FLEXIBLE MODELS FOR HETEROGENEOUS BIOMEDICAL DATA

A Dissertation

by

LIDA ZHANG

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Bobak J. Mortazavi
Committee Members,	Shuiwang Ji
	Theodora Chaspari
	Xiaoning Qian
Head of Department,	Scott Schaefer

August 2023

Major Subject: Computer Science

ABSTRACT

With the development of biomedical sensing techniques and data storage, machine learning has been widely applied to many healthcare applications from the abundance of data resources. However, biomedical data, from real-world applications, has the nature of heterogeneity, and this heterogeneity has not been comprehensively considered and successfully addressed. The heterogeneity in biomedical data includes the various data distributions, the irregularly sampled time-series data, the variation in the time domain, and other heterogeneous factors such as uncertain labeling. These different types of heterogeneity can happen individually or simultaneously, and sometimes a type of heterogeneity can trigger another one, for instance, a patient's health condition changed over time, and the doctors made adjustments to the measurements and treatments which causes the irregular feature sampling. Facing the challenge of heterogeneous data, a generalized may have decent performance on average, but fails in certain cases, which should not be ignored in the clinic. In addition, when building individual models for each group of homogeneous data, the training data can become limited, even with a large data size in total. For example, there are a great number of medications, but each of them may not have enough data. The limitation of the generalized models and the possible shortage of training data make the data heterogeneity a very challenging problem to address. Therefore, flexible models are demanded for the various types of heterogeneous biomedical data in real-world applications.

This dissertation investigates data heterogeneity and builds flexible models in biomedical data by focusing on different levels of heterogeneity: different types of heterogeneity happening individually, multi-source simultaneous heterogeneity, multiple data modalities on the same task, and clinical translation of data heterogeneity. We start by building different adaptive models for each individual heterogeneity on a certain type of biomedical data, focusing on time series, and then addressing a more complex situation of simultaneous heterogeneity. Next, the problem setting is extended from time-series data only to multiple data modalities, and finally, we introduce a clinical translation model trying to understand the data heterogeneity. Based on the focus on the hetero-

geneity in each type of data, transfer learning, adversarial training, and meta-learning techniques are proposed and applied to build adaptive models.

DEDICATION

To Mom and Dad, for always being supportive.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to all those who have contributed to my academic and personal growth over the past four years.

First and foremost, I extend my sincerest appreciation to my mentor, Dr. Bobak Mortazavi. Your guidance and support have been invaluable to me. Thank you for creating a conducive environment for learning and for being an exceptional mentor. I look forward to many more years of fruitful collaborations with you.

I would also like to express my gratitude to my labmates, both past and present, for their invaluable contributions to my research. Your dedication, enthusiasm, and support have made the journey much more enjoyable. I have learned a lot from each one of you, and I hope that I have been able to reciprocate in some way.

To my friends and colleagues in the PhD program, and my friend Dr. Moreno, thank you for your unwavering support, kindness, and friendship. You have been there for me through thick and thin, and I am honored to have you as my friends. Thank you for your generosity, hospitality, and countless acts of kindness.

Lastly, I would like to express my heartfelt thanks to my family for their unconditional love, support, and encouragement. Your constant support has been my anchor, and I am deeply grateful for everything you have done for me. Thank you.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Bobak Mortazavi [advisor], Professor Shuiwang Ji and Professor Theodora Chaspari of the Department of Computer Science and Engineering, and Professor Xiaoning Qian of the Department of Electrical and Computer Engineering.

The data analyzed for Chapter 2 was supported by Professor Zhangyang "Atlas" Wang, Dr. Xiaohan Chen, Dr. Tianlong Chen, and Dr. Nathan Hurley.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported in part by the NSF Engineering Research Center for Precise Advanced Technologies and Health Systems for Underserved Populations under NSF Award 1648451.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION AND LITERATURE REVIEW

Healthcare is hugely important for the functioning of society, which is underscored by the fact that it comprised over 18% of the gross domestic product (GDP) of the United States per year [1], motivating the research and data analytics in this area. With the development of wearable devices [2, 3] and health information systems [4], biomedical data becomes ubiquitous in the past few years, providing a great resource for health monitoring and analytics. For example, a smartwatch can generate millions of basic vital sign data points on average each day for each user [5], and the smartwatch shipment is estimated at 68.6 million units per year and is expected to reach 157.2 million by 2026 [6]. With the abundance of data, machine learning (ML) has been successfully applied in various healthcare applications, such as emotion detection and recognition [7, 8, 9], human activity recognition [10, 11, 12], and patient risk prediction [13].

Biomedical data includes biological data and medical data. As real-world datasets, biomedical data has the nature of heterogeneity, including the various data distributions, the irregular sampled time-series data, the time-domain variation, as well as other effects such as treatment for patients in the hospital. The heterogeneous data is a big challenge for machine learning modeling, for example, the similar shape of bio-signal can refer to different blood pressure values (both diastolic and systolic) for different subjects and machine learning models can be confused by these heterogeneous data. In a preliminary experiment, we aim to build a blood pressure regression model from multiple subjects with limited bio-signal training data, because not every subject can stay in the clinic for a long time for data collection to obtain enough labeled data for supervised learning. A general model trained from multiple subjects or fine-tuning to each subject both has a root mean square error (RMSE) of over 10 mmHg, which does not meet ISO standards[1]

The data heterogeneity happens in different types of biomedical data. With the development of smartphone, watch, and wearable devices, bio-signals become good resources for health monitor-

[1]ISO standards need to be met for multiple cohorts which should be representative of different populations. In this work, only one cohort is studied. For the sake of brevity when referring to ISO standards we refer specifically to the studied cohort and not to future cohorts.

ing. For example, Zhang et al. proposed a promising heart rate monitoring method from bio-sensor [14], and King et al. [9] further extend the work to detect stress for pregnancy. Utilizing ECG and PPG signals to predict blood pressure is an active area of research [15, 16]. There are a variety of techniques that have recently been investigated for their potential application to cuffless blood pressure estimation. Chief among those techniques include photoplethysmography (PPG) in conjunction with electrocardiography (ECG) [17], dual PPGs [18, 19], Doppler radar technology [20], or bioimpedance [21]. Each of these techniques attempts to measure the pulse transit time (PTT) or pulse wave velocity (PWV), both of which are known surrogates for blood pressure [22, 23, 16].

One of the major challenges in biological data heterogeneity is the various data distributions among subjects. This heterogeneity might be caused by demographic factors, body shapes, health conditions, or even weather. With this data heterogeneity, the model training can be confused, such as similar signal shapes with different labels or different signals with similar ground truth, especially when the training data is limited to data collection, as our previous example shows that a general model does not meet ISO standard. Some adaptive learning techniques have been proposed and applied for the various data distributions, such as unsupervised representation learning [24] and real-time adaptive learning [25]. However, these methods still require a certain number of data for adaptation. An additional problem with the limited training data is the corresponding limited variation with the potential generalization problem in training data. For example, if the diastolic blood pressure is between 65 to 80 mmHg, the trained model will have difficulty in some special cases like 100 mmHg. With the heterogeneous data distributions among subjects, it is hard to transfer knowledge from other people to address this problem. Therefore, it is very important to build models addressing heterogeneous data distributions. Adversarial learning provides a good strategy for learning regarding the data heterogeneity [26]. With an additional domain classification network and the reversed gradient from it, the feature extractor can be trained to obtain the general information from multiple domains, without focusing on any domain-specific information, and therefore a trained model can be easily adapted to another domain with different data distribution.

An electronic health record (EHR) is a collection of patient information during hospital visits. An EHR stores patient demographic information, records diagnoses, links laboratory test results, stores medication information, and more. EHRs are widely used in hospitals with an 86% adoption rate [27], which provides a rich resource for applying machine learning techniques on clinical outcome analysis, such as cardiovascular disorder diagnosis [28], phenotyping (the presentation of diseases) [29, 30, 31], or prediction of onset of adverse events, such as septic shock [32]. MIMIC-III [4] encourages studies in machine learning for healthcare because of its public availability, such as the MIMIC benchmark from [33]. Xu et al. extend the MIMIC benchmark by combining with MIMIC waveform data including continuous monitoring data (e.g. electrocardiogram data) and evaluate performance on two of the tasks: Decompensation and Length of Stay [34]. Song et al. [35] develop SAnD by replacing the LSTM-based multitask learning model with Transformer [36], and introduce a dense interpolation layer to incorporate the temporal order. SAnD has similar results to the LSTM-based benchmark without applying a recurrent network (0.01 is the upper bound on improvement in SAnD over the MIMIC Benchmark).

EHRs are complex datasets where distinguishing essential data from trivial data relies on individual patient context and disease state. What may be essential in the care of one patient may have no bearing on the outcome of another. These differences drive the heterogeneic nature of the EHR data, and the sparse and limited data does not support building personalized models like biosensing data. This heterogeneity is a challenge in machine learning and deep learning applications and is rarely addressed in existing work [33, 35, 37, 34]. While these models remain accurate, improving upon them will require addressing variable situations from data heterogeneity. For the irregularly sampled EHR data, Luo et al. proposed a new imputation method with filling the missing data with Generative Adversarial Network [38], and Shukla et al. map the irregular time-series data to a regular space with attention mechanism [39]. However, these methods only focus on how to generate a regular space, and do not further explore the reason for the EHR data missingness. A preliminary experiment shows that Shukla's work works well for some data missingness distributions but performs horribly for some others with AUCROC below 0.68. The feature sparsity

in EHRs is, perhaps, a more informative aspect, because various feature distributions may arise from diagnoses, examinations, and treatment decisions that stem from varying states and health conditions.

In addition to feature space heterogeneity, time-domain variation is also an important aspect of biomedical data heterogeneity, including various lengths of data and the change of values over time. To accommodate this complexity and heterogeneity, models need to be built in a way that is able to adapt to rapidly changing conditions within the ICU. Models need to be able to respond one way when the patient is lacking in data, and in another way as additional data is obtained. Training models with the flexibility to respond well in a high variety of situations is difficult. In order to achieve the goal of model adaptation on heterogeneous data, transfer learning provides a solution by using domain data to fine-tune a previously trained model. However, there are two weaknesses of this approach: first, the pre-trained model does not always benefit from the data from a particular domain, especially when the data amount is small, which may even lead the model to a worse situation. Second, transfer learning requires repeated training and thus does not meet the requirement of dynamic adaptation. Different from transfer learning that focuses on optimizing the model parameter, meta-learning learns from multiple domains, which provides a potential solution to the challenge of heterogeneity in EHR data resulting from the high variance in ICU stay. Finn et al. [40] proposed Model-Agnostic Meta-Learning (MAML) to optimize model initialization for multiple tasks, enabling rapid adaptation to specific tasks. Meta-learning has been successfully applied in the medical field by Zhang et al. [41] for the purpose of transfer learning between different disease onset estimations when a dataset contains minimal outcomes for a given condition. However, to the best of our knowledge, meta-learning has not been adopted to develop models that account for heterogeneity in medical datasets as a result of the different available features and timing of that availability within the course of a hospital admission.

Heterogeneity occurs frequently and can be complex across several dimensions, including features, labels, and the time-varying nature of data. The different types of heterogeneity can occur not only individually but also simultaneously, and thus result in a problem of multi-source hetero-

geneity in time-series modeling and applications. Often, the heterogeneous features are handled by training individual models for each subset of data [42], but this requires onerous training of multiple models and may result in poorly performing models if the same subsets have very limited data. Transfer learning and meta-learning are approaches used to aid this limitation across models [43, 44], and can significantly reduce the training time while maintaining performance. However, these techniques are not sufficient for multi-source heterogeneity. Semi-supervised learning algorithms have been developed using self-training [45, 46, 47, 48], pseudo-labeling [49], and a combination of consistency regularization [50, 51, 52, 47]. Semi-supervised learning including active learning is proposed to address the labeling challenge, and is applied in the time-series field [53, 54, 55, 56], but lacks consideration of various types or frequencies of data. Meta-learning is then applied in FixMatch as a new semi-supervised learning approach [57, 58], however, is only applied between the labeled and unlabeled data of the same prediction task, which dose not solve the multi-source heterogeneity problem. Recurrent networks and attention-based transformer model [36] are used to capture the time-domain variation, but a generalized version of these models is static and restrictive across types of data. Methods that address the simultaneous multi-source heterogeneity occurring in time-series data are needed.

We seek to solve the multi-source data heterogeneity challenge in applications in medicine, one of the most complex time-series data types with all three types of data heterogeneity. First, medical data contain thousands of different observations, laboratory tests, medications, etc. from hospitals [4], and the frequency (and category) of these measurements comes from doctors' examinations and implies the potential health condition. Learning from the similar frequency of medical data can lead the model to be more specific for a type of patients, so that risk prediction tasks can be improved and aid in up-to-date clinical decision-making. Second, as a real-world time-series dataset, medical data also has the challenge of obtaining labels. For example, the diagnosis from doctors is time-sensitive, and the development of patients' health conditions can cause changes in the labels. The development of patients' health conditions also raises the third heterogeneity, time-domain variation. In addition, this variation can also be caused by other factors, such as receiving

treatments in hospital [59], hospital transfer [60], ICU admission and release [61], etc. Facing these challenges, we propose a semi-supervised meta-learning algorithm for the heterogeneous features and uncertainty in labels. A discriminator is introduced for adversarial training to improve the model generalization. Regarding the variation over time, we propose a time-domain variation (TDV) framework applying transfer learning and our SSML. Our approach is a new connection between meta-learning, transfer learning, and semi-supervised learning. We test our approaches on two real-world medical datasets, PhynioNet Challenge 2012 and the MIMIC-III ICU dataset.

Data heterogeneity can not only happen within a type of data but also in multiple modalities of data. A single modality of data can sometimes be limited by some uncertain information. For example, continuous glucose monitors (CGMs) have been applied in diet monitoring and macronutrient prediction [62, 63], however, several factors can influence an individual's glucose response, such as the type of food consumed, as evidenced by significant variations in glucose response to foods with similar amounts of carbohydrate, protein, and fat [64]. Specifically, carbohydrates tend to raise glucose levels very quickly and then decrease rapidly, while fat has the lowest effectiveness but provides a longer-lasting effect. Additionally, an individual's health status is also an important factor, as those with untreated diabetes typically experience higher glucose levels after a meal than those without this condition. Similarly, the efficiency of image data can be influenced by the cooking style, types of sauces, etc. Therefore, we propose a model addressing the heterogeneity in data modality, introducing a projector using the late fusion mechanism to aggregate the extracted information from different modalities of data.

In the clinic, patients' health conditions are very complex and hard to interpret. The complexity of the health conditions of hospitalized patients has led to the development of personalized models [65, 66], and Oikonomou et al. proposed a phenomapping strategy that leverages information from all trial participants to phenotype individuals [67], however, personalized models are limited in available training data, and even with the assistance of transfer learning, it is still not optimal to train multiple models for each patient. Meta-learning [40] has been applied to EHR-based risk prediction models with limited training data to create fewer general models that apply across the

varied personal settings[41], but these methods pre-define each patient into one certain domain, and ignores patients' known or unknown health conditions that may result in potential cross-domain patients. Snell et al. proposed prototypical networks with a linear reinterpretation model [68] and Boniolo et al. built prediction models through patient similarity to address this limitation of meta-learning [69]; however, they do not have representative prototypes and flexible alignments for the heterogeneous patients' health conditions. With the meta-learning-based training approach for the prediction models of multiple prototypes, it is still not clear what these prototypes are. Crabbe et al. introduced a latent space explaining selecting some patients as prototypes and calculating the similarity between a new patient and these prototypes [70], but is not clear how are the prototypes selected and if they are representative. Inspired by these works, we introduce meta-prototype networks to develop risk prediction models by leveraging patient heterogeneity through trainable prototypes, representations of the heterogeneous patient conditions, rather than selecting against it, and at the same time, we use our proposed model as an interpretation to understand patients' health conditions.

1.1 Research Goals

The purpose of this dissertation is to propose and implement flexible models to address the heterogeneity in biomedical data. The goals are divided into four aims according to the different levels of addressing data heterogeneity: adaptive models for individual heterogeneity, multi-source heterogeneity, multi-model for multi-modality data, and clinical heterogeneity translation. A brief introduction to the three aims is as below.

Aim 1: Individual Heterogeneity. To address the heterogeneous biomedical data, we first build adaptive models for each data heterogeneity individually. We will discuss our proposed work for the shifting bio-sensing data distributions among subjects under regular feature space, and then focus on the irregular feature space by analyzing its sparsity and frequency instead of how to map irregular time-series data to a regular space. In addition, we will introduce our flexible models for the time-domain variation on time-series EHR data, attempting to adapt a model for each specific duration of an intensive care unit (ICU) stay.

Aim 2: Multi-source Heterogeneity. In time-series biomedical data, time-domain variation is another important aspect of heterogeneity. Time-domain variation includes the various lengths of sequences and the changes of value over time. We will introduce our flexible models for the various length of time-series EHR data, attempting to adapt a model for each specific duration of an intensive care unit (ICU) stay. To better understand what is changing over time and what changes cause significant effects on the final output, we plan to interpret the time-series modeling.

Aim 3: Multiple Data Modalities. In real-world applications, there can be multiple modalities of data. Using one modality of data for modeling can have limitations sometimes. For example, the macronutrients prediction of a meal from people's glucose response can be influenced by their biographic information and health condition, or even the types of food people eat. Therefore, it's important to introduce other resources of data in modeling. We will introduce how a prediction model can be built from multiple modalities of data.

Aim 4: Clinical Translation. In the clinic, it's important to understand the health condition of a patient. Clustering is a way to find which group a patient belongs to, however, sometimes a patient can be at the boundary of multiple, and it's hard to calculate the percentage of the various clusters. More importantly, the clustering result is hard to interpret. There isn't a clear way to understand what does each cluster mean. We propose a disease-based prototype meta-learning model for clinical translation. While adapting models to each disease, we also introduce a prototype network to understand the similarity of each prototype. These similarity scores can help understand what diseases a patient may have, and can also make a joint prediction from their corresponding models.

2. ADAPTIVE MODELS FOR INDIVIDUAL DATA HETEROGENEITY

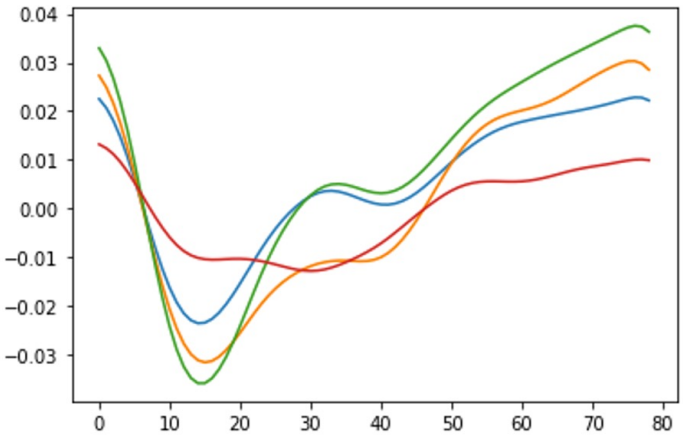
There are many types of data heterogeneity in real-world applications. In this chapter, we focus on three common heterogeneity in time-series data: Heterogeneous data distribution among subjects, irregular feature space, and time domain variation. We address each of them individually in the following three sections.

2.1 Heterogeneous Data Distribution

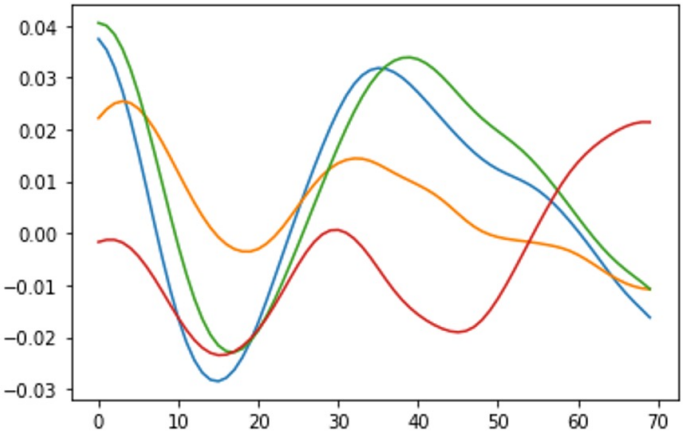
2.1.1 Subject Variation

The heterogeneous data distributions among subjects are a very common problem for bio-sensing data. For example, in a cuffless blood pressure estimation dataset, the DBP ranges from 50 to 100 mmHg and SBP ranges from 90 to 160 mmHg, however, for a specific subject the range is narrowed significantly to DBP 54.1 to 87.9 mmHg and SBP 96.2 to 146.2 mmHg. In addition, the signals are also very different from different subjects, including the signal shapes and lengths, for the similar ground truth blood pressure values, as Figure 2.1 shows. In a preliminary experiment, we trained a generalized blood pressure regression model across multiple subjects, following the multi-task framework in Figure 2.2 with two prediction tasks for diastolic and systolic blood pressure, and obtained the average RMSE values of over 10 mmHg for both tasks, which does not meet the ISO standard. Therefore, a generalized model often does not reach the performance goal [71].

To build a successful personalized deep neural network model, there is a need for a great number of training data. In the blood pressure regression modeling, Previous work on this dataset uses 80% of all available data, over 10 minutes on average, from each subject to train the personal models [23]. However, this calibration period is burdensome and the goal of an independent device should be to minimize the amount of calibration time required to improve utility and align with the clinical need. To that end, in this work, we investigate techniques to reduce the amount of data involved in training a model. Directly training an MTL model on reduced training data fails with errors exceeding ISO standards. Therefore, to meet the ISO standards (in this cohort) while min-



(a) Subject A



(b) Subject B

Figure 2.1: Bio-sensing data variation among subjects. The subjects have very similar ground truth blood pressure values (DBP 63 mmHg and SBP 117 mmHg for the left subject, and DBP 62 mmHG and SBP 118 mmHg for the right subject) but very different shapes and lengths of signals.

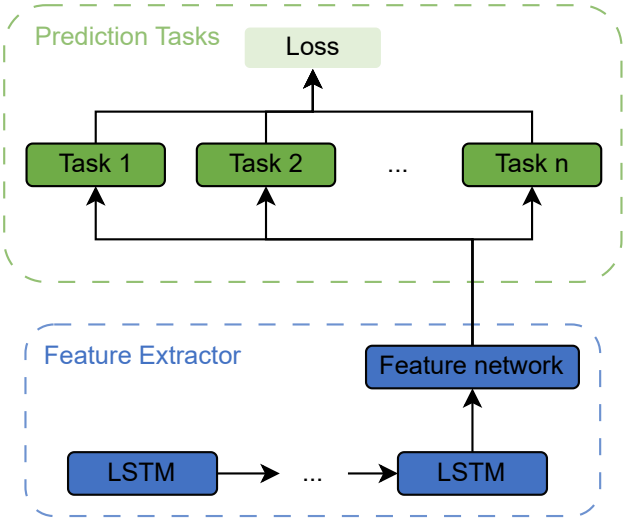


Figure 2.2: A LSTM-based generalized multi-task framework for time-series data.

imizing training data, we must utilize a technique to learn from other subjects. Transfer learning from a pretrained model is another solution, but the difference between subjects still impedes the learning process. Domain adaption [72, 73, 74] is one solution to cross-domain problems, and has recently been applied with deep learning techniques [75, 76] to minimize the maximum mean discrepancy distance between disparate outputs. Domain-adversarial neural networks (DANN) [77] allow for using adversarial training to extract domain-invariant features, allowing for rapid model adaptation with minimal training data.

2.1.2 Adaptive Model for Subject-independent Blood Pressure Regression [1]

2.1.2.1 Blood Pressure Regression

Hypertension is a worldwide chronic disease that causes an estimated 7.6 million deaths every year. The diagnosis of hypertension is usually based on clinical blood pressure readings, but the measurement of blood pressure outside of a clinical visit (also known as ambulatory blood pressure measurement) can provide better prognostic guidance than measurements during a routine clinic visit [78], due to well-known confounders such as masked hypertension [79], white coat hyperten-

[1]This section is from "Developing personalized models of blood pressure estimation from wearable sensors data using minimally-trained domain adversarial neural networks" by Zhang, Lida, Nathan C. Hurley, Bassem Ibrahim, Erica Spatz, Harlan M. Krumholz, Roozbeh Jafari, and Mortazavi J. Bobak.

sion [80], and nocturnal non-dipping hypertension [81]. Ambulatory blood pressure monitoring has been shown to be more predictive of cardiovascular mortality than clinical monitoring in a study of 63,910 adults [82], and nocturnal measurements are likely stronger predictors of cardiovascular risk than diurnal monitoring [83, 84, 85]. Therefore, increased ambulatory measuring is desirable for public health. However, on-market ambulatory monitoring devices are not appropriate for extensive use for a number of reasons: they require specific patient postures, they are obtrusive, they disrupt sleep, and they result in poor adherence. Cuffless blood pressure monitoring devices are desirable for their possibility to overcome each of those shortcomings. Cuffless blood pressure estimation techniques utilize devices to monitor surrogates of blood pressure, and use these surrogates to build regression models to estimate diastolic and systolic blood pressure.

There are a variety of techniques that have recently been investigated for their potential application to cuffless blood pressure estimation. Chief among those techniques include photoplethysmography (PPG) in conjunction with electrocardiography (ECG) [17, 14], dual PPGs [18, 19], Doppler radar technology [20], or bioimpedance [21]. Each of these techniques attempts to measure the pulse transit time (PTT) or pulse wave velocity (PWV), both of which are known surrogates for blood pressure [22, 23, 16]. Ibrahim et al. developed a bioimpedance-based sensor that locates arterial sites to measure these physiologic surrogates of blood pressure [23], and then used a window-based AdaBoost regression technique to measure personal diastolic and systolic blood pressure over windows of 10 consecutive beats to with respective errors of 2.6 mmHg and 3.4 mmHg. This finding falls within the ISO standard requiring errors less than 10 mmHg when comparing with a gold standard device [86] for the particular cohort. We first develop a deep multi-task learning (MTL) regression model using a version of the same dataset produced by Ibrahim et al. [23], but with an additional user. This model allows for more adaptable transfer learning than an AdaBoost regression model, and focuses on a beat-to-beat blood pressure estimation task as a new baseline.

Facing the challenge of heterogeneous data distributions and the limited training data, we propose a DANN-based MTL model to estimate beat-to-beat blood pressure for the goal of maintain-

ing accuracy within ISO standards while minimizing the amount of required training data [77]. To maximize clinical utility, we aim to train this model with a maximum of five minutes of training data for a new user. Our base model, an MTL blood pressure (BP) estimation model, is composed of a long short-term memory (LSTM) coupled to a shared dense layer to extract heartbeat features, and then two task-specific networks, one each for estimating diastolic and systolic blood pressure. When applying DANN, a domain (subject) classifier then attempts to classify a given beat as belonging to a particular subject. The adversarial training approach is then applied to this system with the goal of maximizing the performance of the BP estimator while minimizing the performance of the domain classifier. Throughout this process, the BP estimator is trained with reduced data from the new subject until convergence is achieved.

2.1.2.2 MTL BP Estimation Model

The base model consists of an LSTM layer, a shared dense layer, and two task-specific networks. The heartbeat data (and associated derived channels) are sent to an LSTM layer and then on to a shared dense layer. LSTM can memorize historical information, and therefore is applied to capture the patterns in the signal over time. The ability of an LSTM to retain historical information is valuable across the entirety of the heartbeat. We include a dropout layer following the LSTM. This layer allows the model to avoid overfitting and permits for some robustness to noise. Even if a part of the signal is corrupted, the model will still be able to perform with reasonable accuracy. We add a shared layer after the LSTM to further extract the relational information between channels. The extracted features are then passed on to the BP estimation network, consisting of two separate task-specific networks to estimate diastolic and systolic blood pressures. After each layer in these two task-specific networks, a dropout layer is applied to avoid overfitting. In order to build models for new subjects with reduced data, we further propose using DANN to transfer knowledge from other subjects and focus our attention on the beat-to-beat model as it provides higher potential clinical utility.

