 9 years, which corresponds to 18 timesteps. The age of choice was determined by analyzing the average onset for the 29 NF1-related features from the NF1 Clinical Registry. Most features have an average onset age after the age of 6, and by training on the sequence up to age of 9, the network should have sufficient temporal information. Furthermore, age of 9 is the midpoint of the pediatric life, where we have defined pediatric ages as age of 0 to 18. The DRNN predicts a sequence at an arbitrary time point t, where t was determined to be age of 12, 15, and 18. The algorithm's performance was compared across multiple future time points.

Results and Performances

The performance of DRNN was compared across training on varying lengths of training sequences. The network was trained on the sequences with varying lengths 6, 12, 18, 24, 30, and produces output sequence of phenotypes at the age of 18. The metrics used in this evaluation are maximum F1 score at validation, accuracy at validation, precision at validation, and recall at validation. The results were averaged across the sequence to produce a scalar value, which was averaged after the 10-fold cross validation process.

Table 1

DRNN Performance on NF1 Clinical Registry Data

Length of Training Sequence	Age Range in Years	F1 Score	AUC	Precision	Recall
6	3	0.62	0.89	0.65	0.59
12	6	0.74	0.93	0.75	0.73
18	9	0.84	0.97	0.85	0.84
24	12	0.91	0.98	0.92	0.89
30	15	0.95	0.99	0.98	0.92

Table 1 shows the performance of the DRNN when it is trained on various lengths of training sequences, only on the NF1 Clinical Registry data, and predicts a sequence of phenotypes at age of 18. As we stated in the experiment settings, the predictive ability improves significantly when we train the model on sequences up to age of 9. In particular, improvement in performance is maximized after observing patient records between age of 6 to 9, which is in line with our claim that average onset ages of NF1-associated clinical features are mostly after the age of 6.

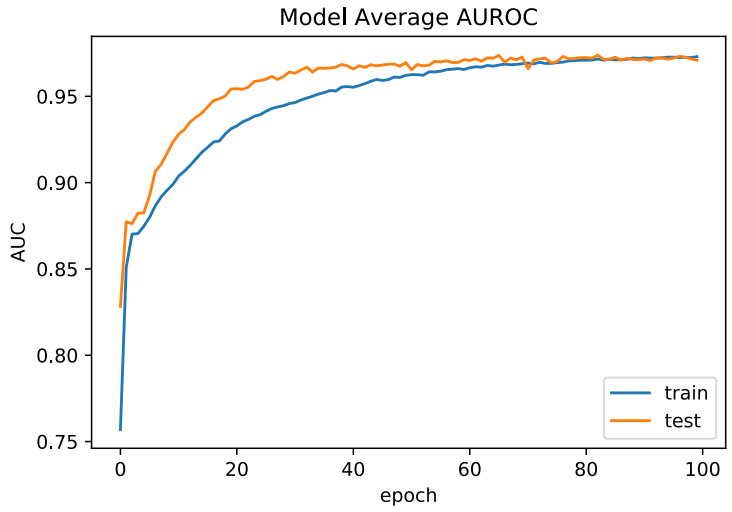

Figure 6. AUC of DRNN output at time step 36, trained on sequences up to age of 9.

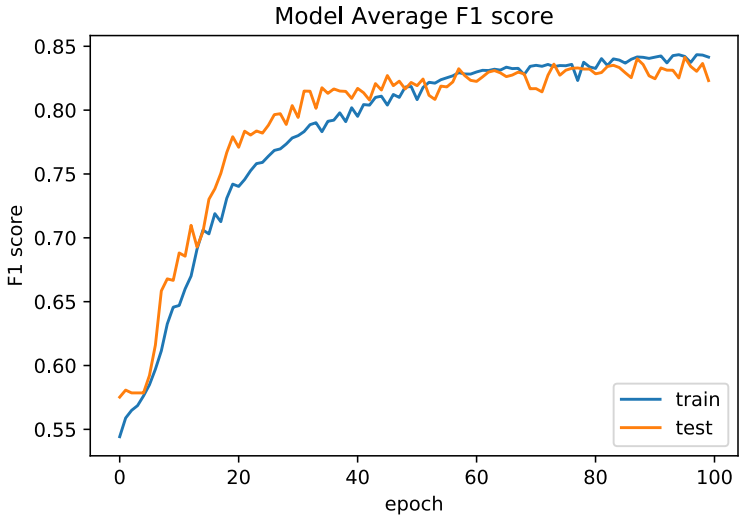

Figure 7. F1 Score of DRNN output at time step 36, trained on sequences up to age of 9.

