META-LEARNING IN MEDICINE

A Specialization Project Report

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

MSc.

by

Yong Huang, Pargol Gheissari

May 2020

ABSTRACT

In recent years, the amount of digital information stored in electronic health records (EHRs) has increased dramatically. At the same time, the advances in the field of machine learning, specifically deep learning has accommodated the opportunity for knowledge discovery and data mining algorithms to gain insight from this digital health data. Predictive modeling of clinical risks from EHRs, such as in-hospital mortality rate, in-hospital length of stay and chronic disease onset, can be helpful to the improvement of the quality of healthcare delivery. However, there are many challenges, such as sparsity, irregularity and temporality, associated with this clinical data. Therefore, this provides an opportunity for meta-learning methodologies to solve such problems and to have a large impact on medicine and quality of healthcare delivery. In this paper, we provide the background of this problem, review the commonly used strategies for solving such problems and discuss the state-of-the-art of meta-learning models. To address the clinical challenges associated with EHR data, we propose a meta-learning model, which uses latent-ODE as the base-learner and LSTM as the meta-learner, to solve disease phenotyping tasks. We then demonstrate that our proposed method outperforms the state-of-the-art models addressing classification tasks on healthcare data.

ACKNOWLEDGEMENTS

We thank all advisors and staff who supported us throughout this process; especially, Dr. Deborah Estrin, Dr. Fei Wang, Dr. Xi Sheryl Zhang and Dr.Calvin Zang.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Each year over 30 million patients visit hospitals in the United States. 83 % of the hospitals use an electronic health record (EHR) system [8]. With the surge in the availability of digital clinical data, a significant increase of interest in using data mining algorithms to improve the quality of healthcare delivery has been observed. Predictive modeling of clinical risks from patient EHRs, such as in-hospital mortality rate, length of stay, chronic disease onset and phenotype classification, have attracted attention in this field [26]. Accurate clinical risk prediction models can help clinicians identify the potential risk at early stages and allow for appropriate actions to be taken in a timely manner; thus, resulting in the improvement of healthcare delivery.

Although there has been a steady growth in developing such algorithms, including both conventional approaches and deep learning models, several obstacles have slowed the progress in harnessing digital health data. In medical settings, labeled data samples are typically limited and are often very expensive to obtain. Therefore, sequentiality, sparsity, noisiness and irregularity are some challenges when working with EHR data [26]. Furthermore, there is accumulating evidence that many prediction tasks are interrelated. For instance, the highest risk and highest cost patients are often those with complex comorbidities while decompensating patients have a higher risk for poor outcomes. Thus, making efficient use of limited patient samples becomes essential to accurately predict complicated clinical risks [2][8].

Another challenge in this field is the absence of widely accepted benchmarks for evaluating competing models. Such benchmarks accelerate progress

in machine learning by bringing the community into focus and facilitating reproducibility and competition. For example, the winning error rate in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) plummeted an order of magnitude from 2010 (0.2819) to 2016 (0.02991). In contrast, practical progress in clinical machine learning has been difficult to measure due to variability in data sets and task definitions [8].

Our machine learning techniques, similar to human intelligence, should be able to learn and adapt quickly from a few examples and continue to adapt as more data becomes available [23]. Meta-learning, also known as learning to learn, addresses this problem [24] [15]. It is the science of systematically observing how different machine learning approaches perform on a wide range of learning tasks, and then learning from this experience [7] [22]. Although meta-learning algorithms have been explored in applications such as robotics and neural machine translation to address the similar problem of limited samples, the application of these algorithms in medical problems are rarely explored. In order to address the problem of learning from a few examples, we reviewed the commonly used strategies in modern state-of-the-art meta-learning methods and constructed a meta-learning model to address the task of phenotyping in ICU data.

CHAPTER 2

RELATED WORK

In this section, we systematically explore and select literature to elucidate and summarize the application of meta-learning in clinical settings, specifically on EHR dataset. This allowed us to discover potential gaps and areas for innovation in this field. We have identified the research questions and relevant methods used to address these questions, which subsequently allowed us to develop an innovative model for solving this problem. We first discuss the solutions that have been proposed to this date that address some challenges with the EHR systems. Next, we discuss the state-of-the-art meta-learning models and the application of certain models to EHR data.