2.1.2.3 Adversarial Training with Minimal Data

With enough data from a subject, we are able to build a blood pressure regression model for that subject to within ISO standards. However, it is desirable to improve upon this and discover the minimal amount of training data that can provide for blood pressure while remaining within the ISO standard. For a device to be implemented in clinical practice, it should be widely adaptable to a variety of patients with minimal calibration time. Therefore, our objective here is to push the limits of training data utilized while remaining precise to the necessary standards.

When simply training with fewer data, the model quickly produces erroneous estimates that fall out of ISO standards after a small reduction in training data. To address this issue, we investigate transfer learning solutions to more rapidly adapt our model to a previously unknown subject. However, a chief challenge of model adaptation is that the difference in wearable sensor signal data between individuals is too large, and a single generalized model fails. Therefore, we need to learn from other subjects but discard the difference between subjects.

To build models for new subjects with reduced data, we utilize DANN to extract user-invariant features for the purpose of knowledge transfer. Figure 2.3 shows the implementation of DANN within our MTL model. Our DANN model has three key parts: a feature extractor, a BP estimator, and a domain classifier. The feature extractor and BP estimator are as described above: the feature extractor is an LSTM and the BP estimator is two task-specific networks. The domain classifier is described in detail below and serves as a new module that pushes learning of subject-agnostic features of the data.

We treat each subject as an individual domain. The domain classifier is trained to maximize its accuracy in recognizing to which subject a beat belongs. The BP estimator is trained to maximize the BP regression accuracy. The feature extractor is trained using each of these losses, but the gradient is reversed for the domain classification. This gradient reversal pushes the feature extractor to be blind to subjects, causing the extracted features to be subject-invariant. This coupling of the BP regression with reversed domain classification is the key adversarial component of this model. The BP estimator, domain classifier, and feature extractor are updated using back propagation as

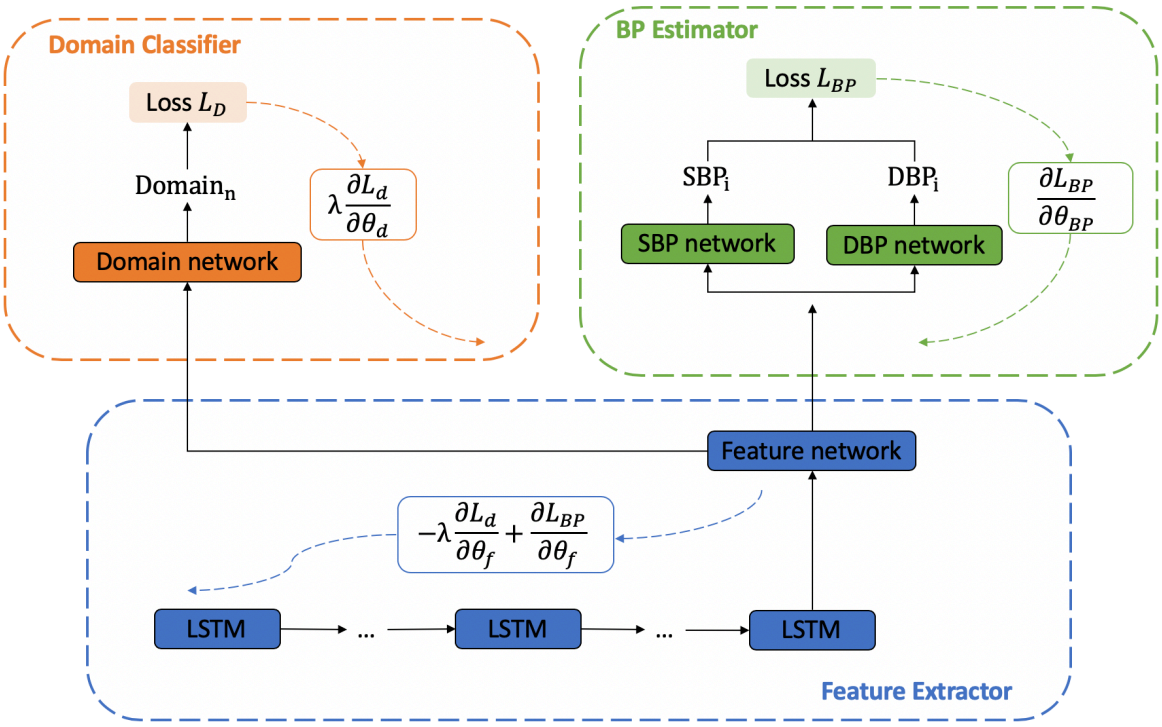


Figure 2.3: Adversarial training structure. There are three components: Feature extractor (blue), BP estimator (green), and Domain classifier (orange). The black solid lines represent data and arrows with dashed lines represent the Systolic and Diastolic loss, respectively, for gradient descent.

follows:

$$\theta_{BP} = \theta_{BP} + \alpha \cdot \frac{\partial L_{BP}}{\partial \theta_{BP}} \tag{2.1}$$

$$\theta_d = \theta_d + \alpha \cdot \lambda \cdot \frac{\partial L_d}{\partial \theta_d} \tag{2.2}$$

$$\theta_f = \theta_f + \alpha \cdot \left(-\lambda \cdot \frac{\partial L_d}{\partial \theta_f} + \frac{\partial L_{BP}}{\partial \theta_f} \right) \tag{2.3}$$

Here θ refers to the parameters in a model: θ_{BP}, θ_d and θ_f indicate the parameters in the BP estimator, domain classifier, and feature extractor, respectively. L_{BP} is the loss of the BP estimator and L_d is the cross-entropy loss from the domain classification. α is the learning rate, and λ is the loss weight, which balances the BP estimator and the domain classifier and is set to 1 in our

experiments. L_{BP} is given as

$$L_{BP} = \sum_i \left((E_i^S - T_i^S)^2 + (E_i^D - T_i^D)^2\right) \tag{2.4}$$

where E^D and E^S represent the estimated diastolic and systolic pressures, respectively, and T^S and T^D are their target values. This loss function ensures that both feature regression networks are related. Using an adversarial training approach, the feature extractor is trained to be blind to the source of the samples. Using DANN, we try to discriminate the difference between subjects, and lead the feature extractor to obtain common information that is related to blood pressure among different subjects, so that the new subject can learn from other subjects with greater training data. The loss function of domain classifier L_d is

$$L_d = \sum_i^N d_i log(p_i) \tag{2.5}$$

where p_i is the prediction of the domain, and d_i is the ground truth.

Initially, we use the new subject with reduced data as our target domain, and randomly choose another subject as the source domain. However, in this case, the domain classifier always predicts the domain to be the source. This results from the unbalanced data between source and target domains, and any actions to the domain classifier result in a decrease in temporal accuracy. Therefore, the domain classifier stays in the local minimum and can not be updated further. To solve this problem, we introduce a second training subject as the target domain which guides the network being trained toward the new subject. We select the subjects randomly because of a lack of feasible subject similarity metric. After training DANN to have stable loss, we use the reduced training data from the new subject again to retrain the model, converting the obtained knowledge from the other two subjects to align better with the new subject. Finally, we train a model under a leave-one-subject-out scheme where all other subjects are used to train the DANN. However, this approach does not converge and no usable results are produced.

2.1.3 Experiments

To test the model performance with the reduced data, we initially limit training data to three minutes for each subject, using the remaining data as the test set. Three minutes was selected as a length of time that would be feasible for in-clinic calibration of the blood pressure system. We first train the model directly without any pretrained model loaded or technique applied during training, so that we can understand the performance from the limited training data. Then, in order to learn from other subjects, we load the pretrained model with 80% training data and retrain the model with the reduced training set from the new subject. All layers of the pretrained model are retrained to adapt both the feature extraction and BP estimation functions to the new subject. For each subject, we test the pretraining approach from all the other subjects individually and calculate the average RMSE and correlation. To evaluate the DANN model, we need two other subjects as the source domain and the target domain for the adversarial training approach other than the new subject. These two subjects are randomly picked from all other subjects, and we run the test 10 times for each subject as a new subject for robustness. The average RMSE and correlation are calculated as well after the 10 rounds of testing. We use the same model structure and hyperparameters in these experiments: three layers of task-specific networks with hidden size 30, learned from manually trained MTL models. This work is implemented in Python 3.6 with Tensorflow 1.15, Numpy 1.18, sklearn 0.21. The average computation time is 8.5 ± 0.5 minutes per subject without additional parallelization or fine-tuning on our server of 2 Xeon 2.2GHz CPUs, 8 GTX 1080ti GPUs, and 528 GB RAM. Code for this implementation can be found at `https://github.com/stmilab/cufflessbp_dann`.

The results of training with three minutes are not sufficient to reach ISO standards with this model. Therefore, we repeat these experiments with four and five minutes of training data. After analysis of training with four minutes, the DANN model performs within the ISO standards of 85% of all diastolic and systolic data points having less than 10 mmHg absolute error within this cohort.

The results of utilizing only three minutes of training data are shown in table 2.1, and table 2.2

Table 2.1: Results using three minutes of subject-specific training data for diastolic and systolic blood pressure (DBP & SBP)

Subject		DANN		Pretrained		Directly Trained	
		RMSE	R	RMSE	R	RMSE	R
1	DBP:	$\mathbf{4.56 \pm 0.07}$	$\mathbf{0.43 \pm 0.05}$	4.93 ± 0.14	0.33 ± 0.07	4.93	0.16
	SBP:	$\mathbf{5.98 \pm 0.06}$	$\mathbf{0.25 \pm 0.03}$	6.19 ± 0.11	0.11 ± 0.05	12.88	0.00
2	DBP:	$\mathbf{5.39 \pm 0.12}$	$\mathbf{0.57 \pm 0.03}$	5.72 ± 0.14	0.47 ± 0.04	6.44	0.00
	SBP:	$\mathbf{8.45 \pm 0.20}$	$\mathbf{0.65 \pm 0.02}$	9.24 ± 0.30	0.55 ± 0.04	12.91	0.02
3	DBP:	$\mathbf{4.08 \pm 0.11}$	$\mathbf{0.40 \pm 0.02}$	4.22 ± 0.11	0.23 ± 0.11	13.65	0.00
	SBP:	$\mathbf{6.06 \pm 0.14}$	$\mathbf{0.50 \pm 0.03}$	6.81 ± 0.19	0.36 ± 0.10	7.41	0.00
4	DBP:	$\mathbf{4.21 \pm 0.05}$	$\mathbf{0.07 \pm 0.05}$	4.29 ± 0.16	0.02 ± 0.04	4.12	0.05
	SBP:	$\mathbf{7.63 \pm 0.03}$	$\mathbf{0.18 \pm 0.03}$	8.11 ± 0.21	0.16 ± 0.07	17.26	0.00
5	DBP:	$\mathbf{5.15 \pm 0.07}$	0.22 ± 0.06	5.52 ± 0.23	$\mathbf{0.23 \pm 0.04}$	5.61	0.20
	SBP:	$\mathbf{5.95 \pm 0.12}$	0.26 ± 0.09	6.22 ± 0.24	$\mathbf{0.28 \pm 0.02}$	6.02	0.30
6	DBP:	$\mathbf{6.25 \pm 0.09}$	$\mathbf{0.29 \pm 0.04}$	6.41 ± 0.18	0.23 ± 0.04	7.26	0.19
	SBP:	$\mathbf{7.59 \pm 0.13}$	$\mathbf{0.55 \pm 0.02}$	8.16 ± 0.24	0.46 ± 0.04	9.16	0.00
7	DBP:	$\mathbf{5.20 \pm 0.07}$	$\mathbf{0.29 \pm 0.05}$	5.60 ± 0.14	0.22 ± 0.05	6.09	0.25
	SBP:	$\mathbf{8.21 \pm 0.06}$	$\mathbf{0.37 \pm 0.07}$	8.76 ± 0.15	0.33 ± 0.05	8.89	0.00
8	DBP:	$\mathbf{5.50 \pm 0.11}$	$\mathbf{0.27 \pm 0.10}$	5.77 ± 0.13	0.24 ± 0.11	5.74	0.20
	SBP:	$\mathbf{12.06 \pm 0.24}$	$\mathbf{0.30 \pm 0.03}$	12.88 ± 0.54	0.30 ± 0.09	12.82	0.30
9	DBP:	$\mathbf{4.02 \pm 0.06}$	$\mathbf{0.34 \pm 0.02}$	4.22 ± 0.10	0.21 ± 0.05	4.72	0.21
	SBP:	$\mathbf{5.47 \pm 0.06}$	$\mathbf{0.18 \pm 0.08}$	5.81 ± 0.20	0.07 ± 0.04	5.56	0.00
10	DBP:	$\mathbf{4.23 \pm 0.01}$	$\mathbf{0.12 \pm 0.02}$	4.34 ± 0.08	0.12 ± 0.05	4.24	0.07
	SBP:	$\mathbf{5.86 \pm 0.02}$	$\mathbf{0.17 \pm 0.02}$	6.00 ± 0.10	0.12 ± 0.04	5.93	0.06
11	DBP:	$\mathbf{4.24 \pm 0.08}$	$\mathbf{0.51 \pm 0.02}$	4.61 ± 0.09	0.36 ± 0.06	4.44	0.49
	SBP:	$\mathbf{7.42 \pm 0.18}$	$\mathbf{0.51 \pm 0.03}$	8.14 ± 0.17	0.33 ± 0.07	8.47	0.00
Mean	DBP:	$\mathbf{4.80 \pm 0.74}$	$\mathbf{0.32 \pm 0.15}$	5.06 ± 0.78	0.24 ± 0.12	6.11 ± 2.56	0.16 ± 0.14
	SBP:	$\mathbf{7.34 \pm 1.88}$	$\mathbf{0.36 \pm 0.17}$	7.84 ± 2.06	0.28 ± 0.15	9.75 ± 3.57	0.06 ± 0.12

Table 2.2: Results using four minutes of subject-specific training data for diastolic and systolic blood pressure (DBP & SBP)

Subject		DANN		Pretrained		Directly Trained	
		RMSE	R	RMSE	R	RMSE	R
1	DBP:	$\mathbf{4.49 \pm 0.08}$	$\mathbf{0.45 \pm 0.03}$	4.81 ± 0.11	0.32 ± 0.08	5.10	0.37
	SBP:	$\mathbf{5.92 \pm 0.08}$	$\mathbf{0.26 \pm 0.05}$	6.12 ± 0.10	0.12 ± 0.05	6.20	0.18
2	DBP:	$\mathbf{5.32 \pm 0.11}$	$\mathbf{0.58 \pm 0.03}$	5.60 ± 0.15	0.51 ± 0.04	5.36	0.57
	SBP:	$\mathbf{8.19 \pm 0.29}$	$\mathbf{0.68 \pm 0.03}$	9.12 ± 0.30	0.58 ± 0.04	8.90	0.62
3	DBP:	$\mathbf{3.96 \pm 0.06}$	$\mathbf{0.42 \pm 0.03}$	4.16 ± 0.10	0.23 ± 0.12	4.18	0.36
	SBP:	$\mathbf{6.03 \pm 0.28}$	$\mathbf{0.57 \pm 0.05}$	6.60 ± 0.29	0.43 ± 0.09	7.30	0.00
4	DBP:	$\mathbf{4.06 \pm 0.06}$	$\mathbf{0.09 \pm 0.02}$	4.07 ± 0.05	0.06 ± 0.06	4.44	0.05
	SBP:	$\mathbf{7.68 \pm 0.14}$	$\mathbf{0.25 \pm 0.05}$	7.96 ± 0.21	0.17 ± 0.10	8.26	0.21
5	DBP:	$\mathbf{5.03 \pm 0.18}$	0.23 ± 0.25	5.01 ± 0.12	0.21 ± 0.04	5.08	$\mathbf{0.28}$
	SBP:	$\mathbf{5.77 \pm 0.09}$	$\mathbf{0.28 \pm 0.03}$	5.82 ± 0.14	0.26 ± 0.06	6.20	0.00
6	DBP:	$\mathbf{5.34 \pm 0.23}$	$\mathbf{0.33 \pm 0.09}$	5.74 ± 0.06	0.20 ± 0.04	5.32	0.30
	SBP:	$\mathbf{6.30 \pm 0.19}$	$\mathbf{0.63 \pm 0.03}$	7.46 ± 0.39	0.53 ± 0.07	8.47	0.39
7	DBP:	$\mathbf{5.17 \pm 0.10}$	0.33 ± 0.27	5.24 ± 0.11	0.26 ± 0.08	5.77	$\mathbf{0.29}$
	SBP:	$\mathbf{8.12 \pm 0.16}$	$\mathbf{0.43 \pm 0.04}$	8.41 ± 0.18	0.34 ± 0.06	9.01	0.40
8	DBP:	$\mathbf{5.34 \pm 0.12}$	$\mathbf{0.39 \pm 0.05}$	5.50 ± 0.15	0.34 ± 0.08	5.35	0.38
	SBP:	$\mathbf{11.62 \pm 0.34}$	$\mathbf{0.44 \pm 0.06}$	12.07 ± 0.25	0.38 ± 0.07	12.14	0.34
9	DBP:	$\mathbf{3.98 \pm 0.07}$	$\mathbf{0.33 \pm 0.07}$	4.18 ± 0.06	0.21 ± 0.06	4.15	0.28
	SBP:	$\mathbf{5.47 \pm 0.08}$	$\mathbf{0.24 \pm 0.04}$	5.68 ± 0.08	0.06 ± 0.04	5.68	0.13
10	DBP:	$\mathbf{4.19 \pm 0.03}$	0.12 ± 0.04	4.25 ± 0.07	$\mathbf{0.13 \pm 0.03}$	4.68	0.07
	SBP:	$\mathbf{5.83 \pm 0.02}$	$\mathbf{0.13 \pm 0.02}$	5.90 ± 0.09	0.13 ± 0.04	6.43	0.04
11	DBP:	$\mathbf{4.15 \pm 0.06}$	$\mathbf{0.52 \pm 0.06}$	4.54 ± 0.12	0.38 ± 0.08	4.56	0.38
	SBP:	$\mathbf{7.25 \pm 0.13}$	$\mathbf{0.52 \pm 0.02}$	7.99 ± 0.16	0.37 ± 0.08	8.07	0.42
Mean	DBP:	$\mathbf{4.64 \pm 0.60}$	$\mathbf{0.34 \pm 0.15}$	4.83 ± 0.62	0.26 ± 0.12	4.90 ± 0.53	0.31 ± 0.15
	SBP:	$\mathbf{7.10 \pm 1.79}$	$\mathbf{0.40 \pm 0.18}$	7.56 ± 1.90	0.31 ± 0.17	7.88 ± 1.84	0.25 ± 0.20

Table 2.3: Results using five minutes of subject-specific training data for diastolic and systolic blood pressure (DBP & SBP)

Subject		DANN		Pretrained		Directly Trained	
		RMSE	R	RMSE	R	RMSE	R
1	DBP:	$\mathbf{4.39 \pm 0.07}$	$\mathbf{0.45 \pm 0.04}$	4.79 ± 0.16	0.34 ± 0.09	4.87	0.37
	SBP:	$\mathbf{5.85 \pm 0.08}$	$\mathbf{0.38 \pm 0.03}$	6.04 ± 0.07	0.15 ± 0.06	6.03	0.00
2	DBP:	$\mathbf{5.26 \pm 0.07}$	$\mathbf{0.59 \pm 0.03}$	5.63 ± 0.10	0.51 ± 0.02	5.34	0.56
	SBP:	$\mathbf{7.98 \pm 0.17}$	$\mathbf{0.68 \pm 0.02}$	9.00 ± 0.15	0.60 ± 0.02	8.58	0.66
3	DBP:	$\mathbf{3.89 \pm 0.06}$	$\mathbf{0.44 \pm 0.03}$	4.13 ± 0.13	0.23 ± 0.15	3.90	0.44
	SBP:	$\mathbf{5.77 \pm 0.15}$	$\mathbf{0.58 \pm 0.02}$	6.26 ± 0.37	0.51 ± 0.08	7.19	0.00
4	DBP:	4.06 ± 0.04	$\mathbf{0.10 \pm 0.02}$	$\mathbf{4.04 \pm 0.06}$	0.03 ± 0.07	4.37	0.11
	SBP:	$\mathbf{7.69 \pm 0.06}$	$\mathbf{0.29 \pm 0.04}$	8.02 ± 0.18	0.20 ± 0.10	7.87	0.23
5	DBP:	$\mathbf{4.83 \pm 0.29}$	$\mathbf{0.26 \pm 0.07}$	4.87 ± 0.14	0.18 ± 0.10	5.11	0.25
	SBP:	$\mathbf{5.61 \pm 0.06}$	$\mathbf{0.27 \pm 0.04}$	5.85 ± 0.18	0.21 ± 0.08	6.05	0.19
6	-	-	-	-	-	-	-
	-	-	-	-	-	-	-
7	DBP:	$\mathbf{5.04 \pm 0.05}$	$\mathbf{0.36 \pm 0.02}$	5.21 ± 0.08	0.32 ± 0.03	5.61	0.30
	SBP:	$\mathbf{7.94 \pm 0.06}$	$\mathbf{0.47 \pm 0.02}$	8.28 ± 0.16	0.41 ± 0.04	8.69	0.43
8	DBP:	$\mathbf{5.27 \pm 0.16}$	$\mathbf{0.40 \pm 0.03}$	5.50 ± 0.21	0.34 ± 0.11	5.49	0.37
	SBP:	$\mathbf{10.83 \pm 0.39}$	$\mathbf{0.48 \pm 0.05}$	12.04 ± 0.57	0.39 ± 0.12	12.98	0.35
9	DBP:	$\mathbf{3.84 \pm 0.06}$	$\mathbf{0.34 \pm 0.03}$	4.04 ± 0.10	0.23 ± 0.07	4.29	0.24
	SBP:	$\mathbf{5.29 \pm 0.04}$	$\mathbf{0.29 \pm 0.03}$	5.53 ± 0.11	0.09 ± 0.03	5.63	0.03
10	DBP:	$\mathbf{4.20 \pm 0.02}$	0.13 ± 0.03	4.31 ± 0.09	0.11 ± 0.05	4.44	$\mathbf{0.18}$
	SBP:	$\mathbf{5.87 \pm 0.02}$	$\mathbf{0.18 \pm 0.04}$	5.98 ± 0.06	0.13 ± 0.05	6.32	0.16
11	DBP:	$\mathbf{4.02 \pm 0.05}$	0.53 ± 0.02	4.30 ± 0.16	0.48 ± 0.05	4.13	$\mathbf{0.54}$
	SBP:	$\mathbf{6.91 \pm 0.12}$	$\mathbf{0.54 \pm 0.02}$	7.62 ± 0.19	0.45 ± 0.05	8.48	0.48
Mean	DBP:	$\mathbf{4.48 \pm 0.57}$	$\mathbf{0.36 \pm 0.16}$	4.68 ± 0.60	0.28 ± 0.15	4.76 ± 0.58	0.33 ± 0.14
	SBP:	$\mathbf{6.79 \pm 1.70}$	$\mathbf{0.41 \pm 0.17}$	7.46 ± 2.00	0.31 ± 0.18	7.78 ± 2.05	0.25 ± 0.22

and table 2.3 show results utilizing four and five minutes of training data respectively. From the results, using three minutes of training data obtains an RMSE of 4.80 ± 0.74 mmHg for diastolic blood pressure and 7.34 ± 1.88 mmHg for systolic blood pressure. DANN improves RMSE over the pretrained model by 0.20 mmHg for diastolic blood pressure and 0.60 mmHg for systolic blood pressure. When utilizing four minutes of training data, the model obtains an RMSE of 4.64 ± 0.60 mmHg for diastolic blood pressure and 7.10 ± 1.79 mmHg for systolic blood pressure. DANN improves RMSE over the pretrained model by 0.19 mmHg for diastolic blood pressure and 0.46 mmHg for systolic blood pressure. For five minutes training data, DANN improves RMSE over the pretrained model by 0.26 mmHg to 4.48 mmHg for diastolic blood pressure, and improves by 0.67 mmHg to 6.79 mmHg for systolic blood pressure[2]. We also test the pretrained models on new users without any retraining process. The average RMSE for DBP is 6.94 mmHg and for SBP is 11.51 mmHg, and the average correlation for DBP is 0.07 and for SBP is -0.01.

Figures 2.4, 2.5, and 2.6 show the Bland-Altman plots for the DANN model trained with three, four, or five minutes of training data, respectively. In the three-minute model, 96.0% of predictions have a diastolic error less than 10 mmHg, however, only 84.5% of predictions have a systolic error less than 10 mmHg. This result is below the ISO standard and prompts repeating the experiment with five minutes of training data. In the four-minute and five-minute model, this error improves to be within the ISO standard: 96.2% diastolic error and 85.9% systolic error are less than 10 mmHg in the four-minute model, and 96.2% diastolic error and 85.5% systolic error are less than 10 mmHg in the five-minute model. The decrease from the four-minute model to the five-minute model might be the missing subject in the five-minute model.

Figure 2.7 is an example of estimated and target blood pressure with five minutes of training data applying DANN. Five minutes of training data can track the change of blood pressure, e.g. the systolic blood pressure in the figure. However, the reduced training data cannot always respond to changes in blood pressure, especially for cases with lower variability such as the estimation of diastolic blood pressure in the figure.

[2]Subject 6 has only five minutes of data in total, and so is excluded from analyses with five minutes training data.