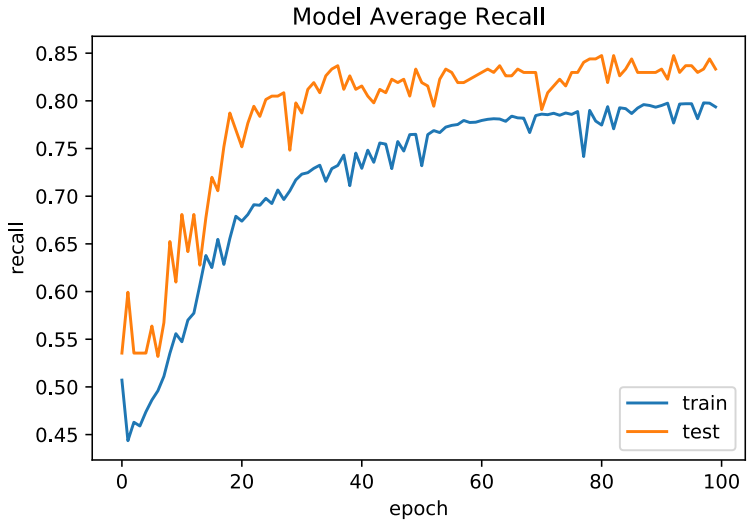

Figure 8. Recall of DRNN output at time step 36, trained on sequences up to age of 9.

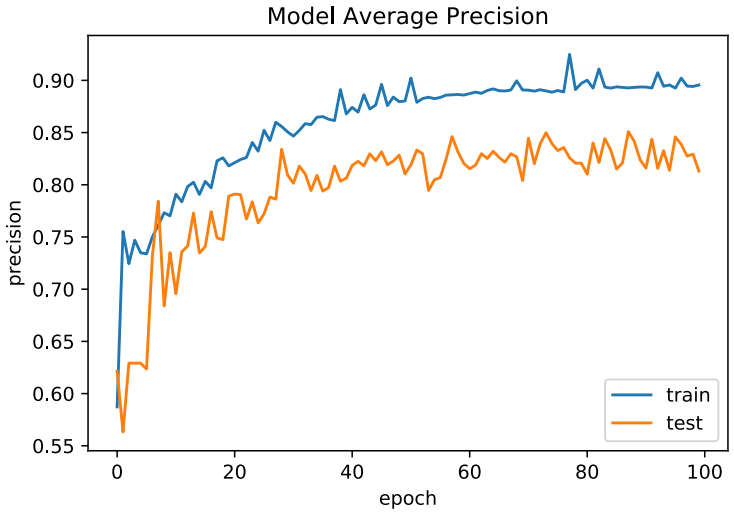

Figure 9. Precision of DRNN output at time step 36, trained on sequences up to age of 9.

As we can see from Figures 6–9, when trained on sequences up to age of 9, the performance metrics converge after 100 epochs of training.

Table 2

DRNN Performance on EHR-merged NF1 Clinical Registry Data

Length of Training Sequence	Age Range in Years	F1 Score	AUC	Precision	Recall
6	3	0.61	0.88	0.64	0.59
12	6	0.68	0.92	0.65	0.70
18	9	0.76	0.94	0.74	0.79
24	12	0.83	0.97	0.80	0.86
30	15	0.90	0.98	0.87	0.92

Table 2 shows the performance of the DRNN when it is trained on various lengths of training sequences, on the EHR-merged NF1 Clinical Registry data. Comparing with the results in Table 1, with the addition of 1000 more features, Table 2's results show a decrease in predictive performance as a tradeoff learning from complex high dimensional but it is still able to capture the complex patterns in a sufficient manner.

Furthermore, the performance of DRNN was compared across predicting phenotype sequence across multiple future time steps. The network was trained on the sequences with fixed length of 18 and produces output sequences of phenotypes at the age of 12, 15, and 18. The metrics used in this evaluation are maximum F1 score at validation, accuracy at validation, precision at validation, and recall at validation. The results were averaged across the sequence to produce a scalar value and then averaged after the 10-fold cross validation process.