2.1 Challenges and Solutions Associated with EHR Systems

EHRs are a comprehensive collection of patient care details, such as order of medications, procedures, lab tests, and diagnosis. They are an important source of information for healthcare technology [6]. The patient records in these databases are usually recorded with thorough details, including both numerical, structured, and textual data. The sources of numerical data are measurements such as heart rate, blood pressure, and clinical test results. The structured data are in the form of medical codes associated with each patient record. These codes are manually assigned by the clinical decision maker for billing and administration purposes. The unstructured data comes from the clinical notes that contain detailed natural language description of the healthcare provided to the patients during the admission [12].

As obtaining publicly available data for running experiments in clinical setting is an issue, many studies focus on using the public Medical Information Mart for Intensive Care (MIMIC) data. This data is associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012, and includes information such as demographics, vital sign measurements, laboratory test results, procedures, medications, imaging reports, and mortality, both in and out of hospital. Recently, investigators have also explored how interpretation mechanisms for deep learning models could be applied to clinical predictions [9]. Most common tasks involving EHR data include but are not limited to information extraction, representation learning, outcome prediction, phenotyping and de-identification. A more detailed description of these tasks and benchmark methods can be seen in figure 2.1.

Task	Subtasks	Input Data	Models
Information Extraction	(1) Single Concept Extraction (2) Temporal Event Extraction (3) Relation Extraction (4) Abbreviation Expansion	Clinical Notes	LSTM, Bi-LSTM, GRU, CNN RNN + Word Embedding AE Custom Word Embedding
Representation Learning	(1) Concept Representation (2) Patient Representation	Medical Codes	RBM, Skip-gram, AE, LSTM RBM, Skip-gram, GRU, CNN, AE
Outcome Prediction	(1) Static Prediction (2) Temporal Prediction	Mixed	AE, LSTM, RBM, DBN LSTM
Phenotyping	(1) New Phenotype Discovery (2) Improving Existing Definitions	Mixed	AE, LSTM, RBM, DBN LSTM
De-identification	Clinical text de-identification	Clinical Notes	Bi-LSTM, RNN + Word Embedding

Figure 2.1: Common tasks using EHR data and benchmark models for these tasks

Deep learning approaches have achieved significant results in many of these

tasks, such as medical concept representation. Efficient representations for medical concepts is an important element in healthcare applications. Medical concepts contain rich latent relationships that cannot be represented by simple one-hot coding. For example, pneumonia and bronchitis are clearly more related than pneumonia and obesity. In one-hot coding, such relationships between different codes are not represented [5]. Recently, studies using deep learning approaches in this field have demonstrated significant improvement in the performance of various predictive models without the need for medical expertise. Methods such as word embedding, recurrent neural networks (RNN), convolutional neural networks (CNN) or stacked denoising autoencoders (SDA), have been developed [6]. As these models do not require expert feature reconstruction, they perform significantly better than logistic regression or multilayer perceptron models.

Nonetheless, efficiently learning the representations of healthcare concepts remains a challenge. First, EHR data have a unique structure where the visits are ordered with respect to time but the medical codes within a visit are unordered. As there is a sequential relationship between the visits, the sequence of visits cannot be captured by simply aggregating code-level representations. Second, while the interpretability of the state-of-the-art representation learning methods in healthcare is essential, it is difficult to interpret many of these models such as recurrent neural networks (RNN). Finally, the algorithm should be scalable enough to handle large EHR datasets with hundreds of thousands of patients and millions of visits [5]. However, the good performance of current deep learning methods relies much on the amount of high quality labeled data. In many prediction tasks, labeled data are desired but very limited. As we discussed before, meta-learning approaches have strong potential for solv-

ing this problem. Modern meta-learning algorithms, such as Model-Agnostic Meta-Learning, enable us to quickly adapt to new tasks and make accurate predictions with few examples. In the following part, we will discuss the most commonly used strategies in meta-learning approaches.

2.2 Neural ODE on EHR

RNNs are currently the dominant model class for high-dimensional, regularly-sampled time series data. However, since EHR data typically consists of sporadically observed longitudinal patient data that have no standard method to align patient trajectories, RNN models will not be able to provide accurate results. On the other hand, since neural ODEs are essentially a continuous version of neural networks, they facilitate modeling time series data. Thus, the difficulties associated with the analysis of these datasets provide the opportunity for neural ODEs to have a large impact on solving such problems. In these models, instead of specifying a discrete sequence of hidden layers, the continuous dynamics of the hidden units is parameterized using an ordinary differential equation (ODE) specified by a neural network [4].