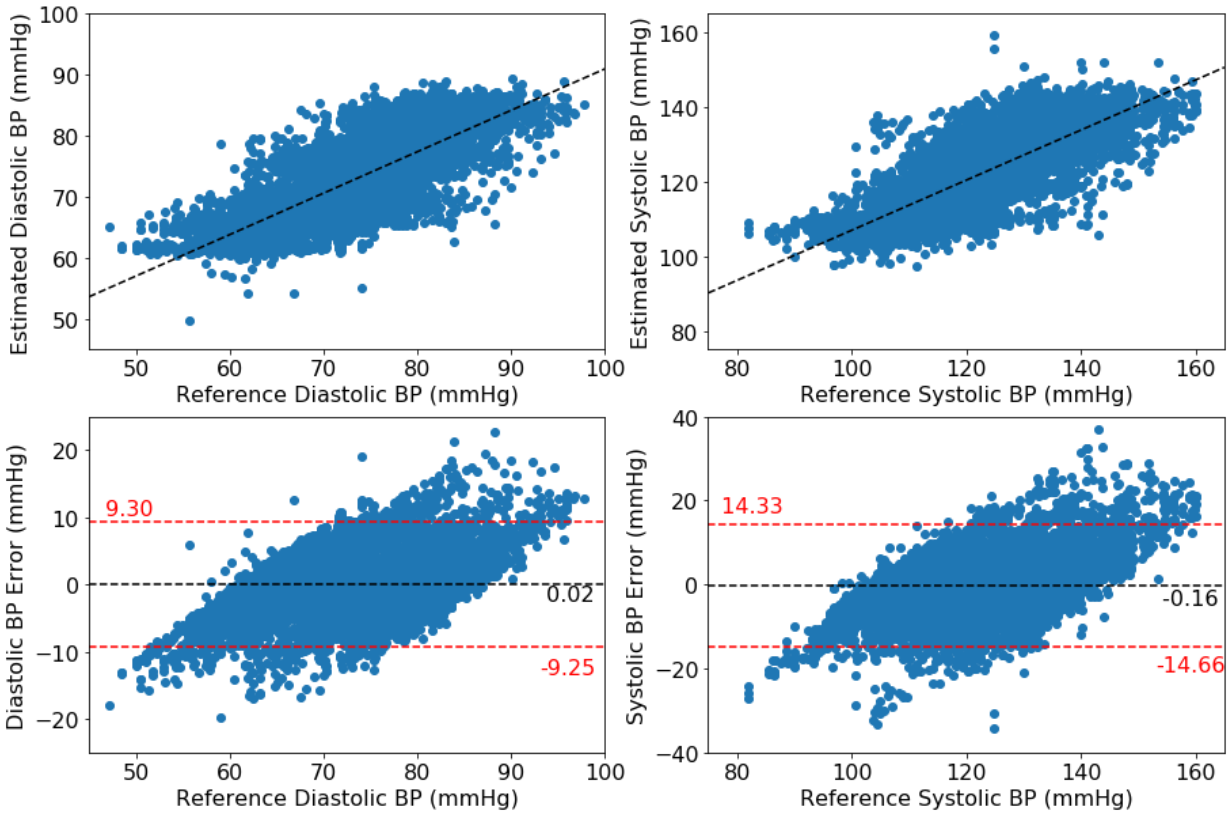


Figure 2.4: Bland-Altman plot for DANN model using three minutes of subject-specific training data

From these results, we observe that model performance decreases significantly when reducing the training data, and less training data results in much lower accuracy (higher RMSE). With three minutes of training data, the original MTL model without a pretrained model or the adversarial training process fails for many subjects. There are five subjects with RMSE over 10 mmHg, which is outside of the acceptable range for ISO standards in blood pressure. However, when training with a pretrained model from another subject, the model performance improves for both estimations. When applying the DANN-based training method, the RMSE further decreases, particularly for systolic blood pressure more so than for diastolic.

When training with four or five minutes of data, all three training approaches show an increase in performance. It is interesting to note that 10 out of 11 subjects obtain RMSE below 10 mmHg when directly training the MTL model without DANN. Compared to training with the pretrained

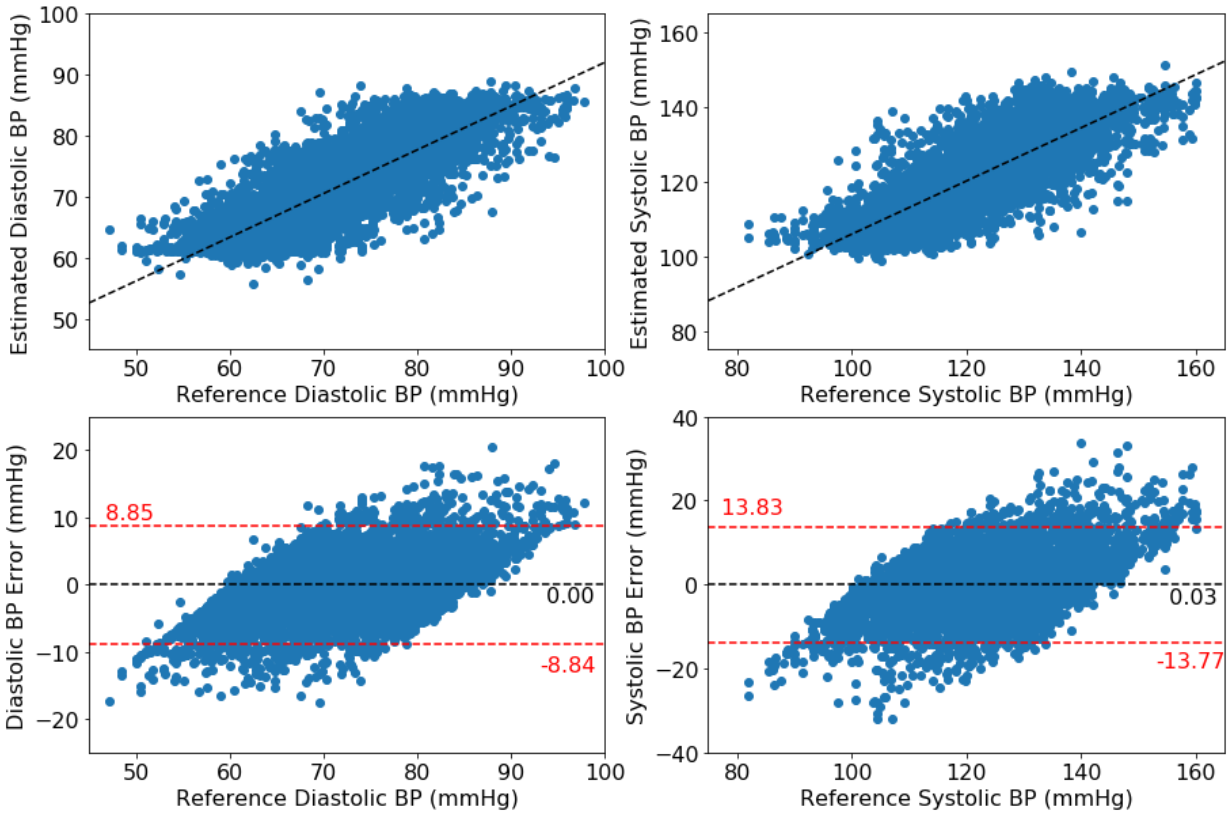


Figure 2.5: Bland-Altman plot for DANN model using four minutes of subject-specific training data

model and direct training approaches, our DANN-based model still has significant benefits. The DANN-based model has lower RMSE, lower standard deviation, and higher correlation, meaning that it performs better and more robustly for additional subjects. In comparison to direct training of the MTL model, both the DANN-based model and the pretrained model help improve the model performance, meaning that it is important to learn from other subjects when a new subject does not have enough training data. The advantage of the DANN-based model indicates that it is more useful to learn from other subjects and to discard the difference between subjects.

With less training data, the model tends to estimate blood pressure as closer to mean values, causing significant errors for extreme high and low blood pressure. When training the model with three minutes of data, the 85% absolute error for systolic blood pressure is 10.11 mmHg. The 85% error is greater than 10 mmHg of the ISO standard, even though it is already improved by applying

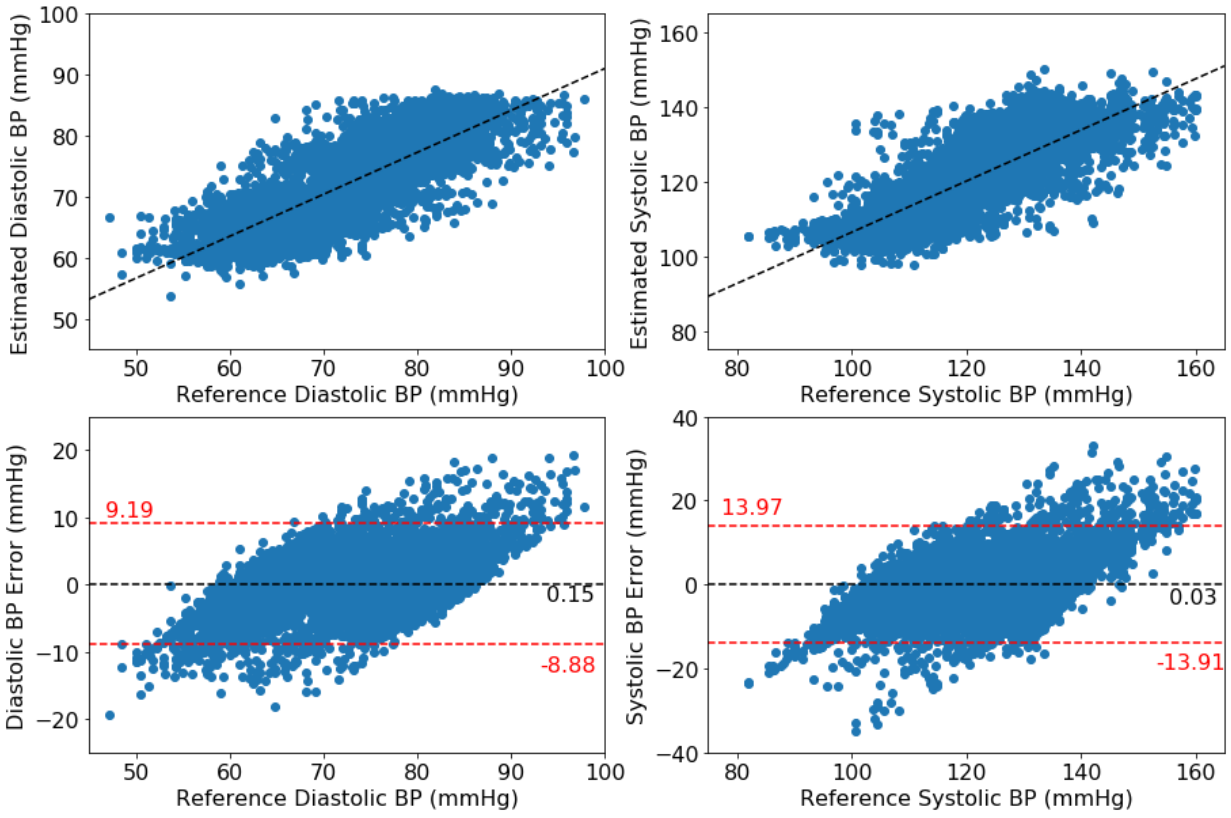


Figure 2.6: Bland-Altman plot for DANN model using five minutes of subject-specific training data

DANN. Then, we extend the training data to meet the ISO standard requirement. In this model, four minutes of training data is the minimum required amount of training data to obtain confident blood pressure estimations within ISO standards, and while maximizing clinical convenience for future use.

Model Performance Relative to ISO Standard

For the development of a blood pressure device, ISO standards require that 85% of measurements be within 10 mmHg of a standardized reference value for a given cohort. For each subject in our dataset, Table 2.4 reports the percentage of measurements that fall within this range for varying lengths (3 minutes, 4 minutes, or 5 minutes) of training data. The mean values are reported as well, showing that with 4 minutes of training data, 96.1% of DBP and 85.2% of SBP measurements fall within this range for the cohort studied. This framework presented allows for further adaptation of

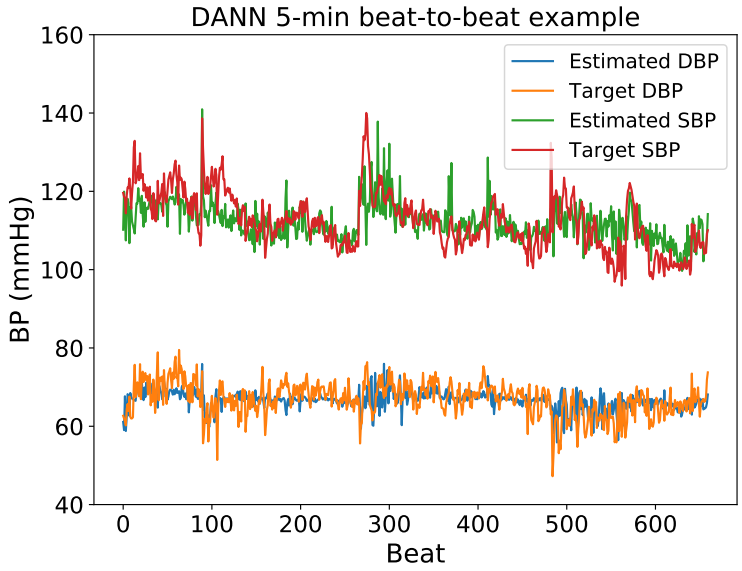


Figure 2.7: Estimated and target blood pressure plots from a subject. The estimation here is provided by the DANN model and trained with five minutes of training data. This plot is not completely representative: for some subjects with lower variability, the model does not respond to changes in blood pressure and instead predicts a near constant blood pressure.

DANN training times as data collection from future cohorts progresses.

Model Interpolation

To further evaluate DANN's ability to generate a general regression model, which may aid in future reduction of needed training data, we test the ability of the model to interpolate blood pressures in specific ranges that are intentionally withheld from training. For each individual we adapt the model to using DANN, we first remove from all samples with either diastolic or systolic blood pressure within a specific range (for example, systolic blood pressure from 120-125 mmHg) from the training set. We then repeat model training (using 4-minutes of training data) and test on the full, held out test set. This is analogous to the experiment with results recorded in Table 2.2 but with a different distribution of training data, reducing the ranges of blood pressures seen from the new individual.

After training, we test the model with the full test set, which includes blood pressure values from the test individual withheld from training. We report both overall RMSE for all test data and in-gap RMSE where the test data exclusive comes from the omitted blood pressure range.

Table 2.4: The percentage (%) of results from the DANN model that fall within 10 mmHg of the reference value. To meet ISO standards in a given cohort, at least 85% of measurements must fall within that range.

	3 mins		4 mins		5 mins	
Subject	DBP	SBP	DBP	SBP	DBP	SBP
1	95.7%	91.3%	95.8%	91.7%	95.9%	90.7%
2	94.4%	77.1%	93.8%	81.4%	91.9%	74.2%
3	98.4%	91.7%	97.4%	91.4%	98.3%	93.7%
4	98.2%	80.4%	98.7%	82.3%	98.8%	82.5%
5	90.0%	85.3%	94.0%	91.0%	92.3%	91.3%
6	92.7%	78.0%	96.4%	87.3%	-	-
7	95.2%	79.5%	94.7%	81.7%	95.4%	83.2%
8	92.7%	65.6%	92.5%	63.1%	94.2%	65.8%
9	99.7%	89.4%	99.6%	93.5%	99.8%	95.2%
10	96.6%	89.4%	97.1%	89.6%	97.2%	89.4%
11	96.9%	82.4%	97.5%	83.9%	97.2%	85.6%
Mean	95.5%	83.1%	96.1%	85.2%	96.1%	85.2%

Will illustrate model performance with diastolic gaps of 5 mmHg and systolic gaps of 6 mmHg. Specifically, we tested diastolic gaps of 55-60, 65-70, 70-75, 75-80, 80-85, and 85-90 mmHg, and systolic gaps of 90-96, 95-101, 100-106, 105-111, 110-116, 115-121, 120-126, 125-131, 130-136, 135-141, 140-146, and 145-151 mmHg. Due to variations between subjects, not all gaps were tested on all subjects. For instance, a subject whose systolic blood pressure never fell below 106 mmHg would not be included in a gap test for the systolic range of 100-106 mmHg. The overall RMSE and in-gap RMSEs were averaged over each subject, and those values are reported in Table 2.5. We test these gaps at intervals throughout the distribution of blood pressures present. We note that, even with the errors introduced from the missing values, DANN still outperforms the other models.

As seen in Table 2.5, the model error tends to increase slightly in the gaps of training data. This is expected given that this model is never trained on values from within those gaps. However, the mean of the error within the gap and overall is still small, showing that the model is able to successfully interpolate to unseen values.

Figure 2.8 further illustrates this finding, showing pooled test predictions for all users, including

Table 2.5: Model results when trained using gaps in training data. DBP gap size is 5 mmHg and SBP gap size is 6 mmHg. Results shown are averaged across varying gap locations as described in the text.

	DBP		SBP	
Subject	Overall RMSE	In-Gap RMSE	Overall RMSE	In-Gap RMSE
1	4.64	5.19	5.99	7.38
2	5.61	5.56	8.23	9.19
3	4.13	5.06	5.73	7.27
4	3.75	4.03	7.80	8.94
5	5.14	5.96	5.86	7.18
6	6.01	6.95	8.46	9.11
7	5.24	5.62	8.03	9.48
8	5.68	6.93	10.76	10.81
9	4.07	4.41	5.74	6.43
10	4.15	4.15	5.86	6.47
11	4.33	4.77	7.09	8.27
Mean	4.80 ± 0.74	5.33 ± 0.96	7.23 ± 1.60	8.23 ± 1.40

for diastolic gaps of 5 mmHg located at 70-75 mmHg. In these plots, the orange points represent samples that fell within the excluded range (in-gap samples) and the blue points represent samples from outside of the gaps. The left side of each figure shows the diastolic pressures, and the right side of each figure shows the systolic pressures. As can be seen, omitting a range of diastolic pressures does not clearly omit a range of systolic pressure, reflecting the lack of simple relationship between diastolic and systolic pressures.

Similarly, a plot of gaps in systolic blood pressures are shown in Figure 2.9 of 6 mmHg gaps located at 125-131 mmHg. As before, the orange points represent samples that fell within the excluded range (in-gap samples) and the blue points represent samples from outside of the gaps. The left side of each figure shows the diastolic pressures, and the right side of each figure shows the systolic pressures.

When a gap falls in the middle of the blood pressure distribution, both low and high blood pressures are equally trained, and when a gap falls at an extreme range, one side is trained well and the side with reduced data is trained poorly. This unbalanced training for an extreme gap results in higher difficulty of generalization. Therefore, the middle gap has fewer errors than gaps in low and

high blood pressure ranges and why the in-gap RMSE reported in Table 2.5 sees a slight increase compared to the overall RMSE.

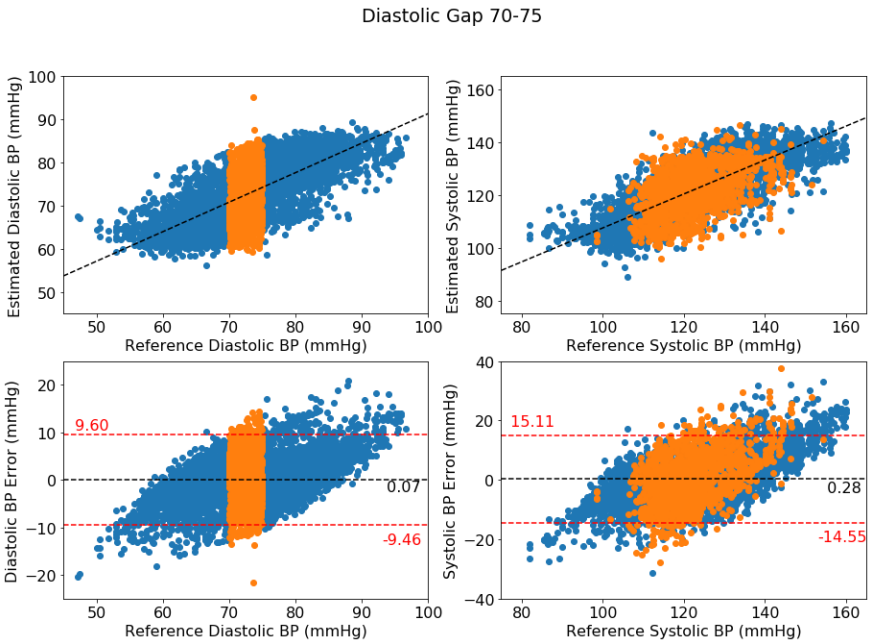


Figure 2.8: Bland-Altman plot for DANN model using four minutes with middle DBP gap

To further test the generalizability of this model, we explored the impact of other gap sizes. We tested diastolic gap sizes of 3 mmHg, 5 mmHg, 7 mmHg, and 10 mmHg and systolic gap sizes of 5 mmHg, 6 mmHg, 7 mmHg, and 10 mmHg. Results of this test on Subject 1 are shown in Table 2.6. As expected, increasing gap size results in higher RMSE for both overall and in-gap evaluations. In particular, this model struggles in generalizing across gaps of 7 mmHg or larger. For the samples tested here and the data length available, generalizations across this gap size appear to be ill-advised. While this paper centrally focuses on the duration of data needed, aiming to reduce data collection burdens on new users, additional work is needed to identify if further reductions are possible, including the type of blood pressure data needed, such as only low and high blood pressure values. While not an explicit goal of this work, DANN finds preliminary results indicating gaps are possible, and because of the distribution of training data available from

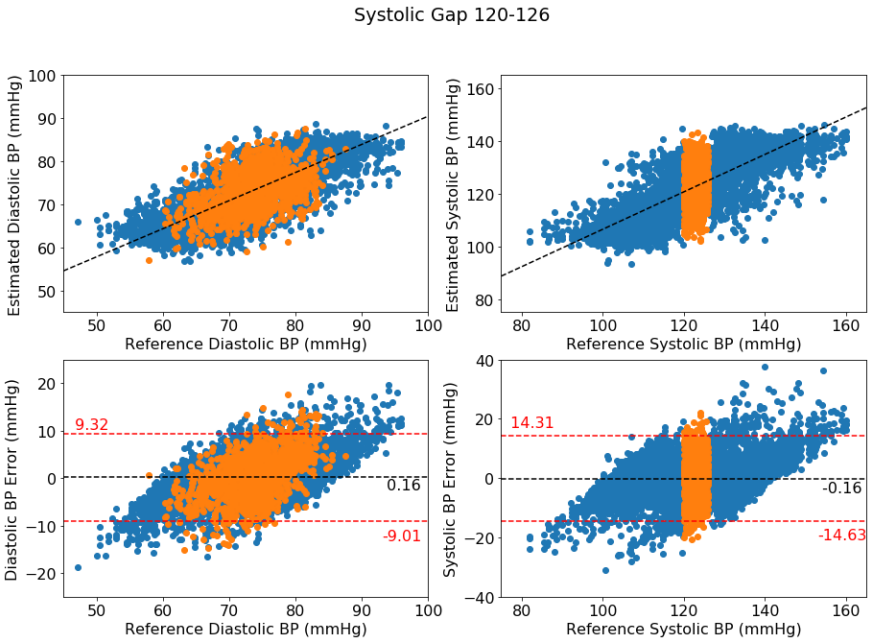


Figure 2.9: Bland-Altman plot for DANN model using four minutes with middle SBP gap

other subjects, indicates low and high blood pressure values are more important to provide for the new user than middle (normal) blood pressure values.

Table 2.6: Generalization results for varying gap sizes applied to Subject 1. As would be expected, increasing gap size results in poorer performance.

Gap Type	Gap Size	Overall RMSE	In-Gap RMSE
DBP	3	4.58	4.87
DBP	5	4.64	5.18
DBP	7	4.85	6.69
DBP	10	5.59	6.38
SBP	5	5.96	6.24
SBP	6	5.99	6.36
SBP	7	6.21	7.07
SBP	10	6.37	8.24

MTL Beat-to-Beat Performance Per Subject with 80% Training Data

While the primary focus of this work is on the performance of this model with the application

of DANN, we separately studied the performance of the isolated MTL model. For each subject, we split the data to be 80% as training set, 10% as validation set, and 10% as test set. We test the whole dataset without repetition from 10-fold cross-validation. This experiment shows overfitting: while the training set is modeled with high average correlation and low average RMSE, the test set suffers significantly in comparison. Performances on the test set are shown in Table 2.7. These values still provide a basis for modeling of blood pressure but demonstrate the need for more intelligently trained models, such as DANN in this work.

Table 2.7: MTL beat-to-beat performance per subject with 80% training data for diastolic and systolic blood pressure (DBP & SBP) RMSE (mmHg) and R.

Subject	DBP RMSE	SBP RMSE	DBP R	SBP R
1	4.40	5.84	0.48	0.29
2	5.40	8.55	0.54	0.63
3	3.95	5.86	0.42	0.58
4	4.14	7.55	0.11	0.33
5	5.29	5.73	0.23	0.27
6	6.15	8.16	0.25	0.49
7	4.94	7.84	0.40	0.48
8	5.30	10.93	0.40	0.50
9	3.70	5.49	0.42	0.22
10	4.06	5.65	0.25	0.23
11	4.30	7.18	0.47	0.54
Mean	4.69 ± 0.73	7.16 ± 1.68	0.36 ± 0.13	0.41 ± 0.15

2.2 Irregularly Sampled Time-series Clinical Data

Electronic health records (EHRs) provide large quantities of time-varying, real-world clinical data. Machine learning has increasingly focused on EHR data to provide clinical predictions for individual patients, which support decision-making for doctors [87, 88]. However, as real-world biomedical datasets, the challenge of heterogeneity also exists in EHRs. These heterogeneous EHR data may influence the estimation of patient risk of adverse events [89], and a generalized model is too static and restrictive across types of patients. Models capable of handling data heterogeneity in EHRs may improve risk prediction and aid in up-to-date clinical decision-making.

In clinical, despite the various data distributions, another important problem of heterogeneity is irregular data sampling. For waveform data, it can come from noise or data missing, similar to bio-sensing data. A common solution is noise detection and removal [14], which does not affect model building since the remaining data still have a consistent frequency. However, for vital data such as lab tests, the number of data points changes between features and over time, further, there is not even any certain length of the time interval between two consecutive records, because the tests can be run anytime in a day. The irregular sampling on the time-series EHR data thus causes difficulty in data preparation and preprocessing for model training and testing.

Imputation is a solution to address this problem by organizing the irregular data to a regular space and filling the missing values. Lipton et al. compared the different strategies of imputation and concluded that imputing zeros with the indicators of missing data works better than without indicators and other imputations including the previous value and average values [90]. Luo et al. proposed a new imputation method by using Generative Adversarial Networks to generate the data and fill the missing part of data from the generated data. Instead of imputation, Shukla et al. [39] address the irregularly-sampled data by mapping it to a regular space, but there is no specified analysis about each homogeneous set in the heterogeneous in EHRs. However, these works only focus on how to obtain a regular space of data and do not consider the underneath cause and consequence of the data missing. The feature space in EHRs is, perhaps, a more informative aspect, for meta-learning to apply to, because various feature distributions may arise from diagnoses,

examinations, and treatment decisions that stem from varying states and health conditions.

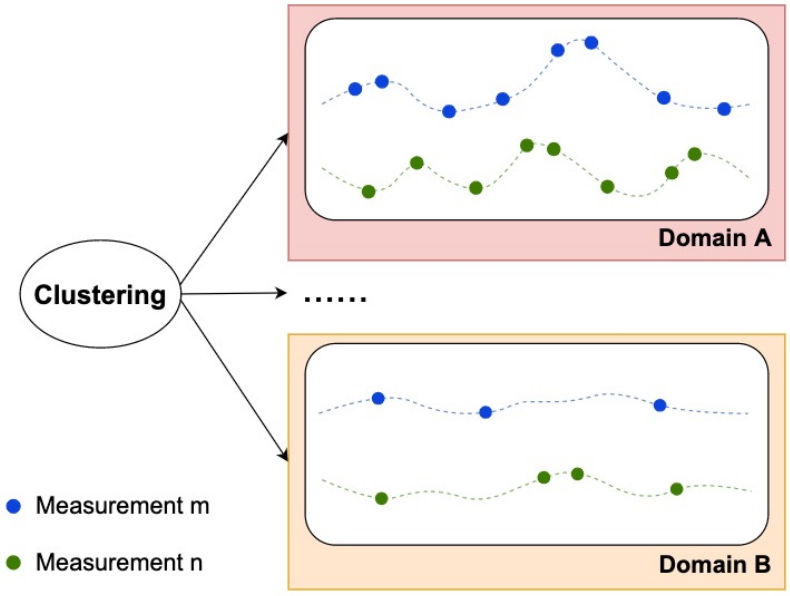


Figure 2.10: The example of clustering for EHRs

2.2.1 Irregular EHRs Clustering

Feature space is an important aspect of data heterogeneity, stemming from the potential diagnoses and clinical observations, for example, patients with cardiovascular diseases have more frequent monitoring of blood pressure, and oxygen saturation is more important to anemia or pulmonary patients. Therefore, the distribution of features, including the presence and frequency of condition-specific features, is valuable. However, the challenge of EHR heterogeneity analysis is that EHRs also vary on the temporal dimension: patients' health conditions are changing over time, and the features are potentially based on varying health statuses. Evaluating the feature distribution of an entire visit is unreasonable because it ignores change of health conditions, and the different timing, often leading to homogenized feature sets.