Table 3

DRNN Performance on NF1 Clinical Registry Data

Time Steps in the Future	Age in Years	F1 Score	AUC	Precision	Recall
6	12	0.86	0.97	0.98	0.77
12	15	0.80	0.96	0.88	0.74
18	18	0.80	0.96	0.83	0.77

Table 4

DRNN Performance on EHR-merged NF1 Clinical Registry Data

Time Steps in the Future	Age in Years	F1 Score	AUC	Precision	Recall
6	12	0.77	0.96	0.83	0.72
12	15	0.72	0.95	0.72	0.72
18	18	0.72	0.94	0.68	0.76

From Tables 3 and 4, we can compare the performance of DRNN outputs in multiple target time steps in the future, while trained on fixed training sequence length. Interestingly, in predicting time steps after 3 years in the future, the prediction performance converges and does not show a significant decrease. Rather, we see an increase in recall with a tradeoff of decrease in precision. As the model tries to predict a sequence in the far future, it loses its ability to identify correct phenotypes but gains more power in capturing broader phenotypes that are relevant. Even after merging the EHR data to the NF1 Clinical Registry data, the DRNN is not able to find meaningful patterns in the features from EHR data.

Chapter 7: Conclusion

Hidden temporal trends across the longitudinal EHR are usually challenging to identify especially when the temporality exists across large feature dimension. In order to capture the temporal patterns that often span across multiple feature dimensions, a temporal prediction pipeline using DRNN architecture was developed and proposed in this study. From the datasets extracted from multiple sources of EHR, the prediction pipeline was able to merge and preprocess the dataset while taking the clinical properties of the longitudinal health data into account. Then, it was able to transform and discretize the data across the temporal dimension, which gives the researchers the freedom to explore the progression of the disease in the temporal scale of their choice. The choice of this deep architecture enabled us to develop a model that can outperform the stochastic statistical models. This disease progression management model can be applied for early stratification of risk, and furthermore, it has the potential to be utilized to stratify temporal progression patterns and rates of diverse clinical representations for diseases with extreme clinical variability.

Discussion

As we stated in the experiment settings, the predictive ability improves significantly when we train the model on sequences up to age of 9. In particular, improvement in performance is maximized after observing patient records between the ages of 6 to 9, which is in line with our claim that average onset ages of NF1-associated clinical features are mostly after the age of 6. Furthermore, incorporating the entire visit histories from the EHR was not effective in disease progression management of this particular disease condition. Such behavior can be explained in a few ways: (a) high dimensional features are much harder for the model to capture important relationships; (b) sparsity is brought in by the merging of EHR, thereby harder to observe

temporal trends across bigger feature set; and (c) 29 NF1-associated clinical features are sufficiently strong enough to produce a prediction. This result is surprising in that even when we incorporate the entire visit history information from the EHR, because of the sparsity paired with extremely high dimensional feature representations, the network could not predict the phenotype complex as well as how it would just by learning from the directly associated features. Extremely variable clinical representations of NF1 were apparent when merged with the high dimensional feature vector from the EHR. NF1 patient cohort data was particularly challenging to model, as it has small sample size and complex high dimensional sub-phenotypes to learn from. However, dimensionality reduction techniques were not explored in this research in order to keep the interpretability of the features when the predicted outputs are analyzed by clinicians. On the other hand, results leads to a more promising insight in that even with a small set of known phenotypical features identified clinically, with sufficient temporal information, the DRNN can produce much better predictive results than the stochastic models. Also, this research demonstrates a more practical approach of utilizing DRNN in statistical analysis of EHR. By utilizing clinical insights to formulate the input and design the experiment, we were able to achieve interpretability in analyzing prediction results and its potential to be utilized by the clinical researchers and bioinformaticians to model complex relationships between heterogeneous data.

References

Wei,W-Q., Bastarache, L. A., Carroll, R. J., Marlo, J. E., Osterman, T. J., Gamazon, E. R., et al. (2017). Evaluating phecodes, clinical classification software, and ICD-9-CM codes for phenome-wide association studies in the electronic health record. *PLoS ONE*, *12(7),* e0175508. doi:10.1371/journal.pone.0175508

Miled, Z. B., Haas, K., et al. (2020). Predicting dementia with routine care EMR data. *Artificial Intelligence in Medicine, Volume 102,* 101771. doi:10.1016/j.artmed.2019.101771

Dodek, P. M., Wiggs, B. R. (1998). Logistic regression model to predict outcome after in-hospital cardiac arrest: validation, accuracy, sensitivity and specificity. *Resuscitation, 36(3),* 201–208. doi:10.1016/S0300-9572(98)00012-4