These models have several benefits including memory efficiency, adaptive computation, parameter efficiency, scalable and invertible normalizing flows, continuous time-series models.

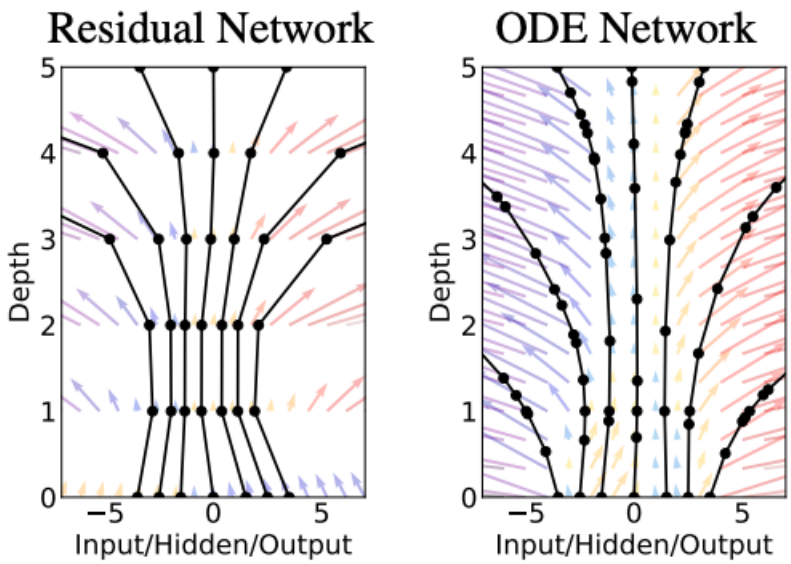

Figure 2.2: Continuous dynamics of ODE network [4]

2.3 Meta-learning

Meta-learning, also known as learning to learn, essentially allows for machines to learn new skills and concepts fast with a few training examples. A good meta-learning model is expected to be capable of adapting to new tasks and new environments quickly, even if it has never seen those before during training time. Meta-learning methods have a wide range of applications in supervised learning and reinforcement learning. Few-shot classification problems are examples of meta-learning in supervised learning setting. In this paper, we focus on discussing meta-learning in a supervised learning setting.

In a typical machine learning setting, the dataset D is split such that the parameters θ are optimized based on the training set D_{train} and their generalization is evaluated on the test set D_{test}. On the other hand, in a few-shot learning setting, the meta-sets D contain multiple regular datasets, where each $D \in D_{meta}$

has a split of D_{train} (support set) and D_{test} (query set). Typically, a K-shot N-class classification task is considered in such problems, where the support set contains K labelled examples for each of N classes. The objective during meta-training is to learn an efficient learning procedure (meta-learner) that can produce a classifier (the base-learner) with high average classification performance on its corresponding test set D_{test}. During meta-testing, the generalization performance on $D_{meta-test}$, where there are labels not seen during the meta-training process, is evaluated. An example of this setup can be seen in figure 3.1. In this figure, the top represents the meta-training set $D_{meta-train}$,where inside each gray box is a separate dataset that consists of the training set D_{train} (left side of dashed line) and the test set D_{test} (right side of dashed line). In this illustration, we are considering the 1-shot, 5-class classification task where for each dataset, we have one example from each of 5 classes (each given a label 1-5) in the training set and 2 examples for evaluation in the test set. The meta-test set $D_{meta-test}$ is defined in the same way, but with a different set of datasets that cover classes not present in any of the datasets in $D_{meta-train}$ (similarly, we additionally have a meta-validation set that is used to determine hyper-parameters) [15].

Overall, modern meta-learning models can be categorized into three different types: metric-based meta-learning [20][21][16][11][3], model based meta-learning [14][25][18], and optimization-based meta-learning [16]. Most recent meta-learning research has been focusing on optimization-based models. Traditional gradient-based optimization in training deep models is not designed to deal with a small number of samples or to converge with a small number of optimization steps. Optimization-based meta-learning methodologies attempted to solve this problem by designing special optimization algorithms that are able to learn good initialization of the model's parameters with small amount of data