In order to analyze EHR feature space with the potential influence from other data heterogeneity, we first fix the variety at the temporal dimension. Considering periodicity in hospital visits,

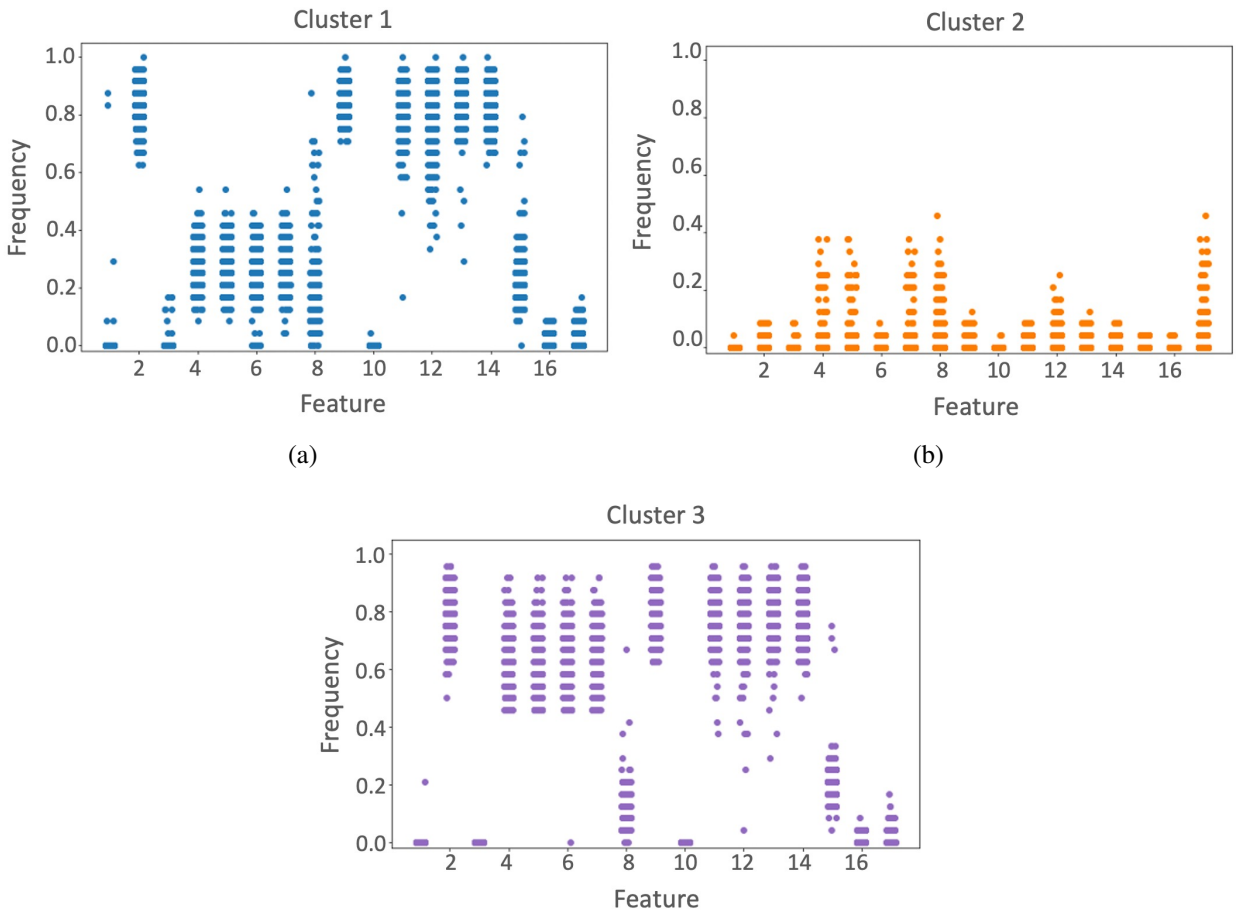


Figure 2.11: Three clusters selected from K-means clustering. The x-axis represents the 17 features being used in clustering, and y-asix is their corresponding frequency from within each cluster. The 17 features are shown in Table 2.8.

e.g., clinical rounds every morning, we evaluate the feature distribution in a 24-hour window. With such a fixed-length time window, every long sequence is split into a few fixed-length sequences and each sequence is treated as an individual sample. This addresses the challenge of patients' health condition changing over time, and the different hospital admission time does not affect the frequency calculation anymore. We calculate the frequency of each feature within every time window, and use K-means to cluster the sequences based on the combination of frequencies of all features, as Figure 2.10 shows. Each cluster then includes the fixed-length sequences with similar feature space distribution, which indicates the potential similar health conditions.

Table 2.8: Features Being Used for Clustering

Index	Feature Name	Index	Feature Name
1	Capillary refill rate	10	Height
2	Diastolic blood pressure	11	Mean blood pressure
3	Fraction inspired oxygen	12	Oxygen saturation
4	Glascow coma scale eye opening	13	Respiratory rate
5	Glascow coma scale motor response	14	Systolic blood pressure
6	Glascow coma scale total	15	Temperature
7	Glascow coma scale verbal response	16	Weight
8	Glucose	17	pH
9	Heart rate		

2.2.2 Clustering Results Analysis

The results of our feature heterogeneity clustering are displayed in Figure 2.11. We computed the frequencies of 17 selected features listed in Table 2.8 and applied K-means with 10 clusters on these frequencies. Three clusters from the clustering result are illustrated in Figures 2.11(a), 2.11(b), and 2.11(c). By comparing Cluster 1 and 2 from Figures 2.11(a) and 2.11(b), we observed that they had very different frequency ranges for diastolic blood pressure (feature #2). Cluster 1 had an average frequency of around 0.8, while Cluster 2 had an average frequency of below 0.1, indicating that patients in Cluster 1 were likely at risk of or diagnosed with cardiovascular diseases, while patients in Cluster 2 did not have such risks. Focusing on Cluster 1 and 3 from Figures 2.11(a) and 2.11(c), we noticed that they had a similar frequency of diastolic blood pressure, implying that they both had risks of cardiovascular diseases. However, cluster 3 had very high frequencies of features #4 to #7, which are related to coma. Therefore, we can infer that patients in cluster 3 were in coma or had a high risk of coma. Based on these two examples, we conclude that our clustering approach can effectively distinguish patients into groups with potentially similar

health conditions.

2.2.3 Clustering Methods Comparison

In the study, we use clustering to address the heterogeneous feature space. Instead of using the actual feature values, we apply the clustering on the feature frequency, so that the samples in each cluster have similar feature occurrences. In medicine, hierarchical clustering is widely applied. Here we compare the statistical analysis of the two different clustering methods - K-means and hierarchical clustering. We use 18 clusters for both methods, the optimal setting obtained from the prediction tasks. For each cluster, we calculate the percentage of positive samples for the two binary classification tasks mortality and decompensation, and the average length of hospital stay for length-of-stay. Then, the results from all the clusters are used to obtain the mean and standard deviation, maximum, and minimum values. Through these statistical data, we can learn if the two clustering methods have a significant difference, and also if any of the clustering methods have a serious bias, for example, separating the very sick patients from others.

Table 2.9 shows our analyzing results. When comparing the two clustering methods K-means and hierarchical clustering, we learn that the two methods do not have a significant difference. They have very similar average, standard deviation, maximum, and minimum values for all three tasks. To lighten the data preprocessing and focus on addressing the multi-source heterogeneity problem, we use the simpler method K-means in the paper to obtain the learning domains for the heterogeneous feature space problem. On the other hand, both clustering methods have low standard deviation values for all three tasks, indicating that both methods do not cause serious bias in the clustering results.

2.2.4 Adaptive Models for Clustered Irregular EHRs

Often, heterogeneous EHR data is handled by training individual models for each subset of data. However, this requires onerous training of multiple models and may result in poorly performing models if the samples have very limited data. Transfer learning is an approach used to aid this limitation across models [43, 44], which can significantly reduce the training time while

Table 2.9: The label distributions with K-means and hierarchical clustering methods.

	K-means			Hierarchical clustering		
	Avg (stdev)	Max	Min	Avg (stdev)	Max	Min
In-hospital mortality	0.146 (0.039)	0.222	0.082	0.144 (0.033)	0.207	0.073
Decompensation	0.027 (0.010)	0.052	0.014	0.026 (0.011)	0.049	0.011
Length-of-stay	153.6(35.9)	225.2	106.9	159.3 (38.8)	230.7	107.4

maintaining performance. Transfer learning methods still suffer from the multiple models it must handle, both in adapting to cases with very limited training data, as well as providing for an obvious selection of models to use in testing for individuals that may be well-suited to more than one model choice. Meta-learning provides a strategy across domains so that models are easily adapted to any unseen or existing tasks and can be used to build adaptive models. In addition, the few-shot-based meta-learning methods only use very few samples in each learning task, which solves the potential problem of limited training data. [40] propose MAML which is widely used in medical applications [91, 92, 93], and [41] applied MAML to in EHRs for rish predictions. Motivated by these works, we build adaptive models by applying MAML to the clustered irregular EHRs.

Figure 2.12 is the framework of using meta-learning (MAML [40]) to build adaptive models for various clusters. There are two optimization steps in building the adaptive models. Given C clusters, with homogeneous samples inside of each, c clusters are randomly sampled, with a batch of training data in each. In the inner loop, each sampled cluster initializes a model and uses its training data to train an adapted model with n steps. After all the sampled clusters obtain their adapted models, another batch of data from each cluster is sampled and tested on their corresponding adapted model. The loss from all the sampled clusters is collected to update the meta-learner (outer loop). In the next training episode, another set of clusters is randomly sampled, and the updated meta-learning is used to initialize the models for each sample cluster.

Applying the few-shot-based meta-learning method, the model training for each cluster is not

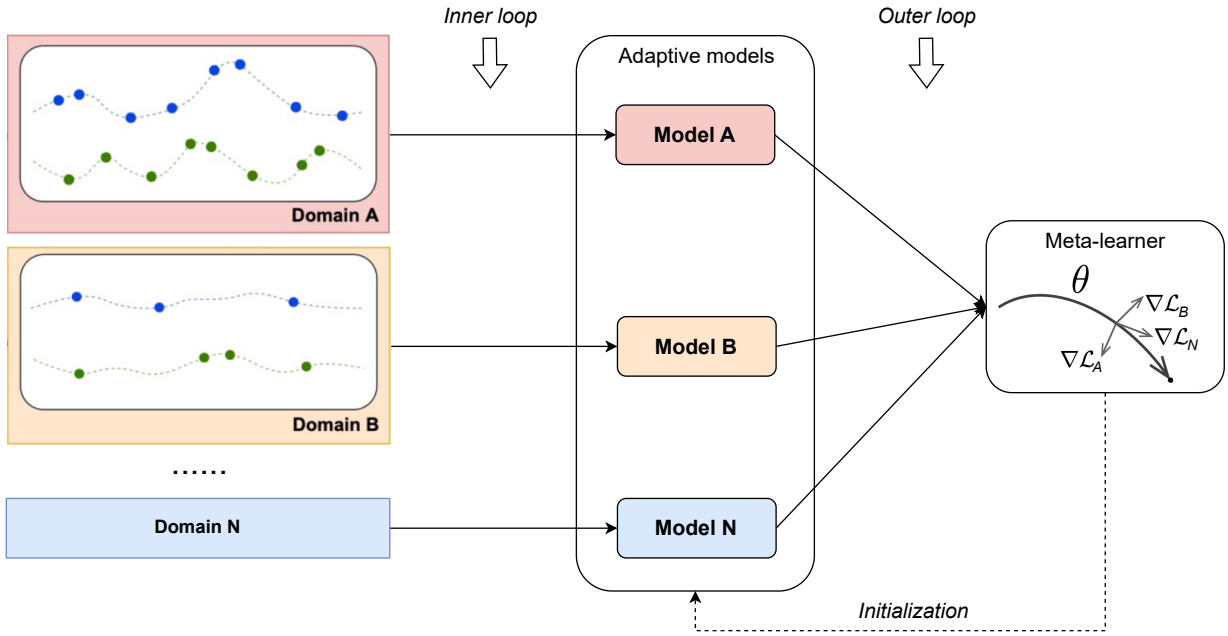


Figure 2.12: Adaptive models on multiple clusters with meta-learning

limited to the potential risk of small data size in certain clusters since there are only two batches of data from each cluster involved in each training episode. Also, the two-stage optimization mechanism helps the meta-learner learn the training path in the very limited n steps from various clusters, and therefore it can be very fast adapted to any cluster with these n steps, and more importantly, the potential new clusters.

2.2.5 Experiments

MIMIC-III (Medical Information Mart for Intensive Care) is a large EHR dataset collected from intensivecare unit (ICU) [4]. MIMIC-III contains the ICU stays of over 38,000 adult patients, which includes a great number of heterogeneous EHR records. We select 17 features and discretize them to be hourly-sampled [13]. The distribution of these selected features of each day is used for clustering and analyzing the data heterogeneity. We test our proposed SSML and the hierarchical structure in two semi-supervised tasks: physiologic decompensation to predict whether a patient's health will rapidly deteriorate in the next 24 hours and length-of-stay to estimate the remaining time until ICU discharge, and additionally test the consistency regularization of SSML on the third

Table 2.10: Average performance (and standard deviations) on MIMIC-III full sequences with time domain variation.

Task	Decompensation		Length-of-stay		In-hospital Mortality	
Evaluation	AUCROC	AUCPRC	Kappa	MAD	AUCROC	AUCPRC
LogisticRegression	0.839	0.246	0.378	161.2	0.825	0.499
	(0.015)	(0.017)	(0.009)	(8.7)	(0.011)	(0.019)
Transformer	0.842	0.260	0.384	147.2	0.836	0.504
	(0.012)	(0.019)	(0.014)	(7.5)	(0.009)	(0.010)
LSTM	0.856	0.313	0.423	152.4	0.847	0.515
	(0.011)	(0.015)	(0.010)	(4.2)	(0.008)	(0.012)
P-LSTM	0.838	0.237	0.426	145.6	0.848	0.505
	(0.009)	(0.013)	(0.012)	(4.9)	(0.006)	(0.008)
MAML-clustering	0.879	0.320	0.428	149.5	0.858	0.540
	(0.008)	(0.011)	(0.011)	(4.7)	(0.009)	(0.014)

task in-hospital mortality predicting the probability of patient mortality in an ICU stay. Decompensation and in-hospital mortality are binary classifications, so we use AUROC and AUPRC as evaluation methods. Length-of-stay is framed as 10 classes/buckets, and Cohen's Kappa coefficient and mean absolute deviation (MAD) are used as the main metrics for this task. We do not deploy Phenotyping because this task lacks variability in MIMIC-III, as phenotypes vary slowly, and are identified by a stable set of sufficient examinations.

2.3 Time Domain Variation

Time domain variation is another important type of heterogeneity in time-series biomedical data. Patients have different lengths of hospital stay, and their health conditions can change over time. Therefore, it is important to build adaptive models for the different variations of data, and understand the variation on the timeline, so that doctors can get suggestions of when should they pay more attention to a patient.

2.3.1 Adaptive Models for Time Domain Variation

Model adaptation techniques, aimed at accounting for different patient conditions, aim to address such variability, but still do not achieve optimal performance. Transfer learning from pre-trained models does not always benefit a new patient, especially when the data amount is small for that patient versus those in the pre-trained model. Different from transfer learning that focuses on optimizing model parameters, meta-learning learns from multiple domains (patients), which provides a potential solution to the challenge of heterogeneity in EHR data resulting from the high variance in ICU stay. Finn et al. [40] proposed Model-Agnostic Meta-Learning (MAML) to optimize model initialization for multiple tasks, enabling rapid adaptation to specific tasks. Meta-learning has been successfully applied in the medical field between different disease onset estimations [41]. However, to the best of our knowledge, meta-learning has not been deployed to develop models that account for heterogeneity in medical datasets as a result of the different available features and timing of that available within the course of hospital admission.

In order to address the variation on the time domain, we propose DynEHR, a dynamic model adaptation method for addressing EHR data heterogeneity with machine learning models for the various lengths of EHR data in MIMIC-III through the duration of admission. The ICU stays are sampled into several domains according to the data lengths by the hour. Inspired by Finn et al. [40], we build DynEHR by applying MAML on an LSTM model, for an optimized initialization that is dynamically adapted to any length of data. In the experiments, DynEHR is compared with four baseline models, an LSTM model trained on all data and tested on the different domains, and

a fine-tuned model on each domain from a pre-trained LSTM model. DynEHR shows the benefits of adapting models to different duration of ICU stay on multiple tasks, including Length of Stay, Phenotyping, Decompensation, and In-hospital Mortality. DynEHR overcomes the challenge of fine-tuning a possible decrease over LSTM, and is able to adapt a model to lengths of EHR data.

2.3.2 DynEHR [3]

2.3.2.1 DynEHR Domains

The heterogeneous nature of EHR data is a challenge in machine learning modeling. Patients can have different characteristics at different durations of ICU stays. For example, during the early stay in ICU, patients' health conditions may change rapidly, which requires a more sensitive model. Therefore, we target dynamically adapting a model to any given length of ICU data that represents a particular duration of ICU stay, as a prototype for heterogeneity adaptation. In order to train our model, we sample N different sequence lengths and generate the EHR data from the training set with the sampled sequence lengths as the training domains (in experiment N is set to be 15). For evaluation, we sample other 18 sequence lengths and generate data from the test set as the test domains. For each test domain, the support set for adaptation is generated from the training set with the corresponding sequence length.

2.3.2.2 DynEHR Modeling

We propose our DynEHR for model adaptation to the various lengths of EHR data. To address the heterogeneity on the temporal dimension, we build an LSTM model on the time-series EHR data as a base model. The adaptation skill is obtained by applying MAML on top of the LSTM model, to understand the characteristic of recurrent computation on an LSTM cell for the various lengths of data.

The domains $D = \{d_1, d_2, ..., d_{15}\}$ are previous defined. Let f donate the LSTM model, and θ indicates the initialization of all the trainable parameters in the LSTM model f. In each training epoch, a subset $D' \sim D$ of m domains is randomly sampled and used in training the model. For

[3]This section is from "DynEHR: Dynamic adaptation of models with data heterogeneity in electronic health records" by Zhang, Lida, Xiaohan Chen, Tianlong Chen, Zhangyang Wang, and Bobak J. Mortazavi.

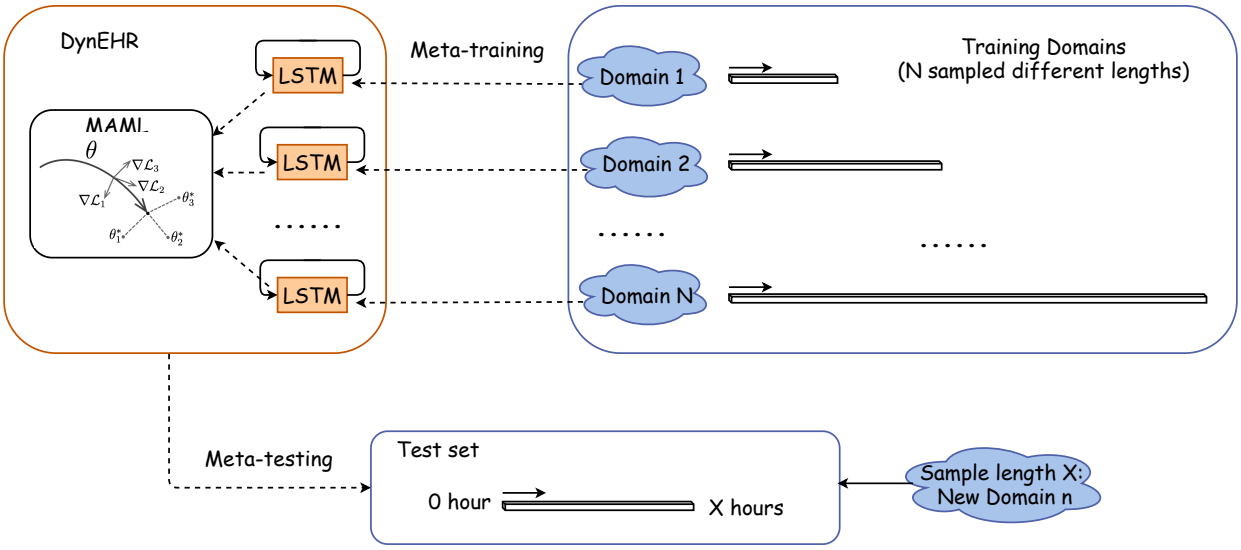


Figure 2.13: Dynamic EHR: DynEHR structure (top left) and meta-training with N training domains (top right). Random length of EHR data is sampled for meta-testing to simulate any duration of ICU stay (bottom).

each domain $d_i \in D'$, we have a corresponding subset of EHR data X_i representing the data with a certain range of length from domain d_i. A support set $X_i^s \subseteq X_i$ and a query set $X_i^q \subseteq X_i$ are sampled from domain d_i, as well as their labels y^s and y^q. A network f_{θ_i} for this length of EHR data in domain d_i is first initialized by $\theta_i = \theta$. The support set X_i^s we have sampled is used for training f_{θ_i} in order to adapt the model to the given length of EHR data in domain d_i:

$$\theta_i \leftarrow \theta_i - \alpha \nabla_{\theta_i} \mathcal{L}_i, \tag{2.6}$$

where α is the learning rate, and $\mathcal{L}_i$ is computed by $\mathcal{L}_i = \mathcal{L}\{f_{\theta_i}(X_i^s), y_i^s\}$ where $\mathcal{L}$ is a loss function for a specific predictive task. For Phenotyping, the loss function is:

$$\mathcal{L} = -\frac{1}{C} \sum y \log \frac{1}{1 + exp(-f_\theta(X))} + (1 - y) \log \frac{exp(-f_\theta(X))}{1 + exp(-f_\theta(X))},$$

where C is the number of classes which equals 25 for Phenotyping. For the other three tasks, $\mathcal{L}$ is

defined as:

$$\mathcal{L} = -\sum p(f_\theta(X)) \log p(y),$$

where p is the distribution of ground truth or prediction. After M steps of training with Eq. 2.6, we can get a network $f_{\theta_i'}$ which has been adapted for domain d_i. Since X_i^s and X_i^q are sampled from the same domain and have a similar length of data and features, $f_{\theta_i'}$ can be applied on X_i^q to test the performance of model adaptation, then we can get the loss from the query set X_i^q:

$$\bar{\mathcal{L}}_i = \mathcal{L}\{f_{\theta_i'}(X_i^q), y_i^q\}.$$

After the various lengths of EHR data from D' has been tested on the adapted model for each domain, the model initialization θ can be updated by loss from all the query sets:

$$\theta \leftarrow \theta - \beta \sum_i^m \nabla_\theta \bar{\mathcal{L}}_i, \tag{2.7}$$

where β is another learning rate.

The training process on the support set X_i^s in Eq.2.6 simulates the adaptation process for each range of data length, and meta-training process in Eq. 2.7 updates the initialization from the adaptation experience in various lengths of EHR data. By learning from the multiple adaptation processes, the meta-learner can obtain the learning skills which allow the model to be adapted to any lengths of EHR data within M steps.

2.3.2.3 Model Testing

As we introduced in section 3.2, 18 unseen domains are sampled to evaluate the model including short, medium, and long ICU stays. There still is a support set sampled from the training set for model adaptation for each unseen domain, and then the adapted model on the different lengths of data is evaluated on the test set of the same domain. By using meta-learning defining domains appropriately, our DynEHR is able to adapt to the various lengths of data. The meta-training pro-

cess allows all the domains to be trained jointly, overcoming the challenge from separate training by domain or fine-tune from a pre-trained model. We will present the experimental details and results in the next section.

2.3.2.4 Experiments

Data Preprocessing

Following the MIMIC benchmark [33], we discretize the irregular EHR data from MIMIC III to be regularly spaced with one hour intervals (and zero pad the beginning of sequences to set a standard length). If there exists more than one data point in an interval, the last value is used, and the most recent value from the previous interval is imputed for the intervals of missing data. The training, validation, and test set are randomly sampled with the size of 32,000, 16,000, and 16,000 ICU admissions from each set, similarly to the MIMIC benchmark, and repeat this process 10 times for model robustness. We re-implemented the standard LSTM benchmark models and find similarly reported values[4], with the exception of the in-hospital mortality task that we have modified.

Task Evaluation

Phenotyping (Phe) As a multi-label classification problem for 25 classes, the phenotyping task is evaluated by the micro-averaged and macro-averaged Area Under the ROC Curve (AUROC) of each predicting class. We train the models with batch size 8 and a learning rate of 0.001. The LSTM model has two layers with unit 128, and the dropout rate is set to 0.2 to the output of LSTM layers. We use the same batch size and learning rate as Length of Stay, as well as the LSTM model. Our baseline LSTM model obtains 0.805 for micro-averaged AUROC and 0.749 for macro-averaged AUROC with all domains.

Decompensation (Dec) The binary classification task Decompensation is evaluated by AUROC as well as the Area Under the Precision-Recall Curves (AUPRC). The model is trained with batch size 8 and a learning rate of 0.001. We choose a single-layer LSTM with a dropout rate of 0.3

[4]We perform slightly better for Length of Stay (Kappa 0.01), slightly worse for Phenotyping (Macro AUC by 0.03). For Decomposition, we do slightly worse for AUPRC (0.02) but better for AUROC (0.01).

of the dropout layer after LSTM. Our baseline achieves 0.897 for AUROC and 0.304 for AUPRC for all domains.

In-hospital Mortality (Mor) Similar to Decompensation, In-hospital Mortality, as a binary classification problem, is also evaluated by AUROC and AUPRC. We use the same batch size, learning rate, and model hyperparameters as Decompensation in this task. When training and testing the model regardless of the data length followed by benchmark, we get AUROC 0.853 and AUPRC 0.541.

Length of Stay (LoS) The models on this task are evaluated by Cohen's kappa coefficient for the inter-annotator agreement, and the mean absolute deviation (MAD) as well as the predicted length of stay and their reference. We use the same hyperparameter setting and batch size as Phenotyping to train the models for Length of Stay. When training and testing the LSTM model with all domains, we obtain 0.443 for Cohen's Kappa score and 130.4 for MAD. We sample our training and test sets without repeating any patient to avoid the potential problem of information leaking.

Baseline Models Our DynEHR is built based on an LSTM model. We compare DynEHR with four baseline models: a Logistic Regression, a pre-trained LSTM, fine-tune in each domain from the pre-trained LSTM model, and a transformer [36]. For the Logistic Regression model, we have a grid-search among penalty and the regularization strength. All LSTM models use unit size 128, and we add a dropout layer with a zeroed probability of 0.2. The Fine-tune method uses the training set from each domain to retrain the pre-trained LSTM model, to adapt the model for the data heterogeneity. The transformer model is also chosen from grid-search, and the optimal setting is query size 8, value size 8, 4 heads, 4 stacks, and the attention window size of 12. We have provided the results of each predictive task by testing without recognizing domains in the previous Section, showing similar results as the MIMIC benchmark. In the following analysis, we will only focus on the model performance on the various domains respecting the model adaptation ability toward the different lengths of EHR data.

Experiments on DynEHR

The base model LSTM in DynEHR is set to be the same structure with the same hyperparameters as the baseline models in section 4.3. To train our DynEHR model, we have a support set and a query set for each domain. During training, the support sets are used to adapt the model to each domain with a given number of training steps, and then query sets are applied to the adapted model and used to calculate the loss to update the model initialization. In our experiments, the support set and query set both have eight samples for each domain, and the optimal number of adaptation steps is set to be 10. The support sets and query sets are randomly sampled from the training set. We initialize the DynEHR with the pre-trained model from LSTM. When testing the performance of DynEHR, we also randomly sample data lengths as a new domain, and sample support set for each domain from the training set for model adaptation. The adapted model is tested on the corresponding domain of the whole test set. Similar to the training process, the support set during testing also has eight samples for each domain, and the number of adaptation steps is 10 as well. The average result of our DynEHR is in Table 2.11.