Goldstein, B. A., Pomann, G. M., Winkelmayer, W. C., Pencina, M. J. (2017). A comparison of risk prediction methods using repeated observations: an application to electronic health records for hemodialysis. *Statistics in Medicine, 36(17),* 2750–2763. doi:10.1002/sim.7308

Cartling, F. J., Wolff, A. H. (2020). Temporal convolutional networks allow early prediction of events in critical care. *Journal of the American Medical Informatics, 27(3),* 355–365. doi:10.1093/jamia/ocz205

Laksana, E., Aczon, M., Ho, L., Carlin, C., Ledbetter, D., Wetzel, R. (2020). The impact of extraneous features on the performance of recurrent neural network models in clinical tasks. *Journal of Biomedical Informatics, 102,* 103351. doi:10.1016/j.jbi.2019.103351

Wang, T., Tian, Y., Qiu, R. G. (2019). Long short-term memory recurrent neural networks for multiple diseases risk prediction by leveraging longitudinal medical records. *IEEE Journal of Biomedical and Health Informatics,* 1–1. doi: 10.1109/JBHI.2019.2962366

Friedman, J. (2002). Neurofibromatosis 1: clinical manifestations and diagnostic criteria. *Journal of child neurology, 17*, 548-554. doi:

Friedman, J. (1999). Epidemiology of neurofibromatosis type 1. *American Journal of Medical Genetics, Part A (89)*, 1–6. doi:

Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *ACM Neural Computation, 9(8)*. doi: 10.1162/neco.1997.9.8.1735

Cho, K., Merriënboer, B., Gülçehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Conference on Empirical Methods in Natural Language Processing,* 1724–1734. doi:10.3115/v1/D14-1179

Chung, J., Gülçehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. Retrieved from arXiv.org, arXiv:1412.3555

Jett, K., Friedman, J. M. (2010). Clinical and genetic aspects of neurofibromatosis 1. *GeneTest Review: Genetics in Medicine, 12,* 1–11.

Denny, J. C., et al. (2010). PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene-disease associations. *Bioinformatics, 26,* 1205–1210. doi:10.1093/bioinformatics/btq126

Denny, J. C. et al. (2013). Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data. *Nature Biotechnology, 31,* 1102-1111. doi:10.1038/nbt.2749

Kho, A. N., et al. (2015). Design and implementation of a privacy preserving electronic health record linkage tool in Chicago. *J Am Med Inform Assoc, 22,* 1072-1080, doi:10.1093/jamia/ocv038

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., Sun, J. (2015). Doctor AI: predicting clinical events via recurrent neural networks. Retrieved from arXiv.org, arXiv:1511.05942

Appendix A: Onset Age Table for NF1-associated clinical features

Feature	Total Number	Total Mean Onset Age	Total Median Onset Age
Cafe-au-lait Macules	438	7.8	7.3
Skinfold Freckling	310	7.6	6.5
Dermal Neurofibromas	212	9.3	9
Lisch Nodules	216	8.6	8.2
Optic Pathway Glioma	67	7.6	6.4
Scoliosis	69	11.3	12.5
Orbital Dysplasia	2	11.5	11.5
Radial or Ulnar or Tibular or Fibular Dysplasia or Psuedoarthrosis	29	7	5.2
Plexiform Neurofibroma	113	8.9	8.6
MPNST	6	12.1	14.3
T2 Hyperintensities: Basal Ganglia	90	8.2	7.6
T2 Hyperintensities: Brainstem	50	8.6	7.7
T2 Hyperintensities: Cerebellum	91	7.5	6.8
T2 Hyperintensities: Optic Pathway	14	7.2	6
T2 Hyperintensities: Other (specify)	114	8.6	7.8
Autoimmune Systems	7	12	13.8
Growth Hormone Deficiency	7	11.2	12
Precocious Puberty	17	9.8	8.9

Endocrine Issues: Other (specify)	12	7.7	6
Heart Murmur (specify)	10	7.5	5.6
Hypertension	7	11.7	11.3
ADHD	124	9.5	9.1
Cognitive Impairment	64	7.8	7.2
Depression	7	14.6	14.9
Learning Disability	165	9.4	8.8
Has or had a Brain Cancer	0	N/A	N/A
Has or had another Cancer	0	N/A	N/A