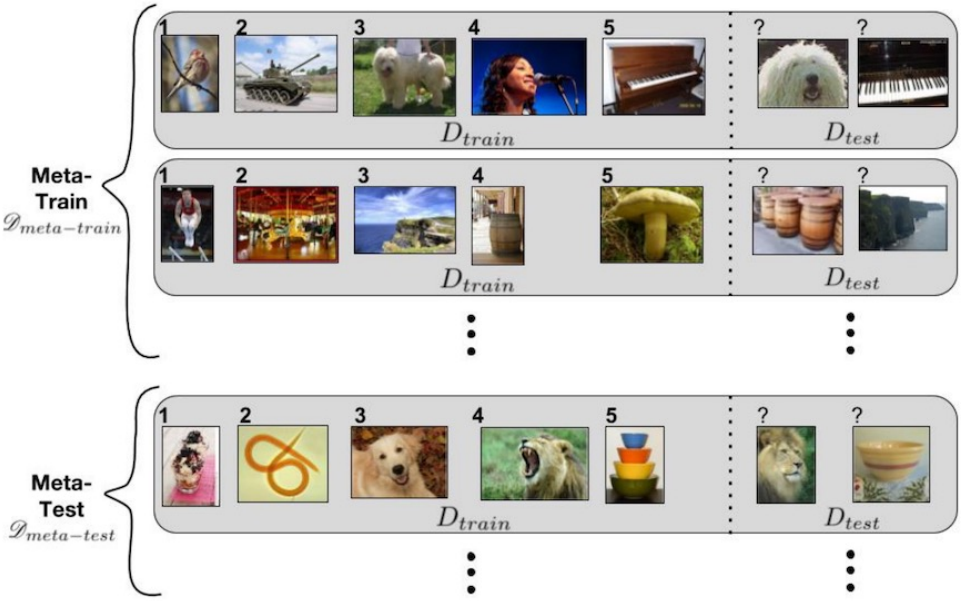

Figure 2.3: Example of meta-learning setup [15]

and small amount of optimization steps. One benefit of optimization-based meta-learning models is that it explicitly introduces the concepts of base-learner and meta-leaner and decouples the functions of these two. This accommodates for the flexibility of the base-learner design, which allows for applying meta-learning methods to not just few-shot image classification problems, but also to many other problems. In meta-learning, the base-learner is the learning component of the model that learns how to perform a specific task. For instance, a base-learner can be a convolutional neural network that learns to classify images. Furthermore, the meta-learner is another learning component of the model that learns how to update base-learner's parameters according to the loss and gradient information from the base-learner on the support set.

CHAPTER 3

METHODS

3.1 Base-Learner

Problems associated to EHR data, such as sparsity and irregularity, create the opportunity for finding new methodologies that can help mitigate these problems. To be more specific, EHR data is recorded in a sporadic manner which means that time intervals between observations are not fixed. A simple way to handle irregularly-timed samples is to include the time gap between observations into RNN update function; however, this approach assumes that the hidden state from the previous observation time-point can be directly used in the next hidden state update step. An alternative to this approach is to introduce an exponential decay of the hidden state between the observation intervals. Recent research on neural ODE has introduced a more accurate way to model the hidden state dynamics. In our work, we follow one key observation from a previous study where an RNN with exponentially-decayed hidden state implicitly obeys the following ODE [13]:

$$\text{ODE } \frac{dh(t)}{dt} = -\tau h; h\left(t_0\right) = h_0$$

Recently proposed models called ODE-RNN and latent ODE [17] provide a more efficient and explicable way to model the hidden state update. For our model, we used ODE-RNN and latent ODE as our base-learner and adapted two ODE-based models. We then applied this model to disease classification task.

3.1.1 ODE-RNN

As mentioned, time series data with non-uniform intervals, especially in medical settings, are difficult to model using standard autoregressive models such as RNNs. These models are often sufficient for densely sampled data, but perform worse when observations are sparse. ODE-RNN tries to model the hidden state using a Neural ODE. A detailed description of ODE-RNN is given in Algorithm 1.

Algorithm 1: ODE-RNN

Input: Data points and their timestamps $\{(x_i, t_i)\}_{i=1...N}$

Output: Last Hidden state and output at each timestamps

1: **procedure** ODE-RNN
2: $h_0 \leftarrow 0$
3: **for** `i in 1,2,...,N` **do**
4: h'_i = ODESolve $(f_\theta, h_{i-1}, (t_{i-1}, t_i))$
5: h_i = GRUCell $\left(h'_i, x_i\right)$
6: $o_i = OutputNN(hi)$ for all i = 1...N
7: **return** $\{o_i\}_{i=1..N}; h_N$

3.2 Latent ODE

RNNs can be generalized to have continuous-time hidden dynamics defined by ODEs to address such difficulties in modeling. Latent ODEs define a generative

process over time series based on the deterministic evolution of an initial latent state, and can be trained as a variational autoencoder [10]. Latent-ODEs can handle time gaps between observations, and remove the need to group observations into equally-timed bins [17].