Results and Discussion

To have a direct comparison of our proposed DynEHR and the baseline models, we provide the average performance of all the domains in Table 2.11. The raw result of each sampled testing domain is presented in the Appendix. Adapting a model to a certain type of EHR data can be challenging. Compared to an LSTM model, fine-tuning the pre-trained LSTM model to each domain only brings very limited improvement on Phenotyping and Decompensation, but can cause a decrease in Length of Stay and In-hospital Mortality. Facing this challenge, DynEHR can successfully achieve the goal model adaptation to the heterogeneity in EHR data. DynEHR is able to improve the average performance on all four tasks. It is interesting that the three LSTM-based models (LSTM, fine-tuning, and DynEHR) perform significantly better than Logistic Regression and Transformer. We believe that LSTM is a better setting for this time-series EHR data. We also observed that for task Length of Stay, the performance on Cohen's Kappa does not always match MAD. The experiments show the dynamic adaptation ability for DynEHR on unseen EHR data heterogeneity.

Table 2.11: Comparison of the average performance over all test domains.

Task	Evaluation	Logistic Regr	LSTM	Fine-tune	Transformer	DynEHR
Phe	Micro AUC	0.786	0.796	0.797	0.778	**0.806**
		(0.018)	(0.023)	(0.023)	(0.027)	**(0.022)**
	Macro AUC	0.710	0.728	0.727	0.702	**0.739**
		(0.021)	(0.025)	(0.031)	(0.030)	**(0.025)**
Dec	AUROC	0.783	0.824	0.824	0.808	**0.836**
		(0.090)	(0.060)	(0.066)	(0.060)	**(0.055)**
	AUPRC	0.193	0.258	0.261	0.176	**0.287**
		(0.146)	(0.102)	(0.103)	(0.102)	**(0.106)**
Mor	AUROC	0.741	0.826	0.822	0.795	**0.839**
		(0.122)	(0.049)	(0.046)	(0.051)	**(0.047)**
	AUPRC	0.398	0.536	0.518	0.434	**0.551**
		(0.206)	(0.097)	(0.082)	(0.101)	**(0.099)**
LoS	Cohen's Kappa	0.230	0.353	0.320	0.325	**0.365**
		(0.140)	(0.080)	(0.093)	(0.087)	**(0.073)**
	MAD	191.4	**132.2**	148.1	155.6	135.0
		(106.1)	**(38.3)**	(56.4)	(64.4)	(44.9)

In Table 2.12, we present some randomly picked domains with short, middle, and long sequence lengths for Phenotyping. From the comparison in this table, DynEHR has constant benefits on all the test domains. The improvement of DynEHR over both LSTM and fine-tuning indicates its ability to adapt a model to any new types of EHR data. We notice that Logistic Regression has slightly higher macro-averaged AUROC than DynEHR in the first domain which includes the shortest sequences, but it performs worse than LSTM, fine-tuning, and DynEHR on all other domains with longer sequence length. This result implies that the Logistic Regression may not be a good choice for long sequential data. When comparing LSTM and fine-tuning, we observed that fine-tuning can bring slight improvement to LSTM on the short sequence domains, but it causes decreases in the long sequence domains. The challenge of fine-tuning on short and long EHR data also commonly happens on the other three tasks. The problem of long EHR data for fine-tuning

Table 2.12: Phenotyping results in short, middle, and long sequences.

	Logistic Reg		LSTM		Fine-tune		Transformer		DynEHR	
SeqLength	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
short	0.761	**0.685**	0.749	0.659	0.760	0.682	0.750	0.677	**0.762**	0.684
	0.779	0.699	0.781	0.727	0.780	0.737	0.778	0.709	**0.794**	**0.749**
	0.815	0.727	0.834	0.760	0.834	0.767	0.822	0.747	**0.847**	**0.777**
middle	0.824	0.746	0.825	0.754	0.824	0.759	0.816	0.738	**0.831**	**0.773**
	0.805	0.738	0.816	0.743	0.815	0.744	0.803	0.729	**0.822**	**0.752**
	0.793	0.729	0.821	0.751	0.825	0.759	0.784	0.689	**0.833**	**0.758**
long	0.783	0.720	0.781	0.718	0.775	0.696	0.770	0.705	**0.792**	**0.720**
	0.762	0.698	0.769	0.685	0.752	0.641	0.716	0.625	**0.775**	**0.686**
	0.763	0.695	0.762	0.702	0.768	0.696	0.736	0.665	**0.780**	**0.724**

method comes from the memory constraints of recurrent networks [36], and the small amount of long EHR data can easily cause overfitting problems in fine-tuning.

Table 2.13: Decompensation results in short, middle, and long sequences.

	Logistic Reg		LSTM		Fine-tune		Transformer		DynEHR	
SeqLength	auroc	auprc	auroc	auprc	auroc	auprc	auroc	auprc	auroc	auprc
short	0.811	0.186	0.817	0.251	0.828	0.243	0.812	0.140	**0.835**	**0.272**
	0.861	0.075	0.863	0.180	0.867	0.138	0.834	0.110	**0.870**	**0.244**
	0.889	0.101	0.907	0.341	0.903	0.348	0.876	0.328	**0.908**	**0.423**
middle	0.800	0.400	0.907	0.319	0.905	0.404	0.886	0.257	**0.915**	**0.410**
	0.822	0.211	0.840	0.192	0.846	0.195	0.848	0.187	**0.848**	**0.212**
	0.802	**0.250**	0.803	0.137	0.806	0.187	0.786	0.071	**0.810**	0.180
long	0.822	0.350	0.823	0.381	0.795	0.404	0.830	0.266	**0.832**	**0.421**
	0.687	0.087	**0.807**	**0.197**	0.804	0.170	0.734	0.052	0.793	0.153
	0.680	0.085	0.723	0.378	0.802	0.393	0.691	0.043	**0.806**	**0.460**

Table 2.13 is the results of selected domains on Decompensation. The benefit of DynEHR is shown on AUROC by improving 0.012 over LSTM and fine-tuning on average, and on AUPRC by improving 0.029 over LSTM and 0.026 over fine-tuning. From Table 2.13, DynEHR still has significant benefits in different lengths of domains, especially in short sequence domains. There is one domain from middle sequences that Logistic Regression has better AUPRC, and one domain from long sequences that LSTM performs better. There is no other model that can have a close or better average performance than DynEHR. On this task, fine-tuning has a slight increase compared to LSTM.

Table 2.14: In-hospital Mortality results in short, middle, and long sequences.

	Logistic Reg		LSTM		Fine-tune		Transformer		DynEHR	
SeqLength	auroc	auprc	auroc	auprc	auroc	auprc	auroc	auprc	auroc	auprc
short	0.753	0.117	0.775	0.390	0.768	0.367	0.740	0.283	**0.785**	**0.390**
	0.717	0.052	0.808	0.428	0.812	**0.435**	0.782	0.309	**0.821**	0.426
	0.833	0.315	0.833	0.519	0.850	0.544	0.813	0.434	**0.854**	**0.530**
middle	0.873	0.757	0.850	0.564	0.846	0.565	0.821	0.457	**0.860**	**0.578**
	0.878	0.596	0.878	0.571	0.838	0.506	0.855	0.468	**0.898**	**0.599**
	0.773	0.488	0.878	0.637	0.875	0.506	0.859	0.558	**0.889**	**0.662**
long	0.705	0.419	0.812	0.564	0.813	0.557	0.808	0.537	**0.823**	**0.590**
	0.761	0.587	0.799	0.595	0.783	0.583	0.779	0.503	**0.821**	**0.613**
	0.400	0.238	0.772	0.598	0.761	0.526	0.708	0.417	**0.792**	**0.614**

The results of In-hospital Mortality in domains with different sequence lengths are presented in Table 2.14. This task is modified to extend 48-hour EHR data to various lengths of EHR data to predict patient mortality in hospitals. DynEHR again shows its significant benefits of dynamically adapting to various types of EHR data. The average performance of DynEHR has an improvement of AUROC 0.013 and AUPRC 0.015 over LSTM, and AUROC 0.017 and AUPRC 0.033 over fine-

tuning. From Table 2.14, fine-tuning has better AUPRC values in short-sequence domains than LSTM, but brings decreases on both AUROC and AUPRC in all the middle- and long-sequence domains compared to LSTM. DynEHR can successfully address the limitation of fine-tuning by increasing the AUROC and AUPRC in long sequence domains compared to LSTM, which shows its ability to dynamically adapt to various lengths of EHR data. Logistic Regression and Transformer perform both worse than the other two baseline models LSTM and fine-tuning.

Table 2.15: Length of stay results in short, middle, and long sequences.

	Logistic Reg		LSTM		Fine-tune		Transformer		DynEHR	
SeqLength	Kappa	MAD	Kappa	MAD	Kappa	MAD	Kappa	MAD	Kappa	MAD
short	0.108	108.8	0.163	109.5	0.173	107.6	0.101	110.4	**0.177**	**103.4**
	0.258	85.3	0.322	**84.1**	0.205	91.8	0.252	89.5	**0.329**	87.9
	0.297	**92.0**	0.382	97.2	0.319	96.0	0.338	92.3	**0.386**	96.7
middle	0.284	108.2	0.459	101.6	0.448	104.1	0.427	111.9	**0.460**	**101.3**
	0.401	194.5	0.409	133.2	0.397	158.7	0.388	162.0	**0.411**	**132.1**
	0.243	233.6	0.389	156.4	0.395	173.9	0.374	194.8	**0.399**	**153.2**
long	0.207	260.6	0.327	175.3	0.335	217.9	0.329	199.2	**0.342**	**172.3**
	0.022	367.7	0.338	**179.1**	0.302	217.1	0.251	257.2	**0.346**	207.8
	0.002	350.5	0.202	**195.3**	0.180	252.8	0.225	265.4	**0.230**	212.5

Table 2.15 shows the performance of the five models in the selected domains on the task of Length of Stay. DynEHR always has the best Cohen's Kappa score in all the sequence length ranges and the best MAD in most domains, but the other models may have better MAD. From the table, DynEHR has the best Cohen's Kappa score in all lengths of sequences, and it is interesting that in one short sequence domain, Logistic Regression has a higher Cohen's Kappa score than DynEHR and LSTM. However, in long sequence domains, Logistic Regression performs very poorly with Cohen's Kappa score lower than 0.1 and MAD higher than 200, which matches the

observation from the other three tasks that Logistic Regression does not perform well on long sequences. As we mentioned before, the changes of MAD do not always stay consistent with Cohen's score. In some long sequence domains, LSTM has a worse Cohen's Kappa score but lower MAD than DynEHR. The shortcoming of fine-tuning is very obvious in the long-sequence domains by decreasing Cohen's Kappa score and increasing the MAD value. Even though DynEHR has higher MAD values than LSTM in some domains, this change is much less than fine-tuning brings and it still significantly improves Cohen's Kappa score. Here we pay more attention to Cohen's Kappa score because it shows the coefficient between the predictions and targets, and a low MAD value may come from constant prediction values without any variance from a poorly performed model.

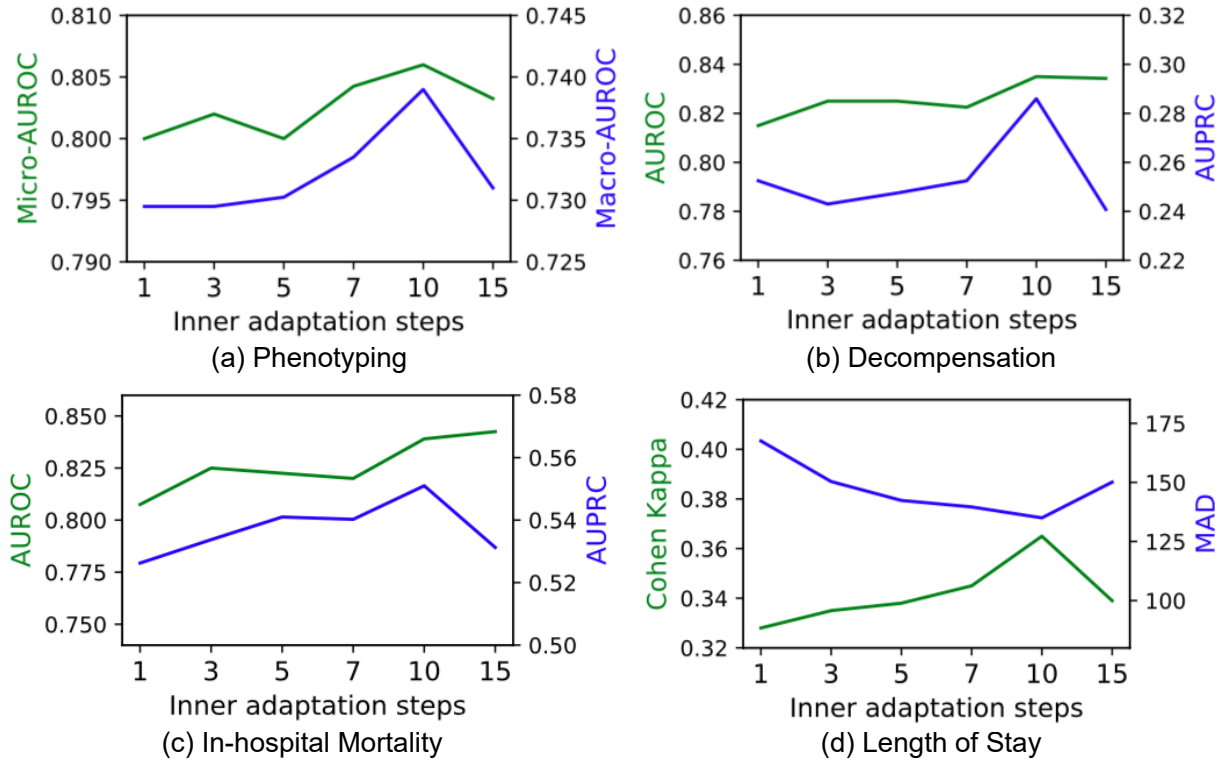


Figure 2.14: Average Meta-EHR performance with different numbers of inner adaptation steps (X-axis) and evaluation measurements (y-axis). The green line is the measurement on the left axis and the blue the right. Ten steps of adaptation is the optimal for all four tasks

Figure 2.14 shows the average DynEHR performance on all domains trained by different numbers of inner adaptation steps. When there is only one adaptation step, DynEHR does not perform well on all the tasks. For example, in Figure 2.14(d), DynEHR on Length of Stay trained by one adaptation step has the lowest Cohen's Kappa score and the highest MAD. As the number of adaptation steps gets increases, the average performance goes up until the maximum of 10 steps of adaptation. After 10 steps of adaptation, the model performance drops down again when trained with 15 steps. In Figure 2.14(c), In-hospital Mortality, even though the AUROC score is the highest when training with 15 adaptation steps, the AUPRC score has a decrease compared to 10 adaptation steps. The low performance with fewer than 10 steps of adaptation indicates the great differences of the domains. With the optimized initialization, DynEHR still requires a decent number of adaptation steps to each domain. The decrease in 15 adaptation steps is caused by overfitting.

2.4 Conclusion

In this chapter, we present our solutions for three different forms of individual data heterogeneity problems: heterogeneous data distribution, irregularly sampled time-series data, and time domain variation. We propose a DANN-based MTL model to estimate beat-to-beat blood pressure from cuffless bioimpedance signals for new subjects with reduced training data. When reducing the training data to three, four, and five minutes, the base MTL model cannot directly be trained successfully to be within ISO standards. Therefore, in order to transfer knowledge from other subjects efficiently, we modify the DANN training approach to train the feature extractor for subject-invariant features. With DANN, the model obtains average RMSE 4.80 ± 0.74 mmHg for diastolic blood pressure and 7.34 ± 1.88 mmHg for systolic blood pressure when using three minutes training data, 4.64 ± 0.60 mmHg and 7.19 ± 1.79 for diastolic and systolic blood pressure from four minutes training data, and 4.48 ± 0.57 mmHg and 6.79±1.70 for diastolic and systolic blood pressure, respectively, when applying five minutes training data. DANN improves the knowledge transfer ability for three, four, and five minutes of training data in comparison to directly training or training with a pretrained model from another subject, decreasing RMSE by 0.19 to 0.26 mmHg for

diastolic blood pressure and by 0.46 to 0.67 mmHg for systolic blood pressure in comparison to the best baseline model of utilizing a pretrained model from another subject. The model performance increases with additional data, and we conclude that four minutes is the minimum requirement to achieve the ISO standard with our proposed model and participant cohort. For the irregularly sampled time-series data, we first apply clustering to analyze the irregularity, attempting to group patients with similar health conditions, and then apply meta-learning to build adaptive models for each group of patients. Our adaptive model outperforms multiple baseline models on different prediction tasks, which shows the ability of our model of addressing irregular time-series data. For the time domain variation, we propose DynEHR to address the various lengths of EHR data as a protocol for dynamic model adaptation. DynEHR uses meta-learning to train an optimized initialization and learning the optimization process, so that it can be easily adapted and applied to any duration of an ICU admission. By testing on the four MIMIC benchmark tasks, Length of Stay, Phenotyping, Decompensation, and In-hospital Mortality, DynEHR can successfully adapt the model to the various lengths of data, and has significant benefits over the possible performance decrease caused by fine-tuning. To address limitations in our findings, future work will test our DynEHR on other data heterogeneity in EHRs, such as the sampling frequency of vitals.

3. MULTI-SOURCE HETEROGENEITY

Data heterogeneity is a natural attribute of many real-world applications and datasets in the time-series domain. Heterogeneity occurs frequently and can be complex across several dimensions: features, labels, and the time-varying nature of data. On the feature dimension, heterogeneity may come from the development of new sensors [94, 95], missing data [90], data noise [14, 96, 97] or different setups for data collection [98]. The difficulty in observing ground truth [53, 54] and obtaining inconsistent user feedback [99] may result in label uncertainty. In the time domain, the variation present in changing health conditions of patients [100], changes in seasons [101], cycles in the economy [102], or even the spreading of disease in a pandemic can all lead to vastly different data representations and ranges. The different types of heterogeneity can occur not only individually but also simultaneously, and thus result in a problem of multi-source heterogeneity in time-series modeling and applications.

Often, heterogeneous EHR data is handled by training individual models for each subset of data. However, this requires onerous training of multiple models and may result in poorly performing models if the sames have very limited data. Transfer learning is an approach used to aid this limitation across models [43, 44], which can significantly reduce the training time while maintaining performance. Transfer learning methods still suffer from the multiple models it must handle, both in adapting to cases with very limited training data, as well as providing for an obvious selection of models to use in testing for individuals that may be well-suited to more than one model choice. Meta-learning provides a strategy across domains, so that models are easily adapted to any unseen or existing tasks and can be used to build adaptive models. In addition, the few-shot-based meta-learning methods only use very few samples in each learning task, which solves the potential problem of limited training data. [40] propose MAML which is widely used in medical applications [91, 92, 93], and [100] provide an example of applying MAML to address the EHR heterogeneity across the time domain but not across the feature space. Zhang et al. proposed a layer-flexible RNN model for the variation in each time-series sequence [103], however, is limited

Table 3.1: Overview of the ML techniques addressing various types of time-series heterogeneity

Algorithm	Heterogeneous features	Label uncertainty	Time domain variation
Recurrent network			✓
Transformer			✓
Transfer learning	✓		✓
Meta-learning	✓		
Semi-supervised learning		✓	
SSML (Ours)	✓	✓	
SSML-TDV (Ours)	✓	✓	✓

to the time domain variation.

The feature space in EHRs is, perhaps, a more informative aspect, for meta-learning to apply to, because various feature distributions may arise from diagnoses, examinations, and treatment decisions that stem from varying states and health conditions. We learn the homogeneous sets in the heterogeneous EHRs by fixing the temporal dimension with a time window and clustering the window-length sequences based on their feature distributions. However, only the last window of each sequence has a guaranteed label, and the temporal level of heterogeneity in EHRs results in a labeling problem in the earlier windows. For example, a patient in a stable (non-risky) status at the last window might have experience decomposition in health condition earlier in hospital stay that was properly treated and resolved. This may result in incorrectly classifying patient risk as high or low depending on the window of time perceived. This labeling problem also happens in other fields of applications, such as image [104], wearable sensing [105], and language [106, 107].

Our objective is to address the intricate challenge of multi-source data heterogeneity in medical applications, which are among the most complex types of time-series data, involving all three types of data heterogeneity. Firstly, medical data encompasses numerous observations, such as laboratory tests and medications, from hospitals [4], and the frequency and category of these measurements depend on doctors' assessments, implying the potential health condition. Incorporating

similar frequency patterns in medical data can enhance the model's specificity for certain patient types, leading to better risk prediction tasks and facilitating timely clinical decision-making. Secondly, medical data presents challenges in obtaining labels, as diagnoses from doctors are time-sensitive, and changes in patients' health conditions can affect the labels. Thirdly, the variation in patients' health conditions introduces heterogeneity in the time domain, and this can be further compounded by factors such as hospital treatments [59], hospital transfers [60], and ICU admission and release [61].

Facing these challenges, the goal of this paper is to build adaptive models to address the multi-source heterogeneity that can occur simultaneously in time-series data. We propose a semi-supervised meta-learning algorithm for the heterogeneous features and uncertainty in labels. Meta-learning, in the manner of few-shot learning, addresses the potential data limitation in certain types of feature space and the demand for fast adaptation in the future. A discriminator is introduced for adversarial training to improve the model generalization. Regarding the variation over time, we propose a time domain variation (TDV) framework applying transfer learning and our SSML. Our approach is a new connection between meta-learning, transfer learning, and semi-supervised learning. We test our approaches on two real-world medical datasets, PhysioNet Challenge 2012 and the MIMIC-III ICU dataset. To the best of our knowledge, we are the first the address this complex real-world simultaneous multi-source heterogeneity of feature space, time domain variation, and label uncertainty on time-series data (Table 3.1). Our proposed model is flexible to address all or part of the heterogeneity problem, and is also adaptive for future model update demands.

3.1 Related Work

Meta-Learning

Meta-learning is designed to extract information about the optimization process on a few samples for various learning tasks [40, 108, 109]. Finn et al. [40] propose MAML, which optimizes the model initialization as the meta-learner, and is widely applied to a large number of healthcare applications [91, 92, 93]. Zhang et al. [41] apply MAML on EHR data to predict clinical risk for patients, and Zhang et al. [100] propose DynEHR based on MAML to model for the various

duration of EHR data. Ren et al. [104] first introduce semi-supervised learning to the few-shot learning algorithm Prototypical Network [68]; however, refining the prototype of each class without differentiating the domains cannot achieve the goal of building adaptive models for various EHR sequences. Our proposed model is also motivated by MAML and we compare SSML with MAML for heterogeneous EHR data modeling.

Semi-supervised Learning

The goal of semi-supervised learning is to make use of unlabeled data. Self-training uses the model prediction of unlabeled data as the produced label and is applied in many applications [45, 46, 47, 48, 110]. Pseudo-labeling further converts the confident prediction to hard labels [49], but this may not be stable [48, 111]. Consistency regularization [50, 51, 52] is then introduced to self-training [47]. Sohn et al. [112] propose FixMatch using augmentation [113] as a consistency regularization into pseudo-labeling. Meta-learning is then applied in FixMatch as a new semi-supervised learning approach [57, 58]. However, meta-learning here is only applied between the labeled and unlabeled data of the same learning task, and these two papers, which are semi-supervised learning algorithms, cannot be used on multiple learning tasks and datasets, nor do they serve as an adaptive model for our data heterogeneity problem. Therefore, we do not directly compare them.

EHR clinical analysis

EHR has been studied in both medicine and machine learning since its wide use in hospitals. Harutyunyan et al. [13] propose an LSTM-based multi-task model for clinical prediction with EHR variables, and Xu et al. [34] introduce waveform data in their model. Transformer [36] is first used in the EHR model as a replacement of LSTM by Song et al. [114]. However, none of these works have considered the heterogeneity in EHR data. Shukla [39] addresses the irregularly-sampled data by mapping it to a regular space, but there is no specified analysis about each homogeneous set in the heterogeneous in EHRs. Zhang et al. [100] propose DynEHR as an adaptive model for EHRs, but the method is not flexible enough to be applied in other types of data heterogeneity other than the temporal source.

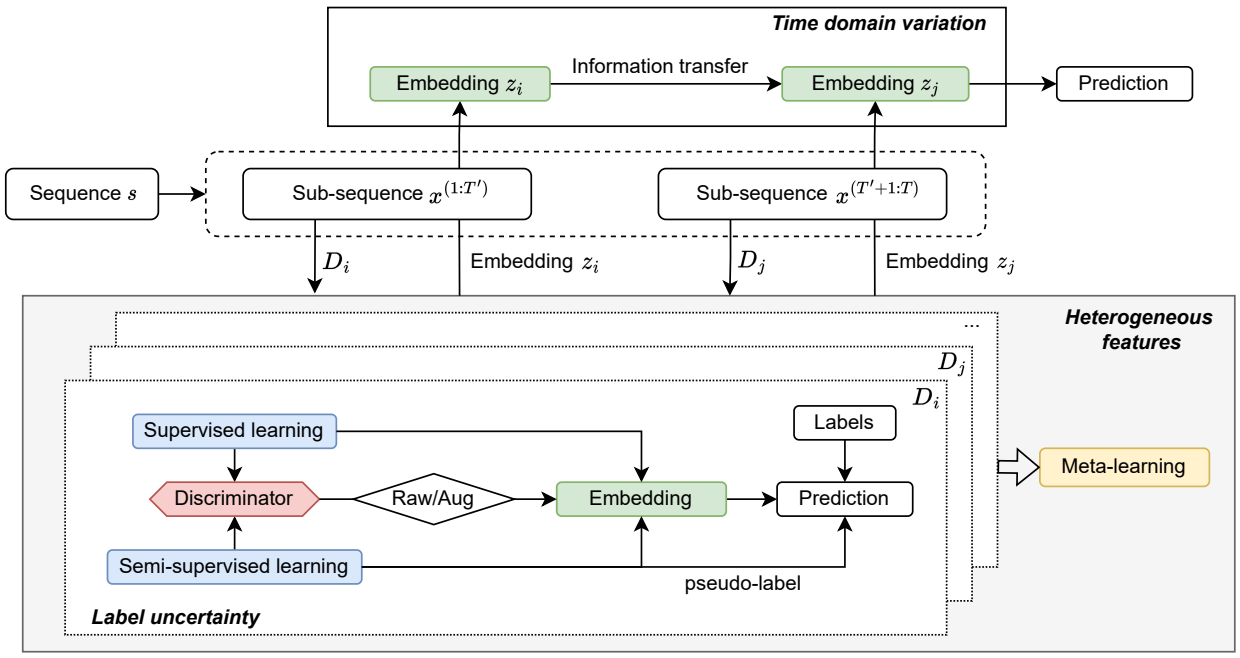


Figure 3.1: The framework of SSML-TDV for multi-source time-series heterogeneity. (***Bottom***) Semi-supervised meta-learning (SSML) with adversarial training for heterogeneous features and label uncertainty. (***Top***) The SSML-based time domain variation framework (SSML-TDV). Each sequence participates in SSML training, and applies the trained SSML with transfer learning for predictions.

3.2 Methodology

In this section, we present our solution for the multi-source heterogeneity in time-series data. We define the heterogeneous features challenge as a multi-domain problem, and each domain includes homogeneous examples. We use the meta-learning framework as a fast adaptive model for each domain, and propose the semi-supervised meta-learning algorithm (SSML) with adversarial training for the label uncertainty in the multi-domain setting, and SSML is then applied with transfer learning in a time domain variation (TDV) framework for the third level of heterogeneity.

3.2.1 Problem Setup

In this study, a set of domains represents the varied, heterogeneous feature space for the learning tasks. Each domain includes sequences with similar feature frequency distribution. Let $\mathcal{D}$ denote all domains, and $D_i \in \mathcal{D}$ represents the i-th domain. Let $\mathcal{X}_i$ and $\mathcal{U}_i$ denote the labeled

and unlabeled data in domain D_i, and $\mathcal{Y}_i$ is the corresponding label of $\mathcal{X}_i$, then domain D_i has $D_i = \{\{\mathcal{X}_i, \mathcal{Y}_i\}, \mathcal{U}_i\}$.

Let $\mathcal{S}$ be a set of time-series data. Given a sequence example $s = x^{(1:T)}$ from $\mathcal{S}$ ($s \in \mathcal{S}$) containing T time stamps, and $x^{(t)}$ represents the feature vector at time point t ($1 \leq t \leq T$). Assume the time domain variation occurs on s (e.g., complication happening to a patient), which splits the sequence into sub-sequences at time point T':

$$s = \underbrace{x^{(1)}, ..., x^{(T')}}_{D_i}, \underbrace{x^{(T'+1)}, ..., x^{(T)}}_{D_j} \tag{3.1}$$

where sub-sequences $s^1 = x^{(1:T')}$ belongs to domain D_i and $s^2 = x^{(T'+1:T)}$ belongs to D_j. The time domain variation on long sequences also causes label uncertainty among the sub-sequences, such that $s^1 \in \mathcal{U}_i$ and $s^2 \in \mathcal{X}_j$ (the uncertain label may also come from unlabeled data). The goal of our work is to build adaptive models under a multi-domain setting respecting the potential shifts among different domains within each sequence s and the uncertain label problem.

3.2.2 Semi-supervised Meta-learning

Supervised meta-learning

An underlying challenge of the heterogeneous feature space is the potential limitation of having sufficient training examples in each domain. In addition, standard supervised learning is also limited to future demands of model adaption in practice, for example, when there is a new disease discovered but very limited patient examples are collected. Therefore, we address the data heterogeneity problem in a meta-learning setting.

Given a model $\mathcal{F}$ consisting a feature extractor $\mathcal{F}_\theta$ and a predictor $\mathcal{F}_\eta$, where θ and η represent their parameters correspondingly. The goal of meta-learning is to learn the optimization process of several domains and optimize the model initialization θ and η in $\mathcal{F}$, so that model $\mathcal{F}_{\theta;\eta}$ can be optimized to be very fast adapted to $\mathcal{F}_{\theta_k;\eta_k}$ for any domain D_k.

For a domain D_i from training domains $\mathcal{D}$, the model $\mathcal{F}_{\theta_i;\eta_i}$ is initialized with θ and η. Given the labeled data $\{x_i, y_i\} \subseteq \{\mathcal{X}_i, \mathcal{Y}_i\}$ in D_i, model $\mathcal{F}_{\theta_i;\eta_i}$ can be trained through supervised learning

with cost

$$\mathcal{L}^{l}_{D_i}(\theta_i, \eta_i) = \mathcal{L}(\mathcal{F}_{\theta_i;\eta_i}(x_i), y_i), \tag{3.2}$$

where $\mathcal{L}$ represents the cost function (mean-squared error for a regression task or cross-entropy for a classification task). After N steps of training with gradient descent, $\mathcal{F}_{\theta_i;\eta_i}$ becomes the adapted model $\mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i}$:

$$\bar{\theta}_i = \theta_i - \alpha \frac{\partial \mathcal{L}^{l}_{D_i}(\theta_i, \eta_i)}{\partial \theta_i}, \quad \bar{\eta}_i = \eta_i - \alpha \frac{\partial \mathcal{L}^{l}_{D_i}(\theta_i, \eta_i)}{\partial \eta_i}, \tag{3.3}$$

where α is the step size.

For the purpose of fast adapting to any domain, model $\mathcal{F}_{\theta;\eta}$ needs to learn from several domains. In each training episode, we randomly generate a set of domains $D \subseteq \mathcal{D}$, and train their adapted model from Equation 3.2 and 3.3. After each domain $D_i \in D$ obtaining its adapted model $\mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i}$, another set of data $\{\bar{x}_i, \bar{y}_i\} \subseteq \{\mathcal{X}_i, \mathcal{Y}_i\}$ (query set) is sampled and tested on the adapted model:

$$\bar{\mathcal{L}}^{l}_{D_i}(\bar{\theta}_i, \bar{\eta}_i) = \mathcal{L}(\mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i}(\bar{x}_i), \bar{y}_i), \tag{3.4}$$

and θ, η is optimized with all domains in D:

$$\theta = \theta - \beta \frac{\partial \sum_{D_i}^{D} \bar{\mathcal{L}}^{l}_{D_i}(\bar{\theta}_i, \bar{\eta}_i)}{\partial \theta},$$

$$\eta = \eta - \beta \frac{\partial \sum_{D_i}^{D} \bar{\mathcal{L}}^{l}_{D_i}(\bar{\theta}_i, \bar{\eta}_i)}{\partial \eta},$$

where β is another step size.

Semi-supervised learning

Facing the challenge of label uncertainty in the multi-domain setting, we extend supervised-based meta-learning to become semi-supervised learning. Inspired by [49], we convert the model prediction of the unlabeled data to be a hard label as their pseudo-label. Similar to the supervised-learning part, we randomly generate the unlabeled data $\{u_i\} \subset \{\mathcal{U}_i\}$ for each domain D_i. The pseudo-label $\hat{y}_i$ of u_i is produced from the outcome of the model. A problem with using the model

outcome as the pseudo-label is that the produced label may include bias from a poorly-trained model in the early training stage. A threshold τ is then introduced to filter the maximum value of unlabeled data prediction, so that only high-confidence outcomes will be converted to hard labels as the produced pseudo-label:

$$\hat{y}_i = \mathbb{1}(\max(\mathcal{F}_{\theta_i;\eta_i}(u_i)) \geq \tau). \tag{3.5}$$

With pseudo-label, the unlabeled data then have goals to compare with. However, if we directly calculate the cost between the model prediction of u_i and its pseudo-label $\hat{y}_i$, the model will only be trained to maximize the maximum value of u_i, because both the prediction $\mathcal{F}_{\theta_i;\eta_i}(u_i)$ and pseudo-label $\hat{y}_i$ are functions of u_i. Therefore, we further introduce the augmentation from consistency regularization [113, 112]. Augmentation adds noise to the unlabeled data, playing a similar role as the activation function to prevent the prediction of the unlabeled data from being a linear function of u_i. More importantly, as a regularization method, augmentation increases the model generalization and stability: the model should predict the same outcome even with some noise. The semi-supervised part for domain D_i can be presented as

$$\mathcal{L}^u_{D_i}(\theta_i, \eta_i) = \mathcal{L}(\mathcal{F}_{\theta_i;\eta_i}(\mathcal{A}(u_i)), \hat{y}_i), \tag{3.6}$$

where $\mathcal{A}(\cdot)$ denotes the augmentation function for unlabeled data, for example, cropping, flipping, and noise injection techniques [115, 116]. In our study, each feature represents a measurement taken in-hospital, and we augment the data with random feature removal, with the assumption that the model should produce similar output even if some measurements are missing. [1]

Adversarial training

By augmenting the unlabeled data for consistency regularization, noise is introduced in training. In order to minimize the side effect of augmentation in the training process, we further modify

[1] Flipping is not an ideal augmentation because the scales of measurements vary, but it could be an option for other time-series data such as ECG. We also tried augmenting the data by adding noise and found that data removal (cropping) is a better solution.

the semi-supervised domain-adapted model training to be adversarial training [26]. We design the adversarial training between the labeled data and the augmentation of unlabeled data by classifying the source of the latent space from $\mathcal{F}_{\theta_i}$. On the one hand, adversarial training can improve from introducing augmentation, and on the other hand, the potential data shift between labeled and unlabeled data can be addressed too. A discriminator $\mathcal{F}_\phi$ is introduced for the data source classification in each domain D_i:

$$\mathcal{L}^d_{D_i}(\theta_i, \phi_i) = \log(\mathcal{F}_{\theta_i;\phi_i}(x_i)) + \log(1 - \mathcal{F}_{\theta_i;\phi_i}(\mathcal{A}(u_i))) \tag{3.7}$$

where ϕ represents the parameters of the discriminator.

During the model adaptation process of each domain, the feature extractor and predictor $\mathcal{F}_{\theta_i}$ are trained against the the discriminator $\mathcal{F}_{\theta_i;\phi_i}$:

$$\mathcal{L}_{D_i}(\theta_i, \eta_i, \phi_i) = \mathcal{L}^l_{D_i}(\theta_i, \eta_i) + \mathcal{L}^u_{D_i}(\theta_i, \eta_i) - \lambda \mathcal{L}^d_{D_i}(\theta_i, \phi_i) \tag{3.8}$$

where λ is a weighting hyper-parameter. The adversarial training aims finding a balanced point $\mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i;\bar{\phi}_i}$ between the feature extractor $\mathcal{F}_\theta$ and discriminator $\mathcal{F}_\phi$ such that

$$\bar{\theta}_i, \bar{\eta}_i = \arg\min_{\theta_i, \eta_i} \mathcal{L}_{D_i}(\theta_i, \eta_i, \bar{\phi}_i) \tag{3.9}$$

$$\bar{\phi}_i = \arg\max_{\phi_i} \mathcal{L}_{D_i}(\bar{\theta}_i, \bar{\eta}_i, \phi_i) \tag{3.10}$$

By adversarial training, $\mathcal{F}_\phi$ is trained to determine the source of an example (from labeled data or augmented unlabeled data), but $\mathcal{F}_\theta$ is trained to not recognize them, so that the extracted latent space include the information which is only related to the prediction from $\mathcal{F}_\eta$ without any biased information from augmentation or the domain shift between labeled and unlabeled data. The parameters in predictor $\mathcal{F}_{\eta_i}$ and discriminator $\mathcal{F}_{\phi_i}$ are updated by gradient descent:

$$\bar{\eta}_i = \eta_i - \alpha \cdot \frac{\partial(\mathcal{L}^l_{D_i}(\theta_i, \eta_i) + \mathcal{L}^u_{D_i}(\theta_i, \eta_i))}{\partial \eta_i} \tag{3.11}$$

Algorithm 1 SSML

Randomly initialize θ, η
while not done **do**
  Sample domain subset $D' \subseteq D$
  for $i \in D'$ **do**
    Randomly sample batch of domains $D \subseteq \mathcal{D}$
    for $m \in [1, M]$ **do**
      Initialize domain network $\mathcal{F}_{\theta_i,\eta_i} \leftarrow (\theta, \eta)$
      Randomly sample support set $\{\{x_i, y_i\}, u_i\}$ and query set $\{\{\bar{x}_i, \bar{y}_i\}, \bar{u}_i\}$
      Compute cost $\mathcal{L}^l_{D_i}(\theta_i, \eta_i)$ from $\{x_i, y_i\}$ in Equation 3.2 ▷ *Supervised learning*
      Produce pseudo label $\hat{y}_i$ from $\{u_i\}$ in Equation 3.5 ▷ *Pseudo-labeling*
      Compute classification cost $\mathcal{L}^d_{D_i}(\theta_i, \phi_i)$ in Equation 3.7 ▷ *Discriminator*
      Adapt parameters $\bar{\theta}_i, \bar{\eta}_i, \bar{\phi}_i$ with gradient descent in Equations 3.11 3.12 3.13 ▷ *Adversarial training*
    end for
    Compute cost $\bar{\mathcal{L}}_{D_i}(\bar{\theta}_i, \bar{\eta}_i)$ from $\{\{\bar{x}_i, \bar{y}_i\}, \bar{u}_i\}$ in Equation 3.14
  end for
  Update θ and η with domains in D in Equation 3.15 and 3.16 ▷ *Meta-learning*
end while

$$\bar{\phi}_i = \phi_i - \alpha\lambda \cdot \frac{\partial \mathcal{L}^d_{D_i}(\theta_i, \phi_i)}{\partial \phi_i} \tag{3.12}$$

The gradient of feature extractor $\mathcal{F}_{\theta_i}$ is reversed in data source classification $\mathcal{L}^d_{D_i}(\theta_i, \phi_i)$, so that the feature extractor is trained toward two parallel directions: the decrease of prediction cost and increase of discrimination cost:

$$\bar{\theta}_i = \theta_i - \alpha(- \lambda \cdot \frac{\partial \mathcal{L}^d_{D_i}(\theta_i, \phi_i)}{\partial \theta_i} + \frac{\partial (\mathcal{L}^l_{D_i}(\theta_i, \eta_i) + \mathcal{L}^u_{D_i}(\theta_i, \eta_i))}{\partial \theta_i}) \tag{3.13}$$

This way, the feature extractor is trained to not be able to recognize if an example is from the labeled data $\{x_i\}$ or the augmented unlabeled data $\{\mathcal{A}(u_i)\}$, and the extracted information is optimized to be prediction-related regardless the bias from adding noise in augmentation.

Semi-supervised meta-learning Similar to supervised meta-learning in Equation 3.4, after all the domains in D obtained their adapted model with N steps of training, a query set with unlabeled data for each domain D_i is sampled $\{\{\bar{x}_i, \bar{y}_i\}, \bar{u}_i\} \subseteq \{\{\bar{\mathcal{X}}_i, \bar{\mathcal{Y}}_i\}, \bar{\mathcal{U}}_i\}$ and tested on its

adapted model, and adversarial training does not participate in meta-learning

$$\bar{\mathcal{L}}_{D_i}(\bar{\theta}_i, \bar{\eta}_i) = \mathcal{L}(\mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i}(\bar{x}_i), \bar{y}_i) + \mathcal{L}(\mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i}(\mathcal{A}(\bar{u}_i)), \mathcal{F}_{\theta_i,\eta_i}(\bar{u}_i)), \tag{3.14}$$

and model initialization θ and η is then updated with gradient descent:

$$\theta = \theta - \beta \cdot \frac{\partial \sum_{D_i}^{D} \bar{\mathcal{L}}_{D_i}(\bar{\theta}_i, \bar{\eta}_i)}{\partial \theta}, \tag{3.15}$$

$$\eta = \eta - \beta \cdot \frac{\partial \sum_{D_i}^{D} \bar{\mathcal{L}}_{D_i}(\bar{\theta}_i, \bar{\eta}_i)}{\partial \eta}, \tag{3.16}$$

The updated θ can then be used as model initialization in the next training episode. Algorithm 1 is the pseudo-code for our proposed SSML. Section 3.2.4 is the optimization of SSML training.

3.2.3 Time Domain Variation with SSML

In addition to heterogeneous features and label uncertainty, time-series data also has time domain variation, such as the health condition change when taking treatment, hospital transmission, etc. We propose a time domain variation framework (TDV) based on our proposed SSML and transfer learning. Equation 3.1 defines the time domain variation in a sequence $s = x^{(1:T)}$. The variation on each sequence s participates in training SSML, and the trained SSML is applied to the domain shift on s with transfer learning. According to SSML, domain D_i for sub-sequence $x^{(1:t_)}$ can obtain their adapted models $\mathcal{F}_{\theta_i;\eta_i}$, so that sub-sequence $x^{(1:t_1)}$ can be encoded and obtain its latent space $h^{(t_1)}$:

$$h^{(t_1)} = \mathcal{F}_{\theta_i}(x^{(1:t_1)}),$$

and the prediction at time t_1 is $\mathcal{F}_{\eta_i}(h^{(t_1)})$.

Assuming the domain is shifted to domain D_j for sub-sequence $x^{(t_1+1:t_2)}$ ($D_i \neq D_j$), the encoded latent space h_{t_1} from sub-sequence $x^{(1:t_1)}$ is transmitted to domain D_j feature extractor

$\mathcal{F}_{\theta_j}$:

$$h^{(t_2)} = \mathcal{F}_{\theta_j}(x^{(t_1+1:t_2)}|h^{(t_1)}).$$

By applying SSML, the homogeneous data on each sequence can be addressed independently through each domain's corresponding model, and transfer learning in the TDV framework connects the time domain variation and includes the historical information which prevents information loss. The representation of the entire sequence $x_{1:T}$ with a series of information transmissions can then be presented as

$$h^{(T)} = \mathcal{F}_{\theta_{\{D\}}}(x^{(1:T)}|h^{(t_1)}, h^{(t_2)}, ...).$$

Figure 3.1 shows the SSML-TDV framework and the training process of SSML.

3.2.4 Optimization for SSML Training

We now explain the optimization process of training our proposed SSML algorithm. The objective of SSML includes the feature extraction θ, prediction network η, and the discriminator phi:

$$\underset{\{\theta\},\{\eta\}}{\text{minimize}}\, \mathcal{L}(\theta, \eta, \phi), \quad \underset{\{\phi\}}{\text{maximize}}\, \mathcal{L}(\theta, \eta, \phi)$$

such that

$$\theta_n = \Theta_n(\theta_{n-1}),\ \eta_n = \Psi_n(\eta_{n-1}),\ \phi_n = \Phi_n(\phi_{n-1})\ \ (n \in [1, N])$$

where Θ_n, Ψ_n, Φ_n represents the gradient step of parameter optimizations at step n of the SSML adversarial training. The Lagrangian is this:

$$\mathcal{L}(\{\theta\}, \{\eta\}, \{\phi\}, \delta, \epsilon, \sigma) = \ell(\theta, \eta, \phi) + \sum_{n}^{N} \delta_n(\Theta(\theta_{n-1}) - \theta_n)$$
$$+ \sum_{n}^{N} \epsilon_n(\Psi_n(\eta_{n-1}) - \eta_n) - \sum_{n}^{N} \sigma_n(\Phi_n(\phi_{n-1}) - \phi_n)$$

where δ_n, ϵ_n and σ_n are the associated Lagrangian multipliers of step n of Θ, Ψ, and Φ. The derivatives of the last step of SSML inner loop are given as:

$$\nabla_{\theta_N}\mathcal{L} = \nabla_{\theta_N}\ell(\theta_N, \eta_N) - \nabla_{\theta_N}\ell(\theta_N, \phi_N) - \delta_N$$

$$\nabla_{\eta_N}\mathcal{L} = \nabla_{\eta_N}\ell(\theta_N, \eta_N) - \epsilon_N$$

$$\nabla_{\phi_N}\mathcal{L} = \nabla_{\phi_N}\ell(\theta_N, \phi_N) - \sigma_N$$

At each intermediate step n of SSML, the derivatives are:

$$\nabla_{\theta_n}\mathcal{L} = -\delta_n + \delta_n\nabla_{\theta_n}\Theta_{n+1}(\theta_n|\eta_N) - \delta_n\nabla_{\theta_n}\Theta_{n+1}(\theta_n|\phi_N), \;\; n \in [1, N-1]$$

$$\nabla_{\eta_n}\mathcal{L} = -\epsilon_n + \epsilon_n\nabla_{\eta_k}\Psi_{n+1}(\eta_n), \;\; n \in [1, N-1]$$

$$\nabla_{\phi_n}\mathcal{L} = -\sigma_n + \sigma_n\nabla_{\phi_k}\Phi_{n+1}(\phi_n), \;\; n \in [1, N-1]$$

Each derivative is set to zero to optimize the model:

$$\epsilon_N = \nabla_{\eta_N}\ell(\eta_N)$$

$$\epsilon_n = \epsilon_{n+1} + \nabla_{\eta_n}\Psi_{n+1}(\eta_n), \;\; n \in [1, N-1]$$

$$\sigma_N = \nabla_{\phi_N}\ell(\phi_N)$$

$$\sigma_n = \sigma_{n+1} + \nabla_{\phi_n}\Phi_{n+1}(\phi_n), \;\; n \in [1, N-1]$$

$$\delta_N = \nabla_{\theta_N} \ell(\theta_N)$$

$$\delta_n = \delta_{n+1} + \nabla_{\theta_n} \Theta_{n+1}(\theta_n|\eta_n) - \nabla_{\theta_n} \Theta_{n+1}(\theta_n|\phi_n), \;\; n \in [1, N-1]$$

3.3 Experiment

Datasets The PhysioNet Challenge 2012 dataset collects the first 48 hours of measurements after patients are admitted to the intensive-care unit (ICU) [117]. PhysioNet collects 41 variables, including 36 time-series features and five general descriptors: age, gender, height, ICU type, and initial weight. There are 4,000 labeled sequences of mortality, with 13.8 % positive cases, and another 4,000 unlabeled sequences. The hourly average value for each feature is computed, and missing data are imputed with the previous existing values. The mask of data missing is also included as extra features [13], and at the same time is used to analyze the frequency of features and determine the domains for heterogeneous features.

MIMIC-III (Medical Information Mart for Intensive Care) is a large EHR dataset collected from the intensive-care unit (ICU) [4]. MIMIC-III contains the ICU stays of over 38,000 adult patients, which includes a great number of heterogeneous EHR records. We select 17 features and discretize them to be hourly-sampled [13]. Similar to PhysioNet, the missing data is imputed with previous values. We have three classification tasks for risk prediction in MIMIC-III: physiologic decompensation (whether a patient's health will rapidly deteriorate, binary classification with 2.1 % positive), length of stay in the ICU (multi-class classification, 10 classes/buckets), and in-hospital mortality (binary classification with 8.8 % positive). For length of stay, we evaluate the models using Cohen's kappa coefficient for the inter-annotator agreement, and the mean absolute deviation (MAD) between the predicted length of stay and their reference. For the unbalanced classification tasks decompensation and in-hospital mortality, we introduce both AUROC and AUPRC for evaluation.

Data preprocessing and learning domains Feature space is an important aspect of data heterogeneity, stemming from potential diagnoses and clinical observations. For example, patients with cardiovascular diseases require more frequent monitoring of blood pressure, and oxygen sat-

uration is more important to anemia or pulmonary patients. Therefore, the distribution of features, including the presence and frequency of condition-specific features, is valuable. In order to analyze feature space with the challenge of multi-dimension data heterogeneity, we calculate the frequency of each feature and use K-means to cluster the sequences based on the combination of frequencies of all features. Each cluster then includes homogeneous sequences with similar feature frequencies and missingness, which indicates the potential similar health conditions. In medicine, a hierarchical clustering method has been applied to cluster patients [118], however, in this study we only cluster feature frequency instead of the raw values, and a comparison shows similar results between K-means and hierarchical clustering (see 2.2.3), therefore, we apply the simpler method K-means to lighten the data preprocessing. To address the problem of the uncertain labels with our proposed SSML, we randomly remove a feature as the augmentation method in Equations 3.6 and 3.7. The hourly-average values are computed and the missing data is imputed with the previous value.

Implementations and experimental details For the multi-source heterogeneity in time-series data, we first test our SSML on a simpler situation of heterogeneity: feature space and label uncertainty (SSML), and later include the time domain variation into the experiment (SSML-TDV). PhysioNet has both labeled data and unlabeled data. The labeled data is randomly split into 80% training data (20% as validation) and 20% test data for 10 rounds of experiments. Domains of heterogeneous feature frequencies are clustered separately each time for the training set and test set including labeled and unlabeled data. On MIMIC-III, considering the various length of sequences and the variation in the time domain, we build the model based on multiple 24-hour windows on each sequence. Due to the variation and uncertainty in the time domain (e.g., decompensation may happen at multiple random time points during an ICU stay), the early windows are used as the unlabeled data in SSML. PhysioNet only includes sequences with a length of 48-hour which limits the time domain variation, therefore, we only test SSML-TDM on MIMIC-III.

The SSML and SSML-TDV models are implemented on top of an LSTM model with a hidden size of 128. The sequences with heterogeneous feature spaces are clustered into 8 clusters

(domains) for PhysioNet and 18 for MIMIC-III (obtained from hyperparameters tuning). In each training episode, five optimization steps are applied on a support set (with labeled and unlabeled data) for each domain with a learning rate of 0.005, and the optimized model for each domain is then tested on another query set. The loss on the query sets from all the randomly sampled domains in this episode is collected to for meta-training with a learning rate of 0.0005. In validation and test sets, we only apply labeled data to evaluate the model performance. Please see section 3.3.3 for details of hyperparameter tuning for pseudo-labeling threshold τ, number clusters, and optimization steps. This work is implemented in Python 3.6 with PyTorch 1.3.1, Numpy 1.18, sklearn 0.21 on our server of 2 Xeon 2.2GHz CPUs, 8 GTX 1080ti GPUs, and 528 GB RAM.

Baseline Models

We test our SSML and SSML-TDV against:

- **LogsticRegression**: a logistic regression model with grid search among penalty and the regularization strength.
- **Transformer**: an attention-based model for sequential data without recurrent or convolutional mechanism [36].
- **LSTM**: an LSTM model on hourly time-series medical data with missing data imputed [13].
- **P-LSTM**: a phased LSTM model applying a time gate to regulate the access of hidden and cell state of LSTM which captures the time-series irregularity [119].
- **FixMatch**: a semi-supervised learning method producing confident pseudo-label for unlabeled data and compare with its augmentation [112].
- **MAML**: a few-shot-based meta-learning method optimizing global initialization for various tasks and rapidly adapting to any new task [40].
- **DynEHR**: a meta-learning based model for various lengths of medical data [100] (only compare with SSML-TDV for time domain variation).

3.3.1 Experiments on Heterogeneous Features and Label Uncertainty

PhysioNet

Table 3.2 represents the experimental results on PhysioNet mortality prediction task. Our

Table 3.2: Average performance (and standard deviations) on PhysioNet.

Evaluation	AUCROC	AUCPRC
LogisticReg	0.711 (0.003)	0.343 (0.005)
Transformer	0.770 (0.009)	0.405 (0.008)
LSTM	0.784 (0.010)	0.399 (0.007)
P-LSTM	0.756 (0.015)	0.368 (0.009)
FixMatch	0.789 (0.013)	0.401 (0.010)
MAML	0.809 (0.007)	0.431 (0.008)
SSML (Ours)	**0.826 (0.008)**	**0.462 (0.007)**

proposed SSML shows great improvement over all the baseline models on both AUCOC and AUCPRC. For the models not considering data heterogeneity, LSTM performs the best (compared to LogisticReg, Transformer, and P-LSTM). The comparison between MAML and LSTM shows the benefits of addressing the heterogeneous feature space, and by introducing unlabeled data, FixMatch also has an improvement to LSTM. However, both FixMatch and MAML only address a single type of data heterogeneity. For a multi-source heterogeneity situation in PhysioNet, SSML handles both the heterogeneous features and the label uncertainty, and further improves over FixMatch and MAML.

MIMIC-III

Compared to PhysioNet, MIMIC-III is a more complex dataset with various lengths of sequences. In Table 3.3, we first focus on the heterogeneous features and label uncertainty in MIMIC-III by simplifying it using the latest 24-hour data. We test MIMIC-III on three learning tasks decompensation, length-of-stay, and in-hospital mortality, and SSML performs the best for all three tasks. Compared to MAML, the improvements of SSML on decompensation and length-of-stay indicate that valuable information from introducing the unlabeled data, and the results on in-hospital mortality further show a better performed meta-learning algorithm SSML with better noise tolerance from the augmented data. When comparing SSML with LSTM and FixMatch,

Table 3.3: Average performance (and standard deviations) on MIMIC-III for heterogeneous features and label uncertainty.