In 2018, Chen et al. proposed a latent-variable time series model, where the generative model is defined by ODE whose initial latent state z_0 determines the entire trajectory [4]:

$$z_0 \sim p_0$$
$$z_0, z_1, ..., z_N = ODESolve(f_\theta, z_0, (t_0, t_1, ..., t_N))$$
$$eachx_i \overset{indep.}{\sim} p(x_i|z_i) i = 0, 1, ..., N$$

In this model, a variational autoencoder framework is used for both training and prediction, which is essentially an encoder-decoder architecture. The input to the encoder is a variable-length sequence, such as time series data, that is then encoded into a fixed-dimensional embedding. The output from the encoder is then decoded to another variable-length sequence and the trajectory is reconstructed. There are several benefits to using a latent variable framework. First, this framework allows for examining the dynamics of the ODE system, the likelihood of observations, and the recognition model, separately without any association to each other. Second, the posterior distribution over latent states provides an explicit measure of uncertainty, which is not available in standard RNNs and ODE-RNNs. Finally, it becomes easier to answer non-standard queries, such as making predictions backwards in time, or conditioning on a subset of observations.

Algorithm 2: Latent-ODE

Input: Data points and their timestamps $\{(x_i, t_i)\}_{i=1...N}$

procedure LATENT-ODE

$z_0' = \text{ODE} - \text{RNN}\left(\{x_i\}_{i=1..N}\right)$

$\mu_{z_0}, \sigma_{z_0} = g_\mu\left(z_0'\right), g_\sigma\left(z_0'\right)$ ▹ g_μ and g_δ are MLPs

$z_0 \sim \mathcal{N}\left(\mu_{z_0}, \sigma_{z_0}\right)$

$\{z_i\} = \text{ODESolve}\left(f, z_0, (t_0 \ldots t_N)\right)$

$\tilde{x}_i =$ Output $NN\left(z_i\right)$ for all $i = 1..N$

return $\{\tilde{x}_i\}_{i=1...N}$

3.3 Meta-Learner

Optimization-based meta-learning algorithms are designed to adjust the optimization algorithm commonly used in deep learning so that the model can excel in learning with a few examples. Meta-LSTM is one of the approaches that tries to explicitly model the optimization algorithm.

3.3.1 Meta-LSTM

We followed Ravi and Larochelle's [15] work in our project. This is paper is a pioneer in explicitly introducing the concept of "meta-learner", where the model for handling the specific task is called "learner". The goal of the meta-learner is to efficiently update the learner's parameters using a small support set so that

the learner can adapt to the new task quickly.

We denote the learner model as M_θ parameterized by θ, the meta-learner as R_Θ with parameters Θ, and the loss function $\mathcal{L}$.

There are two very interesting intuitions behind this idea. First, there is similarity between the gradient-based update in backpropagation and the cell-state update in LSTM. Second, knowing a history of gradients benefits the gradient update; for instance, the momentum algorithm.

The update for the learner's parameters at time step t with a learning rate α_t is:

$$\theta_t = \theta_{t-1} - \alpha_t \nabla_{\theta_{t-1}} \mathcal{L}_t$$

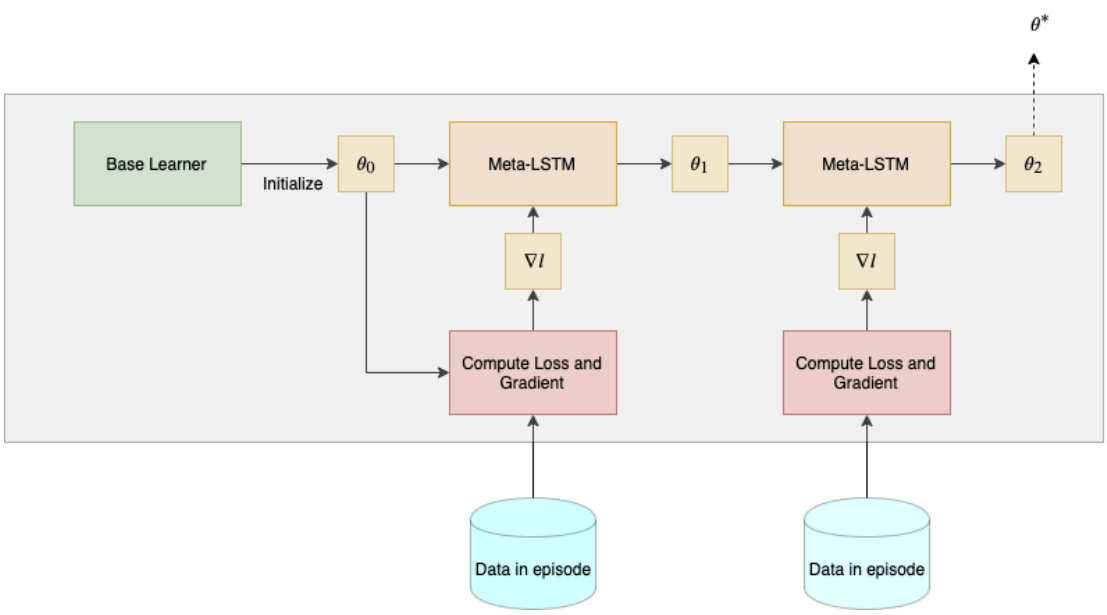

Figure 3.1: LSTM as a meta-learner

It has the same form as the cell state update in LSTM, if we set forget gate

$f_t = 1$, input gate $i_t = \alpha_t$, cell state $c_t = \theta_t$, and new cell state $\tilde{c}t = \nabla_{\theta_{t-1}} \mathcal{L}_t$:

$$
\begin{aligned}
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
&= \theta_{t-1} - \alpha_t \nabla_{\theta_{t-1}} \mathcal{L}_t
\end{aligned}
$$

3.3.2 Our Modification

In our work, we expand the original model by stacking a normal LSTM cell on top of the meta-LSTM cell. The purpose for this design is to match the input dimension and output dimension in meta-LSTM since the output of meta-LSTM is the new base learner parameters while the input consists of the gradients and loss besides the original base leaner parameters.

CHAPTER 4
RESULTS

4.1 Data Pre-Processing

As mentioned in previous sections, despite the rapid growth in applying machine learning methods to clinical data, the progress in this field is less significant than the progress in other applications of machine learning. In addition to factors such as data complexity, noisiness and sparsity, absence of community benchmarks contribute to the slower progress of machine learning in clinical settings. Benchmarks can play an important role in accelerating progress in machine learning research; they facilitate reproducibility and direct comparison of competing ideas.

Recently, many studies focus on using the public Medical Information Mart for Intensive Care (MIMIC) data. This data is integrates de-identified, comprehensive clinical data of patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between, which includes information such as demographics, vital sign measurements and laboratory test results [9]. In order to conduct any analysis on this data, a training must be completed for the data use agreement.

In our analysis, we mainly focus on disease classification. Thus, we executed the necessary filtering steps to obtain the subset of patients with acute diseases. In this process, we first generated a directory per patient that included the ICU stay information, diagnoses and events for that patient. Next, we filtered patients with missing ICU stay ID, missing ICU length of stay and ICU stays with no events. Approximately 80% of the data remained after this filtering step. Next, we filtered patients with mixed diseases and chronic disease. Since we are mainly focusing on disease classification we exclude patients with mixed diseases. Finally, we constructed the episodes needed for few-shot learning.

4.2 Meta-Learner Experiments

In this section, we describe the results of our meta-learner experiments and the comparison against other meta-learning models. We followed the standard experiment settings in the meta-learning community and tested our meta-learner performance on the MiniImageNet dataset. The MiniImagenet dataset was proposed by Ravi & Larochelle [15], and involves 64 training classes, 12 validation classes, and 24 test classes. Our base learner follows the same architecture as the

Methods	Accuracy
Matching Network	51.09 ± 0.71%
Matching Network FCE	55.31 ± 0.73%
RNN-VAE	0.515 ± 0.040
ODE-RNN	**0.833 ± 0.009**
Latent-ODE	0.826 ± 0.007

Table 4.1: 5-shot 5-class Accuracy on MiniImageNet

Methods	AUC-ROC
RNN-impute	0.764 ± 0.016
RNN-decay	0.807 ± 0.003
RNN-VAE	0.515 ± 0.040
ODE-RNN	**0.833 ± 0.009**
Latent-ODE	0.826 ± 0.007

Table 4.2: Mortality prediction on Physionet

CNN used by Vinyals et al.[23], which has 4 modules with a 3 x 3 convolutions and 64 filters, followed by batch normalization, a ReLU nonlinearity, and 2 x 2 max-pooling layer.