Task	Decompensation		Length-of-stay		In-hospital Mortality	
Evaluation	AUCROC	AUCPRC	Kappa	MAD	AUCROC	AUCPRC
LogisticRegression	0.816	0.231	0.346	163.8	0.795	0.492
	(0.016)	(0.026)	(0.008)	(10.9)	(0.011)	(0.019)
Transformer	0.837	0.241	0.371	160.0	0.829	0.497
	(0.012)	(0.019)	(0.019)	(6.9)	(0.012)	(0.013)
LSTM	0.848	0.278	0.405	156.2	0.835	0.500
	(0.009)	(0.012)	(0.013)	(6.4)	(0.011)	(0.010)
P-LSTM	0.836	0.207	0.382	152.4	0.834	0.504
	(0.007)	(0.014)	(0.008)	(7.8)	(0.006)	(0.009)
FixMatch	0.856	0.282	0.413	157.4	0.840	0.507
	(0.008)	(0.016)	(0.016)	(7.5)	(0.004)	(0.008)
MAML	0.868	0.292	0.400	151.5	0.840	0.552
	(0.009)	(0.007)	(0.009)	(4.1)	(0.008)	(0.011)
SSML (Ours)	**0.875**	**0.330**	**0.422**	**148.6**	**0.851**	**0.575**
	(0.010)	**(0.008)**	**(0.007)**	**(4.7)**	**(0.009)**	**(0.008)**

the improvements on SSML further show that specializing the medical sequences from the feature distributions obtain better models on each homogeneous set of data, especially with unbalance dataset, obtaining higher AUCPRC values.

3.3.2 Experiments on Three-source Heterogeneity (including Time Domain Variation)

MIMIC-III

Table 3.4 represents the results of the three-source heterogeneity in MIMIC-III: heterogeneous features, label uncertainty, and time domain variation. Similar to Table 3.3, we also test three tasks and provide the averaged performances and their standard deviation. From the table, SSML-TDV performs better than all the baseline models on all the tasks. For example, SSML-TDV improves AUCPRC on decompensation by 13.2 % (0.042) over FixMatch and 11.1 % (0.036) compared

Table 3.4: Average performance (and standard deviations) on MIMIC-III full sequences with time domain variation.

Task	Decompensation		Length-of-stay		In-hospital Mortality	
Evaluation	AUCROC	AUCPRC	Kappa	MAD	AUCROC	AUCPRC
LogisticRegression	0.839	0.246	0.378	161.2	0.825	0.499
	(0.015)	(0.017)	(0.009)	(8.7)	(0.011)	(0.019)
Transformer	0.842	0.260	0.384	147.2	0.836	0.504
	(0.012)	(0.019)	(0.014)	(7.5)	(0.009)	(0.010)
LSTM	0.856	0.313	0.423	152.4	0.847	0.515
	(0.011)	(0.015)	(0.010)	(4.2)	(0.008)	(0.012)
P-LSTM	0.838	0.237	0.426	145.6	0.848	0.505
	(0.009)	(0.013)	(0.012)	(4.9)	(0.006)	(0.008)
FixMatch	0.876	0.317	0.425	151.7	0.854	0.519
	(0.004)	(0.018)	(0.006)	(5.4)	(0.007)	(0.009)
MAML	0.879	0.320	0.428	149.5	0.858	0.540
	(0.008)	(0.011)	(0.011)	(4.7)	(0.009)	(0.014)
DynEHR	0.863	0.345	0.415	137.4	0.847	0.556
	(0.008)	(0.009)	(0.016)	(7.5)	(0.006)	(0.005)
SSML-TDV(Ours)	**0.906**	**0.359**	**0.443**	**132.6**	**0.869**	**0.566**
	(0.007)	**(0.006)**	**(0.009)**	**(3.6)**	**(0.007)**	**(0.009)**

to the best baseline model MAML. SSML-TDV and MAML are both meta-learning algorithms, and SSML-TDV has an additional consistency regularization mechanism from the label uncertainty. The improvements of SSML-TDV over MAML indicate a more reliable stable model with higher noise tolerance obtained from applying this consistency regularization method to the augmented data. When compared to FixMatch which also has consistency regularization, the benefits of SSML-TDV then imply that the EHR feature distribution is a valuable aspect of heterogeneity to analyze, and modeling it can help the model better concentrate on each homogeneous set of data.

From Table 3.4, logistic regression performs the worse in all the models, and LSTM is slightly better than transformer and P-LSTM. When comparing these four static models with SSML-TDV

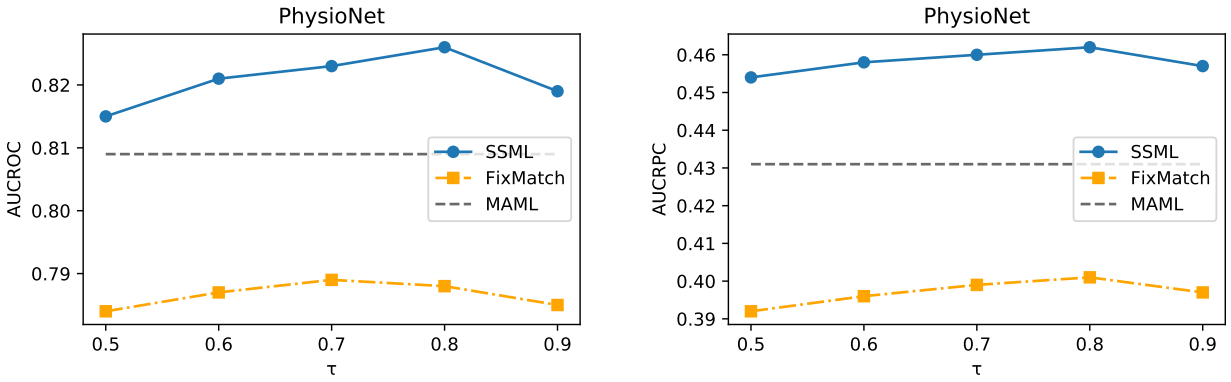
and MAML, we observe that both SSML-TDV and MAML have great improvements, especially on the AUCPRC for decompensation and in-hospital mortality, meaning a better performance on the imbalanced dataset. In addition, the improvement of SSML-TDV over SSML (in Table 3.3) shows the benefit of the transfer learning mechanism in SSML-TDV by handling the time domain variation in time-series sequences.

3.3.3 Hyperparameters Study

One important hyperparameter in our proposed SSML is the threshold τ in pseudo-labeling (Equation 6). We test the different settings for hyperparameter τ of 0.5, 0.6, 0.7, 0.8, 0.9, as well as 0 (using all produced pseudo-labels) for both PhysioNet Challenge 2012 and MIMIC-III datasets, and compare with the baseline models FixMatch (with different settings of τ) and MAML (i.e. $\tau = 1$).

Figure 3.2 shows the experiments on PhysioNet. From the figure, $\tau = 0.8$ is the optimal setting for SSML, and for FixMatch, the optimal τ is around 0.7 to 0.8. When τ is 0.5 or 0.9, the performance decreases for both SSML and FixMatch, and there is a further decrease when τ is 0. This result indicates that the threshold τ can filter out the samples with low-confidence pseudo labels, and can improve the model performance by providing high-confidence samples in consistence generalization. However, a very high value of τ can cause a decrease because too few samples are kept and the model only gets limited benefits from the very little pseudo-labeling. The performance of SSML varies between different values of τ, but all are better than MAML and FixMatch. MAML does not have the hyperparameter τ, so we only compare with its average performance.

Figure 3.3 includes the experiments for hyperparameters τ on all tasks of MIMIC-III: Figures 3.3(a), 3.3(b) are the performance comparison of AUCROC and AUCPRC for Decompensation, 3.3(c) and 3.3(d) are the comparison for In-hospital Mortality, and 3.3(e) and 3.3(f) are Kappa score and MAD for Length-of-stay. Note that the higher values of AUCROC, AUCPRC, Cohen's Kappa, and lower MAD represent better performance. The best performing τ is around 0.7. Similar to the experiment on PhysioNet, SSML and FixMatch perform the worst when τ is 0, and there is also a



(a) PhysioNet AUCROC

(b) PhysioNet AUCPRC

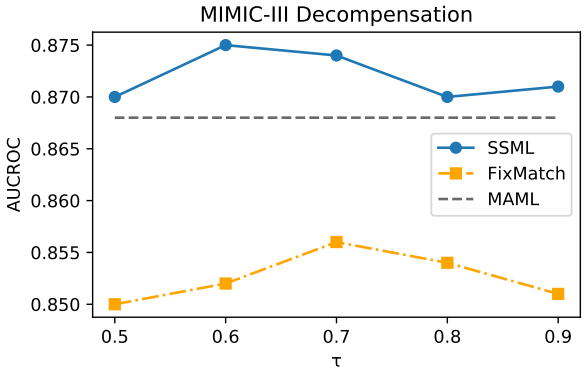
Figure 3.2: Hyperparameters comparison on PhysioNet: Blue, gray, orange represent our proposed SSML, MAML, and FixMatch repectively. X-axis is the hyperparameter τ and y-axis are the AUCROC in (a) and AUCPRC in (b). The optimal τ is around 0.8 on PhysioNet.

decrease when τ is a large value. For Decompensation, In-hospital Mortality, and Cohen's Kappa score of Length-of-stay, SSML performs better than both MAML and FixMatch for all the settings of τ. However, for MAD of Length-of-Stay, SSML is only better than MAML when τ is between 0.5 to 0.8.

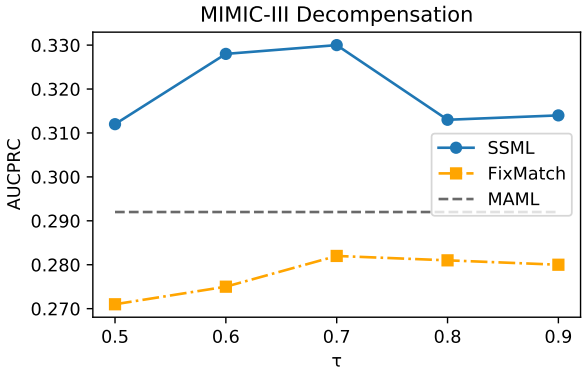
In addition to the hyperparameter τ, we also test the different number of clusters in data pre-processing, and the meta-learning steps (inner loop). We test the number of clusters between 5 to 40, and observe that the optimal setting is eight clusters for PhysioNet and 18 for MIMIC-III. The reason may come from the size of the dataset. PhysioNet only includes 4,000 labeled data and 4,000 unlabeled data, and MIMIC-III has over 38,000 patients recorded, and the bigger dataset needs more clusters. We also run experiments for the steps of inner loop optimization from 1 to 15, and the optimal step is 5 for both PhysioNet and MIMIC-III.

3.4 Limitations and Future Work

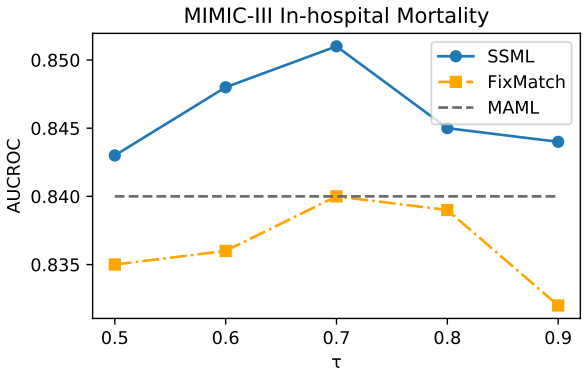
A challenge for addressing the heterogeneity in time-series data is the definition of heterogeneity. Our proposed models require pre-defined domains of heterogeneous data. In our experiment, we process the heterogeneity by computing the frequency of each medical measurement and ap-



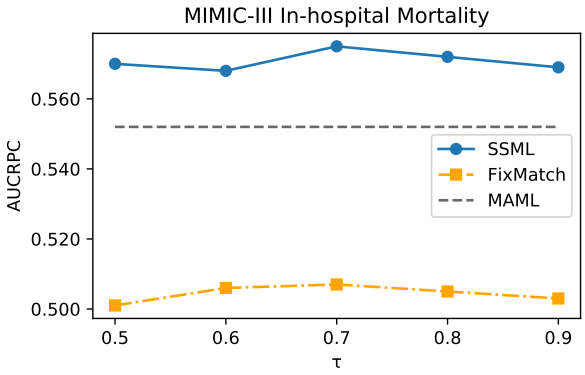
(a) Decompensation AUCROC



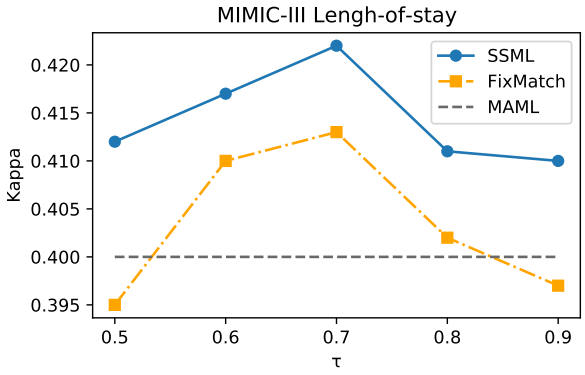
(b) Decompensation AUCPRC



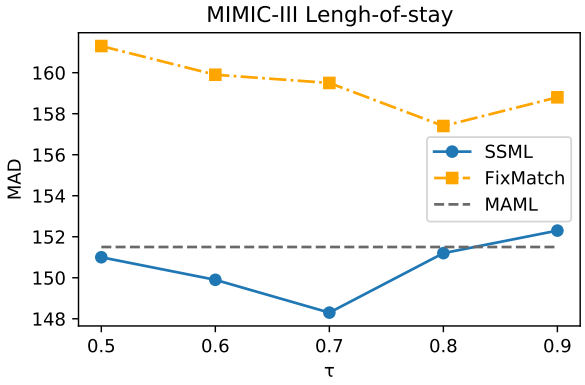
(c) In-hospital Mortality AUCROC



(d) In-hospital Mortality AUCPRC



(e) Lengh-of-stay Kappa



(f) Lengh-of-stay MAD

Figure 3.3: Hyperparameters comparison on MIMIC-III: Blue, gray, orange represent our proposed SSML, MAML, and FixMatch repectively. X-axis is the hyperparameter τ and the optimal τ is around 0.7.

plying an unsupervised clustering method to obtain the groups of patients with similar feature distributions. However, the number of clusters is manually chosen as a hyperparameter, causing the tedious work of searching for the optimal setting. In addition, clustering with a given number of clusters has difficulties handling new activities in practice, for example, a newly discovered disease (e.g., COVID-19) will all be clustered in the existing clusters. In the future, we plan to extend our SSML and SSML-TDV to a flexible number of domains. We look to apply a growing clustering method so that our model can address any new coming data.

3.5 Conclusion

Heterogeneity is a common problem in real-world applications that impedes the development of modeling, and time-series data faces the challenge of multi-source heterogeneity, including heterogeneous features, uncertain labels, and time-varying factors. Traditional machine learning techniques have difficulty addressing these heterogeneities simultaneously. In this paper, we propose a semi-supervised meta-learning (SSML) algorithm with an adversarial training mechanism for the multi-source heterogeneity challenge in time-series data. Our SSML can address the heterogeneous features and label uncertainty at the same time. In addition, for the time-varying factor, we further introduce a time domain variation framework based on our proposed SSML and transfer learning. We test our proposed models on two real-world medical datasets: PhysioNet Challenge 2012 and MIMIC-III ICU dataset, and over-perform all the baseline models.

4.1 Diabetes and Diet Monitoring

Diabetes has become one of the major diseases causing over a million death in the United States [120]. Dietary habit, as one of the most straightforward reasons of causing diabetes, has been studied and monitored [121, 122, 123, 124]. Macro nutrition prediction is of utmost importance in the field of diet monitoring and dietetics. By analyzing and predicting the intake of macronutrients such as carbohydrates, proteins, and fats, healthcare professionals can design personalized dietary plans and monitor individuals' diets. Moreover, it is essential for athletes and people engaged in high-intensity physical activities to know their macronutrient intake to optimize their performance and recovery. With the advent of technology, several tools and applications have been developed to predict macronutrient intake accurately. Hence, macro nutrition prediction has become an integral part of the healthcare industry, facilitating better health outcomes for individuals.

There has been significant research on macro nutrition prediction using continuous glucose monitors (CGMs). Instead of collecting glucose information by pricking the finger a few times every day, the non-intrusive CGMs painlessly and automatically record glucose values every 5-15 minutes. The continuous record of glucose provides the potential of tracking the effect of each meal. Machine learning has been applied to CGMs for diet monitoring [125, 62], such as carbohydrate [126]. Huo et al. proposed a multitask neural network model to predict the macro nutrition of each meal [63]. In addition to CGMs, food images are also an important resource for macro nutrition prediction and dietary tracking [127, 128], especially with the wide applications of mobile devices and systems [129, 130, 131, 132].

4.2 Why Multiple Data Modalities

Uncertain information and Heterogeneity a major challenge in macronutrient prediction and predictions with a single modality of data has limited efficacy. CGM has been applied for macronutrient prediction [62, 63], however, people's glucose response can be influenced by many factors.

The impacts of different types of food on glucose response can have significant variations [64]: Given the similar amount of carbohydrate, protein, and fat, carbohydrate raise glucose to a very high level very shortly, and then decrease very fast, while the effectiveness of fat is the lowest but lasts the longest. In addition, health condition is also an important factor in glucose response. People under diabetic conditions without treatments in general have much higher glucose after a meal than people without this condition. Therefore, macro nutrition prediction using CGM has great bias and is not always promising. The similar limitation also happens to macronutrient prediction using food images. One of the major factors that affect the information extraction from food images is the sauces. For example, a pack of creamy ranch may have around 100 calories more than Ketchup. The cooking style also influences the macronutrient prediction from food images. Many studies are limited to a certain type of food [133], such as Chinese food [134], Thai food [135] and Indonesian food [136]. Therefore, a single modality of data cannot satisfy the demand for macronutrient prediction.

4.3 Methodology: Macronutrient Prediction with Multiple Modalities of Data

4.3.1 Data Preprocessing and Feature Extraction

For every meal, we assume there exist two modalities of data in our study: time-series CGM recordings and an image of the meal. There are two types of CGM sensors: Dexcom and Libre. Dexcom records interstitial glucose levels every five minutes while Libre does it every fifteen minutes. We apply linear interpolation to process both types of CGM data to have a frequency of every minute. Linear interpolation involves estimating the missing data points in a time series by drawing straight lines between known data points. This technique is commonly used in CGM data analysis to fill in gaps in the data, which can occur due to technical issues or patient behavior. Using linear interpolation to estimate missing data, time series CGM data can provide a more complete picture of a patient's blood glucose levels, enabling healthcare professionals to make more informed decisions regarding treatment plans. Additionally, linear interpolation can help to identify trends in blood glucose levels over time, which can aid in understanding the glucose

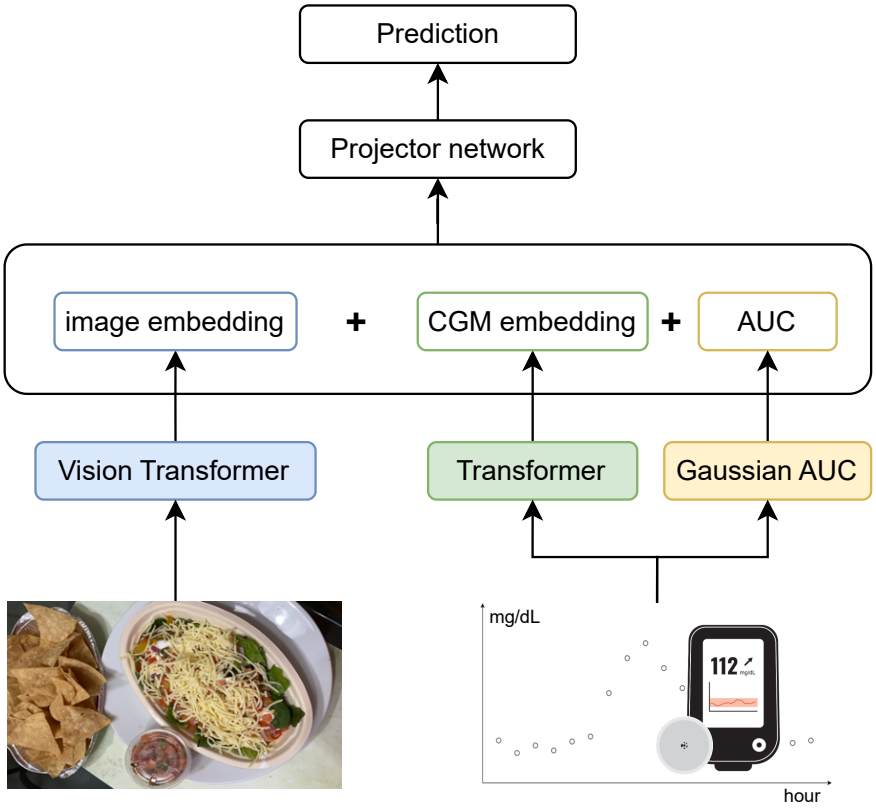


Figure 4.1: The model framework of macronutrient prediction with multiple modalities of data: image and CGMs.

changes.

In addition to linear interpolation, we also extract the Gaussian AUC features from processed CGM data. We apply five Gaussian-based kernel functions to extract the CGM Gaussian AUC features. Each Gaussian kernel is convolved with the time series data, which results in smoothed signals, and then each smoothed signal is used to obtain the statistical feature AUC value by calculating the total area under the smoothed signal curve. Figure 4.2 is an example of Gaussian AUC with five kernels. The AUC feature provides information on the overall shape and distribution of the data. The AUC feature reduces the variation of the digesting speed from different types of food and better understands the total amount of food, for example, given the same amount of food, carbohydrate raises glucose very fast to a very high level but does not last long, and the effects of fat and protein are mild but last longer. Gaussian AUC features can also be useful for time series

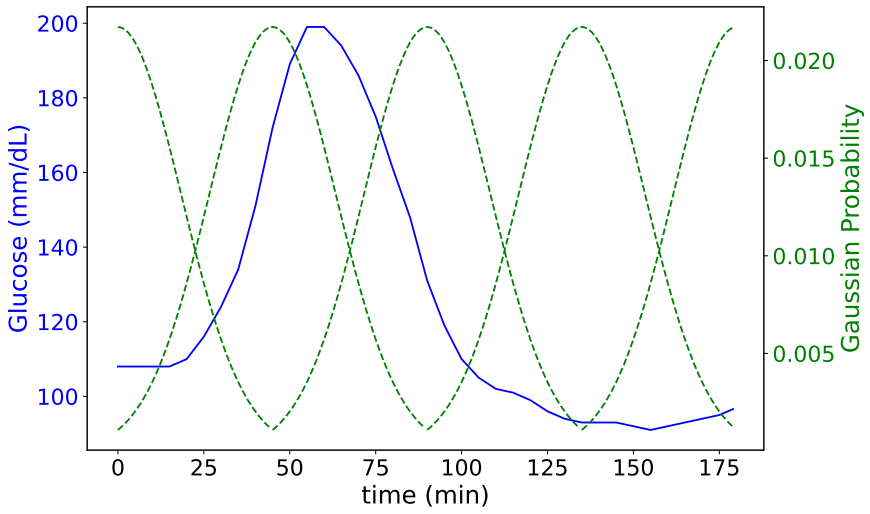


Figure 4.2: An example of five Gaussian kernels.

data that have irregular patterns or contain noise, as the Gaussian kernel can help to smooth out the signal and make it easier to identify important features.

For image data, we resize all the images to be the same size. Resizing images is important because it helps to reduce the computational complexity of image processing algorithms by reducing the dimensionality of the image. Standard size is commonly used in deep neural network models, which require input images to be of a fixed size. Reducing the image size can also help model training by reducing the impact of variations in image size and aspect ratio.

4.3.2 Macro Nutrition with Multiple Modalities Data

Deep Neural Networks for Individual Modality of Data

For time-series CGM data, we apply attention-based Transformer [36] to process the data. Transformer has been used in biomedical time-series data and shows its ability to capture critical medical information [137]. Since we are building a supervised learning task, only the encoder part is used. Compared to LSTM using a forgotten gate to capture important historical information, Transformer uses a multi-head self-attention mechanism that allows them to learn important temporal relationships and dependencies within the data, which is crucial for accurate predictions. By attending to different time points in the input sequence, attention-based transformer models can

effectively capture patterns and trends in the CGM data, which can help to improve the accuracy of predictions.

Vision transformer (Vit) [138] is becoming an increasingly popular tool for analyzing images. We utilize ViTon the images of food, leveraging its ability to learn hierarchical representations of image features to effectively analyze food images and predict the macro-nutritional content of different types of food. ViTcan learn different information from food images, such as their color, texture, and shape, which provides important information about their nutritional content.

Late Fusion

After extracting CGM embedding from Transformer, Gaussian AUC features, and image embedding from ViT, all the information will be used to make macro nutrition predictions. We use the late fusion approach from Shukla et al. [139] to aggregate the different resources of information by designing a projector network. The CGM embedding, AUC features, and image embedding are connected through the fully-connected projector network for a more comprehensive representation of the data and then make the final predictions. The late fusion approach allows each modality of data to be processed independently, which can result in more accurate and robust embeddings. In our study, CGM and images may have very different structures and features, and processing them separately allows each modality to be optimized for its own unique characteristics, and the concatenation of them effectively leverages the complementary information from each modality to improve their performance. Figure 4.1 shows the general structure of our proposed model.

4.3.3 Predictive Tasks and Evaluation Metrics

In this study, we predict calories and carbohydrates (carbs) for each meal as two regression tasks. To evaluate the performance of these models, two common metrics are Root Mean Squared Relative Error (RMSRE):

$$RMSRE = \sqrt{\frac{1}{n}(\frac{y - \bar{y}}{\bar{y}})^2}$$

where n is the number of samples, y is the predicted value, and $\bar{y}$ represents the ground truth. RMSRE measures the average error between the predicted and true values, normalized by the true

value, while correlation measures the strength and direction of the linear relationship between the predicted and true values. RMSRE is particularly useful in evaluating the accuracy of the model in terms of relative errors, which is important when dealing with nutritional data that can vary widely in magnitude. A lower RMSRE indicates that the model is able to predict calorie and carbohydrate values with greater accuracy and precision.

4.4 Experiments and Results

4.4.1 Dateset

Our data was collected from a study trial on 29 participants for continuous 10 days. Each participant has the biographic information recorded, and there are 11 healthy participants, 12 pre-diabetic, and 6 people with T2 diabetes. Both Dexcom G6 and Freestyle Libre are used to collect the glucose records. Dexcom has a frequency of roughly five minutes, and Libre records a data point around every 15 minutes. During the study, each participant was asked to take photos of all the meals and input the logs for them.

4.4.2 Experiment Setup

We set up our experiments from two aspects: data resources and models. First, we compare the model performance using CGM data only, image data only, and both CGM and image data, intended to understand if the multiple modalities of data actually bring benefits to the modeling. For CGM data, we choose a logistic regression and tree-based XGBoost, and two deep learning models LSTM and Transformer. For image data, five deep learning models are selected: VGG16, VGG19, Resnet18, Resnet50, and vision transformer (Vit). For the multiple modalities of data, we test different combinations of the two deep learning models for CGM and all five models for image data, which includes our proposed model of Transformer-ViT. The late fusion mechanism is applied for all ten models for multiple modalities data.

Hyperparameter tuning is applied to all the models, including the number of heads and layers in transformer and ViT, the hidden size, and the dropout rate. We also applied the activation function ReLU for the projector layers of Late fusion. We select the best model and run ten repeated

experiments for each model. In each experiment, we shuffle all the meals, and randomly select 60 % data for training, 20 % for validation, and 20 % for testing. The mean RMSRE and its standard deviation are calculated based on all ten experiments for each model.