One of the major challenges in this implementation is compressing the parameter space in LSTM meta-learner. As the meta-learner is modeling parameters of another neural network, it has hundreds of thousands of variables to learn. We followed the idea proposed by Andrychowicz [1], which is sharing parameters across coordinates. Furthermore, to simplify the training process, the meta-learner assumes that the loss ∇t and the gradient $\nabla_{\theta_{t-1}} \mathcal{L}_t$ are independent. The results of the experiments are shown in table 4.1. This table indicates that although Model-Agnostic Meta-Learning (MAML) [7] achieved better performance than our approach, it requires vast amount of memory usage and is computationally inefficient in since it involves second order derivative computation. For our project, we take model complexity into consideration as well.

4.3 Base-Learner Experiments

We evaluated ODE-based models on the PhysioNet Challenge 2012 dataset [19]. This dataset contains 8000 time series, each time series contains measurements from the first 48 hours of a different patient's admission to ICU. Measurements were made at irregular times, and of sparse subsets of the 37 possible features. Due to label imbalance, we used AUC-ROC as our metrics of evaluation. Our results show that ODE-based models outperform traditional RNN-based models. In our experiments we constructed a classifier to predict in-hospital mortality. We passed the hidden state at the last measured time point into a two-layer classifier and jointly train both the encoder and decoder by maximizing the evidence lower bound (ELBO). Through these experiments, we conclude that the best performance is achieved by reconstructing the trajectory at the same time as predicting disease categories. The results of these experiments are shown in table 4.2

4.4 Disease Classification in a Normal Data Regime

First, we compared the performance of the ODE-based models in a normal machine learning setting where the amount of labeled data is not constricted compared to the few-shot learning setting. In our experiments, we chose the imputed LSTM and GRU-D as our comparison models. The best LSTM baseline performance is achieved using standard LSTM network with missing values

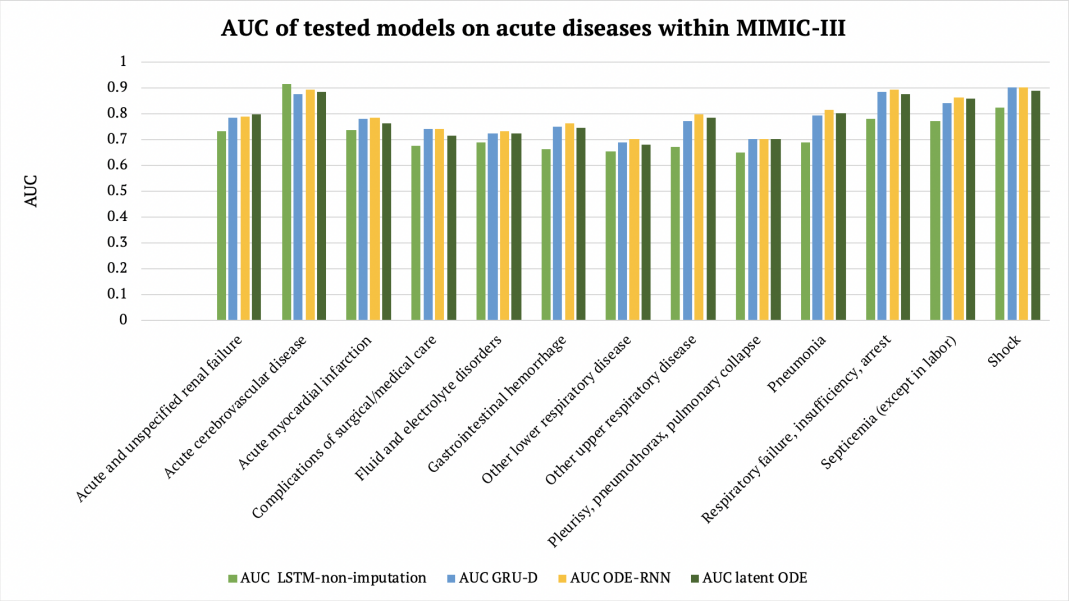

Figure 4.1: AUC of tested models on acute diseases within MIMIC-III

imputation strategy and without leveraging multitasking training. Our ODE-based model achieved good performance without using any imputation strategy. The results from our experiments indicate that the best ODE-based model performance is achieved when we are reconstructing the trajectory at the same time during training. Furthermore, our model has significantly outperformed standard LSTM without using any imputation strategy. However, the LSTM-imputed achieved similar results compared to ODE-based model. One possible reason is that standard RNNs are ignoring the time gaps between points. Typically, standard RNNs work well on regularly spaced data with few missing values, or when the time intervals between points are short. When proper imputation strategy is applied, the input to LSTM is very dense and boosts the performance.

Methods	Accuracy
Meta-LSTM + RNN-impute	0.647 ± 0.013
Meta-LSTM + RNN-decay	0.673 ± 0.004
Meta-LSTM + RNN-VAE	0.563 ± 0.027
Meta-LSTM+ ODE-RNN	**0.737 ± 0.007**
Meta-LSTM+ Latent-ODE	0.719 ± 0.008

Table 4.3: 5-shot 5-class experiments on MIMIC-III

4.5 Disease classification in few-shot learning setting

Finally, we construct a few-shot learning sub-dataset from MIMIC-III to validate the efficiency of both our meta-learner and base-learner. In our experiment, we performed 5-shot 5-class experiments on MIMIC-III data using the adjusted meta-LSTM as our meta-learner. The query set in each episode contains 15 examples. We achieved 0.737 accuracy with a standard deviation of 0.007. We also compared the performance against other base-learners under this low data regime. The results in table 4.3 show that meta-LSTM and ODE-RNN performed very well, even with very limited training data.

CHAPTER 5
DISCUSSION

5.1 Future work

In this work, we built a meta-learning model using Latent-ODE as the base-learner and meta-LSTM as the meta-learner. We primarily focused on phenotyping acute diseases within the MIMIC-III dataset and optimizing the model based on the base-learner. Therefore, the meta-learner design can be further investigated using neural-ODE. Furthermore, tasks beyond phenotyping can be explored on MIMIC-III dataset using this model. The scope of this project can also be expanded to other health datasets outside ICU setting, such as MRI images and more. The construction of few-shot learning setting may not be very practical in real-world clinical settings. A more appropriate usage of meta-learning in medicine could be rare disease classification, where we could construct a meta-learning training scheme and put rare disease in meta-test phase.

CHAPTER 6
CONCLUSION

In this project, we explored meta-learning and its applications in medicine. We selected an optimization-based meta-learning approach which gave us flexibility in designing our base-learner according to a specific task or problem. In our case, we investigated a neural-ODE based base-learner on EHR data. This data is recorded sporadically, and missing values are common. Furthermore, such properties of the dat are one of the major challenges for traditional neural

networks. Our results have shown that neural-ODE could achieve accurate disease classification even without employing imputation strategy, which is commonly used in traditional methods. Not only it simplifies the prediction, but also makes our model more explainable.

BIBLIOGRAPHY

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

[2] Riccardo Bellazzi and Blaz Zupan. Predictive data mining in clinical medicine : Current issues and guidelines. 7:81–97, 2006.

[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 737–744. Morgan-Kaufmann, 1994.

[4] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2018.

[5] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-sojo, and Jimeng Sun. Multi-layer Representation Learning for Medical Concepts. 2016.

[6] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun. GRAM: Graph-based Attention Model for Healthcare Representation Learning. pages 1–15, 2016.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. 2017.

[8] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, and Greg Ver Steeg. Multitask Learning and Benchmarking with Clinical Time Series Data. pages 1–19, 2018.

[9] Alistair E W Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Data Descriptor : MIMIC-III , a freely accessible critical care database. pages 1–9, 2016.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[11] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

[12] Jeong Min Lee. Diagnosis Code Prediction from Electronic Health Records as Multilabel Text Classification : A Survey. 2017.

[13] Michael C Mozer, Denis Kazakov, and Robert V Lindsey. Discrete event, continuous time rnns. *arXiv preprint arXiv:1710.04110*, 2017.

[14] Tsendsuren Munkhdalai and Hong Yu. Meta networks, 2017.

[15] Sachin Ravi and Hugo Larochelle. O m f -s l. pages 1–11, 2017.

[16] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[17] Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent odes for irregularly-sampled time series, 2019.

[18] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1842–1850. JMLR.org, 2016.

[19] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pages 245–248. IEEE, 2012.

[20] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.