Table 4.1: Macronutrient Prediction Performance Comparison among Different Data Modalities and Models

Data	Model	Calorie	Carbs
CGM-only	Logistic Regression	0.72 (0.11)	0.88 (0.10)
	XGBoost	0.54 (0.06)	0.66 (0.07)
	LSTM	0.37 (0.03)	0.45 (0.02)
	Transformer	0.37 (0.04)	0.47 (0.03)
Image-only	VGG16	0.42 (0.04)	0.50 (0.02)
	VGG19	0.42 (0.02)	0.53 (0.03)
	ResNet18	0.42 (0.03)	0.47 (0.04)
	ResNet50	0.41 (0.03)	0.49 (0.01)
	ViT	0.43 (0.02)	0.49 (0.02)
CGM-image	LSTM-VGG16	0.40 (0.03)	0.42 (0.03)
	LSTM-VGG19	0.36 (0.04)	0.44 (0.02)
	LSTM-ResNet18	0.39 (0.02)	0.42 (0.01)
	LSTM-ResNet50	0.39 (0.02)	0.40 (0.01)
	LSTM-ViT	**0.33 (0.01)**	0.40 (0.01)
	Transformer-VGG16	0.35 (0.02)	0.41 (0.03)
	Transformer-VGG19	0.37 (0.03)	0.44 (0.03)
	Transformer-ResNet18	0.40 (0.02)	0.43 (0.02)
	Transformer-ResNet50	0.39 (0.01)	0.40 (0.02)
	Transformer-ViT	**0.33 (0.01)**	**0.39 (0.01)**

4.4.3 Result and Analysis

Table 4.1 shows the results of our experiment. When comparing the different data resources, we observe that using both CGM and image data has a significant improvement over either CGM only or image only. Our proposed model improves the performance of calorie prediction by 10.8

% compared to the best CGM model, and 19.5 % to the best image model. For carbohydrate prediction, our model beats the best CGM model by 13.3 % and the best image model by 17.0 %.

When focusing on model selection, we observe that the two deep learning models LSTM and transformer for CGM data perform much better than logistic and XGBoost. This result shows better robustness of the deep learning models than traditional machine learning models, and the variation among subjects could be the reason for the low performance of traditional machine learning models. The five image models do not make a significant difference in calorie prediction, however, ViTshows its benefits when applied to CGM data. It is interesting that ViT, by itself, is not better than ResNet in carbohydrate prediction, but again, performs better with the assistance of CGM data.

4.5 Limitation and Future Work

In this study, we build general macronutrient prediction models with 29 participants. The amount of training data may not be sufficient for deep neural networks. In the future, we plan to collect more data for our model training. Our models use one image of each meal as input, however, sometimes people may not finish the entire dish, and therefore one image of a meal could cause some bias. We will extend our model in the image part of processing images for both before and after eating, which only the actual intaken food is used in predictions. In addition, the variation among subjects could be a factor challenging the modeling. We plan to also build personalized models with the technique of transfer learning, so that all subject can have their own adapted individual models.

4.6 Conclusion

In this study, we propose a macronutrient prediction model using both CGM and food image data. A transformer is used for CGM data extraction, and vision transformer is applied for image data. In addition, we also extract Gaussian AUC features from CGM data, in order to better understand the accumulated glucose change. All the features are aggregated through a projector using the late fusion mechanism. We test our model on two regression tasks: calorie and carbohydrate.

The experimental results show that our macronutrient prediction model with multiple modalities of data has significant improvement over models with a single data modality. Also, our proposed model outperforms all the baseline models.

5. CLINICAL HETEROGENEITY TRANSLATION

5.1 Heterogeneous Health Conditions in Clinic

Understanding the diverse and intricate health conditions of patients is of utmost importance for both doctors and machine learning modeling. Physicians rely on a thorough comprehension of a patient's heterogeneous health conditions to provide an accurate diagnosis, devise an appropriate treatment plan, and monitor the effectiveness of therapy. On the other hand, machine learning models that aim to predict disease outcomes and provide personalized treatment recommendations heavily rely on an understanding of the variability and heterogeneity of patient health conditions. By identifying and accounting for the unique features of individual patients, machine learning models can improve the accuracy of predictions and enhance the efficacy of personalized treatment recommendations. Therefore, it is essential to gain a deep understanding of the complex and diverse nature of patient health conditions to enhance both clinical practice and machine learning modeling.

5.1.1 Clinical Prototypes for Heterogeneity Translation

The interpretation of patients' health conditions in the clinic is complicated and difficult. Personalized models have been developed to address the complexity of hospitalized patients' health conditions, as evidenced by research conducted by Suo et al. and Liu et al. [65, 66]. Oikonomou et al. [67] proposed a strategy for phenomapping that uses information from all trial participants to phenotype individuals. However, personalized models have limited training data, and training multiple models for each patient is not optimal, even with transfer learning. Moreover, personalized models do not assist in interpreting patients' health conditions or enhance decision-making based on the experiences of similar patients. To address this issue, clustering could be a potential solution for interpreting patients' health conditions and identifying similar patients. However, a problem is that the clusters from unsupervised learning overlap sometimes and usually do not have a clear boundary, and for many samples located between two or more clusters, it is hard to deter-

mine which exact cluster such samples belong to. Additionally, it is very difficult to comprehend the meaning of each cluster, especially considering the fact that the clustering outcomes can be significantly influenced by hyperparameter settings, such as determining the number of clusters.

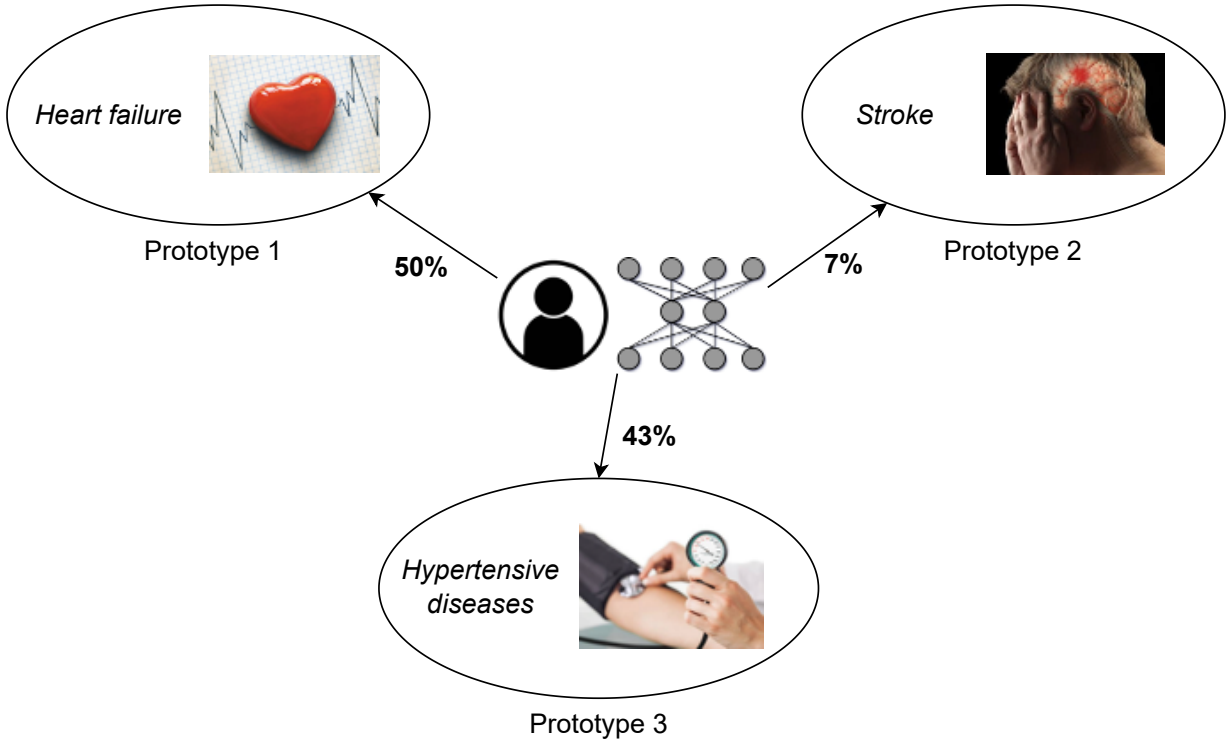


Figure 5.1: An example of the similarity from a patient to disease prototypes.

Meta-learning has been used to create fewer general models that can be applied across different personal settings, but these methods pre-define patients into certain domains and may overlook potential cross-domain patients [41]. Previous works, such as prototypical networks and patient similarity prediction models, do not have representative prototypes or flexible alignments for the heterogeneous health conditions of patients. Crabbe et al. [70] introduced a latent space for selecting some individual patients as prototypes and calculating the similarity between a new patient and these prototypes, but it is unclear how the prototypes are selected or if they are representative. This study proposes meta-prototype networks that leverage patient heterogeneity through trainable pro-

totypes to develop risk prediction models and to interpret patients' health conditions. Inspired by this study, we propose a prototype-based network that focuses on cardiovascular diseases (CVD) and defines each domain as a group of diseases, such as hypertensive diseases. The prototype of each domain is used as a representation, and for each patient, we determine the similarity to each prototype as the cross-domain diagnosis. An example of this can be seen in Figure 5.1, where the similarity of a patient to the prototypes is 50% for heart failure, 43% for hypertensive diseases, and 7 for stroke. Using this prototype-based network, we can interpret patients' health conditions easily and make aggregated predictions with each prototype's associated prediction model.

5.1.2 Challenges for Training Prototypes

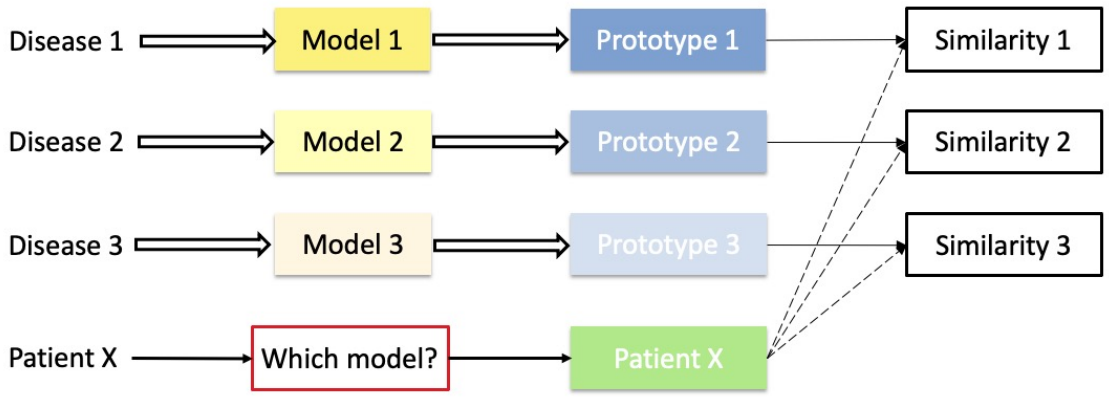


Figure 5.2: Train prototypes with individual models. There will not be a fair comparison among multiple prototypes for a new patient.

The process of constructing a predictive model based on prototypes involves two key stages. The initial step is to identify representative prototypes, followed by determining which prototypes a particular patient belongs to. One approach to obtain prototypes is to calculate the centroid of the embedding from all data points associated with a given disease, using the diagnosed patients in the training set. The embedding of each data point can be learned from self-supervised learning methods such as auto-encoder [140], knowledge reasoning [141, 142, 143], and intermediate output from a prediction model. Subsequently, the similarity score or distance between a new patient

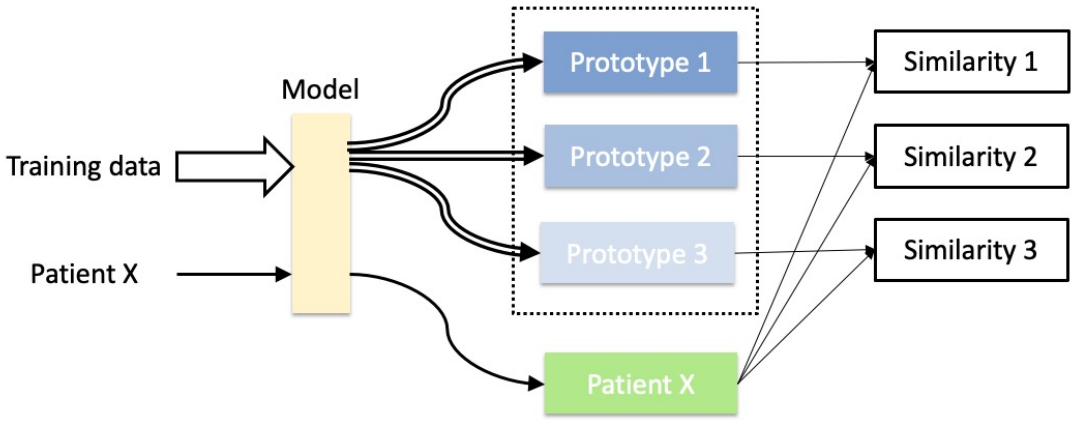


Figure 5.3: Train prototypes with individual models. There will not be a fair comparison among multiple prototypes for a new patient.

and these prototypes can be utilized to determine the degree of alignment with each one, thereby calculating the weights.

Nevertheless, several issues exist with this approach. The primary concern revolves around selecting suitable models for obtaining embeddings and subsequently calculating the centroids. If each disease requires a customized model, as depicted in Figure 5.2, then determining a patient's alignment with each prototype becomes difficult. This is because a fair comparison is required between different prototypes, and no shared model is available for processing the patient. On the other hand, if prototypes are generated using a single model for all diseases, as shown in Figure 5.3, alignment weights can be calculated. However, since there is no tailored model for each disease, these alignment weights do not contribute to prediction tasks.

In the subsequent section, we present our proposed solution that addresses the aforementioned challenges. To overcome these issues, we leverage meta-learning for model adaptation during prediction tasks. Additionally, a flexible and trainable prototype network is introduced, which replaces the manual calculation of centroids for obtaining embeddings and the calculation of prototype alignment weights of a patient.

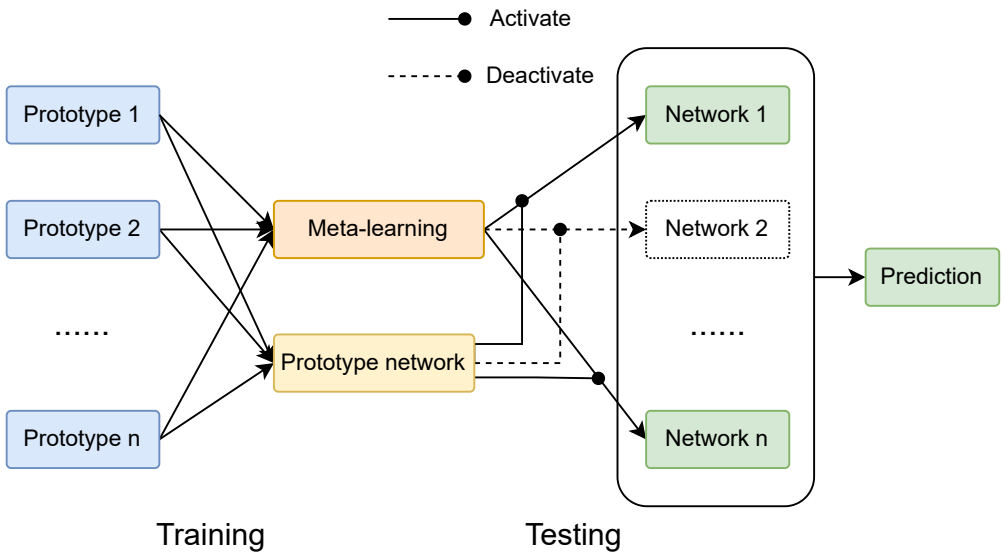


Figure 5.4: Meta-prototype framework. The prototypes are trained through meta-learning, and a prototype network is trained for prototype alignment. During testing, the prototype network decides which prototype-specific network is activated for the final prediction.

5.2 Methodology

In this section, we introduce our work in two parts: training meta-prototype and generating risk prediction with our meta-prototype. Figure 5.4 illustrates a framework of our model.

5.2.1 Meta-prototype training

Given a model $\mathcal{F}$ with a feature extractor $\mathcal{F}_\theta$ and a predictor $\mathcal{F}_\eta$, θ and η are used to indicate their parameters respectively. In a time-series setting, we apply an LSTM for the feature extractor and fully-connected layers for the predictor. For a data point x and its label y, the learning cost of our model is represented as:

$$\mathcal{L}_{\theta,\eta} = \mathcal{L}(\mathcal{F}_{\theta;\eta}(x), y). \tag{5.1}$$

Let D be a set of prototypes. In each episode of training the meta-prototype, a subset D' of prototypes is randomly sampled ($D' \subseteq D$). For each prototype i in D', a model $\mathcal{F}_{\theta_i;\eta_i}$ is first initialized from the meta-learner $\mathcal{F}_{\theta;\eta}$, and the cost $\mathcal{L}_{\theta_i;\eta_i}$ for this prototype can be calculated according to Equation 5.1 on a randomly sampled support set. The model for prototype i can then

be adapted to $\mathcal{F}_{\bar{\theta}_i,\bar{\eta}_i}$ from $\mathcal{L}_{\theta_i;\eta_i}$ with a few steps:

$$\bar{\theta}_i = \theta_i - \tau \nabla_{\theta_i} \mathcal{L}_{\theta_i;\eta_i},$$

$$\bar{\eta}_i = \eta_i - \tau \nabla_{\eta_i} \mathcal{L}_{\theta_i;\eta_i},$$

where τ is a learning rate.

After the adapted model is obtained, a query set from prototype i is then sampled and applied on $\mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i}$ to calculate a cost $\mathcal{L}_{\bar{\theta}_i;\bar{\eta}_i}$ from Equation 5.1.

With the meta-learning-based training approach for the prediction models of multiple prototypes, it is still not clear what these prototypes are. Instead of using the mean of the embedded examples [68] or a certain example [70], we introduce a linear prototype network $\mathcal{F}_\phi$, a fully-connected network without a bias, as the trainable prototypes, and each column ϕ_j can represent a prototype. In each training episode, a set of data points x and their prototype label c are sampled. The representation of x is calculated from the extractor $\mathcal{F}_\theta$ (without any adaptation, in order to have a fair comparison among different prototypes), and the prototype network $\mathcal{F}_\phi$ is used to align the data to certain prototypes. The prototype network can be trained from

$$\hat{\mathcal{L}}_{\theta,\phi} = \mathcal{H}(\mathcal{F}_{\theta;\phi}(x), c),$$

where $\mathcal{H}$ denotes a cross-entropy loss function and c is a ten-class cardiovascular disease phenotype for each patient.

After collecting the prototype classification cost $\hat{\mathcal{L}}_{\theta;\phi}$ and the query set cost $\mathcal{L}_{\bar{\theta}_i;\bar{\eta}_i}$ from all the sampled prototypes D', the meta-learner $\mathcal{F}_\theta$, $\mathcal{F}_\eta$, and prototype network $\mathcal{F}_\phi$ can be optimized as:

$$\theta = \theta - \mu(\sum_{i}^{D'} \nabla_\theta \mathcal{L}_{\bar{\theta}_i;\bar{\eta}_i} + \nabla_\theta \hat{\mathcal{L}}_{\theta;\phi}),$$

$$\eta = \eta - \mu \sum_{i}^{D'} \nabla_\eta \mathcal{L}_{\bar{\theta}_i;\bar{\eta}_i}, \quad \phi = \phi - \mu \nabla_\phi \hat{\mathcal{L}}_{\theta;\phi},$$

where μ is another learning rate.

5.2.2 Risk prediction with meta-prototype

Before making predictions, the prototype-specific network $\mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i}$ for each prototype is first adapted from the trained meta-learner with their corresponding support set. Given a data point x, the prototype alignment β is calculated from $\mathcal{F}_{\theta;\phi}$, and then calculate a mask α_i for each prototypes i ($i \in D$) using Top-k [144, 145]:

$$\beta = \mathcal{F}_{\theta;\phi}(x), \quad \alpha_i = \begin{cases} 1 & \text{if } \beta_i \text{ in top } k \text{ value of all } \beta \\ 0 & \text{otherwise.} \end{cases}$$

A final prediction can be generated from the prototype masks

$$p(x) = \sum_{i}^{D} \alpha_i \cdot \mathcal{F}_{\bar{\theta}_i;\bar{\eta}_i}(x)$$

5.3 Experiments

5.3.1 Dataset and data preprocessing

Medical Information Mart for Intensive Care (MIMIC-III) is a publicly available EHR dataset [4] which collects 53,423 adult patients admitted to Beth Israel Deaconess Medical Center intensive care units (ICUs) between 2001 and 2012. We apply our proposed method meta-prototype on MIMIC-III, focusing on cardiovascular diseases. From the MIMIC-III ICD-9 diagnosis table and its HCUP CCS category [13], ten cardiovascular diseases (or conditions common to cardiovascular-related complications) are retained, as shown in Table 5.1. We treat each disease here as a prototype when building our meta-prototype, and a patient may be aligned to one or multiple prototypes.

There are 17 charted observations and laboratory measurements selected formatting 76 features (one-hot encoding for categorical measures and numeric values for continuous measurements) [13] as the input of our model. The irregular data is split into a series of one-hour time windows without overlapping. The average values are calculated if there is more than one data point in a window,

Table 5.1: Cardiovascular Condition Categories

Acute and unspecified renal failure
Acute cerebrovascular disease
Acute myocardial infarction
Cardiac dysrhythmias
Chronic kidney disease
Congestive heart failure; nonhypertensive
Coronary atherosclerosis and related
Essential hypertension
Hypertension with complications
Shock

and missing data is imputed with the most recent values. In order to apply mini-batch optimization in training, zeros are padded at the end of shorter sequences.

5.3.2 Prediction tasks and evaluation

We test our model on three prediction tasks based on MIMIC-III: decompensation (rapid deterioration of patient conditions), the length of stay in the intensive care unit (ICU), and in-hospital mortality. Decompensation and in-hospital mortality are binary classification tasks. Decompensation has 13.5% of positive examples, and in-hospital mortality has 2.1%. Therefore, in addition to the evaluation metric of AUROC, we also introduce AUPRC to evaluate these two imbalanced classification tasks. The length-of-stay is framed as a multi-class classification problem [13]. Cohen's Kappa score and MAD are used to evaluate this task.

5.3.3 Model implementation and baseline models

In the experiments, we set the hidden size of the LSTM-based feature extractor $\mathcal{F}_\theta$ to be 128, and apply a one-layer fully-connected network for the predictor $\mathcal{F}_\eta$. As we discussed in the previous sections, a fully-connected network without bias is used as the prototype network $\mathcal{F}_\phi$. The

Table 5.2: Average performance (and standard deviations) on MIMIC-III

Task	Decompensation		Length-of-stay		In-hospital Mortality	
Evaluation	AUROC	AUPRC	Kappa	MAD	AUROC	AUPRC
LogisticRegression	0.816	0.231	0.346	163.8	0.795	0.492
	(0.016)	(0.026)	(0.008)	(10.9)	(0.011)	(0.019)
Transformer	0.837	0.241	0.371	160.0	0.829	0.497
	(0.012)	(0.019)	(0.019)	(6.9)	(0.012)	(0.013)
LSTM	0.848	0.278	0.405	156.2	0.835	0.500
	(0.009)	(0.012)	(0.013)	(6.4)	(0.011)	(0.010)
P-LSTM	0.836	0.207	0.382	152.4	0.834	0.504
	(0.007)	(0.014)	(0.008)	(7.8)	(0.006)	(0.009)
MAML	0.837	0.269	0.404	152.7	0.836	0.535
	(0.007)	(0.011)	(0.005)	(4.9)	(0.04)	(0.007)
Meta-prototype	**0.858**	**0.311**	**0.413**	**141.9**	**0.856**	**0.555)**
	(0.008)	**(0.009)**	**(0.006)**	**(5.5)**	**(0.005)**	**(0.008)**

dataset is split into a 70% training, a 15% validation set, and a 15% test set, with 10 repeated experiments. In each training episode, we randomly sample five prototypes and train each prototype-specific model $\mathcal{F}_{\theta_i;\eta_i}$ with five steps, and the model adaptation when making prediction has five steps as well. The prototype-specific model training has a learning rate τ of 0.005, and the training of the meta-learner has a learning rate μ of 0.0005. For the Top-k mechanism, we run hyperparameter tuning experiments and set k to be four. This study is implemented in Python 3.6, PyTorch 1.3.1, NumPy 1.18, scikit-learn 0.21 on the server of 2 Xeon 2.2GHz CPUs, 8 GTX 1080ti GPUs, and 528 GB RAM.

To understand the performance of meta-prototype, we compare our model with five baseline models: a logistic regression model with grid search for penalty and regularization strength, an attention-based transformer model [36], an LSTM model, a phased LSTM (p-LSTM) for time-series irregularity [146], and a meta-learning model [40, 100] with fixed prototypes obtained di-

rectly from cardiovascular diseases phenotype labels (MAML). The transformer model has query and value sizes of eight, two heads, two blocks, and attention size 12. The LSTM and p-LSTM models both have hidden size 128, and the MAML model is built based on the same structure of LSTM. The learning rates for deep neural network models are 0.0005.

5.3.4 Experimental results

Table 5.2 shows the results of our experiments. For MAML and our meta-prototype, we calculate the average performance from all the prototypes (diseases) and their standard deviations. From the table, our meta-prototype has great improvements on all three tasks over all baseline models. For the binary classification tasks decompensation and in-hospital mortality, our model has higher values for both AUROC and AUPRC, especially AUPRC. The significant improvement on AUPRC shows the ability of our model to address the imbalanced datasets and implies a higher sensitivity of our model in predicting at-risk patients and a potential for better performance in saving patients' lives. For length-of-stay, the higher value of the Cohen's Kappa score of our model indicates higher inter-annotator agreements between our predictions and the ground truth, and the lower MAD value additionally reinforces the lower errors of predicting the remaining length of stay in ICUs. When comparing the meta-learning-based models MAML and our meta-prototype with their base model LSTM, we can observe that MAML is sometimes even worse than the LSTM (on decompensation), showing the limitation of vanilla meta-learning in addressing the cross-domain situation, and further indicating the flexibility of meta-prototype in prototype alignment in complex situations.

Figure 5.5 is a heatmap of the Top-k masking in the task of in-hospital mortality. Y-axis is the ten cardiovascular prototypes, and x-axis is the predicted masking from the prototype network and Top-4 mechanism. We observe that the prototype network can predict various prototypes.

5.4 Limitions and Future Work

In this study, we evaluate our proposed model within cardiovascular diseases, and we plan to expand the experiments to other diseases, or a cross-domain setting among different types of

diseases (e.g., cardiovascular and diabetes). In addition, the current prototype network is limited to a pre-defined number of prototypes and therefore needs to be re-trained if a new condition is included. In the future, we also look forward to modifying the prototype network to be flexible to growing prototypes.

5.5 Conclusion

Patients in the hospital often have complex health conditions, such as multiple diseases, complications, or underlying diseases. A generalized model cannot represent the variation among different diseases, and personalized models are limited to the amount of training data and tedious training process. In this paper, we propose meta-prototype networks, applying meta-learning to similar patients, and then introduce a trainable prototype network to represent the prototypes. We test our meta-prototype on cardiovascular diseases in MIMIC-III, and outperform on all three prediction tasks, especially in predicting risky patients.

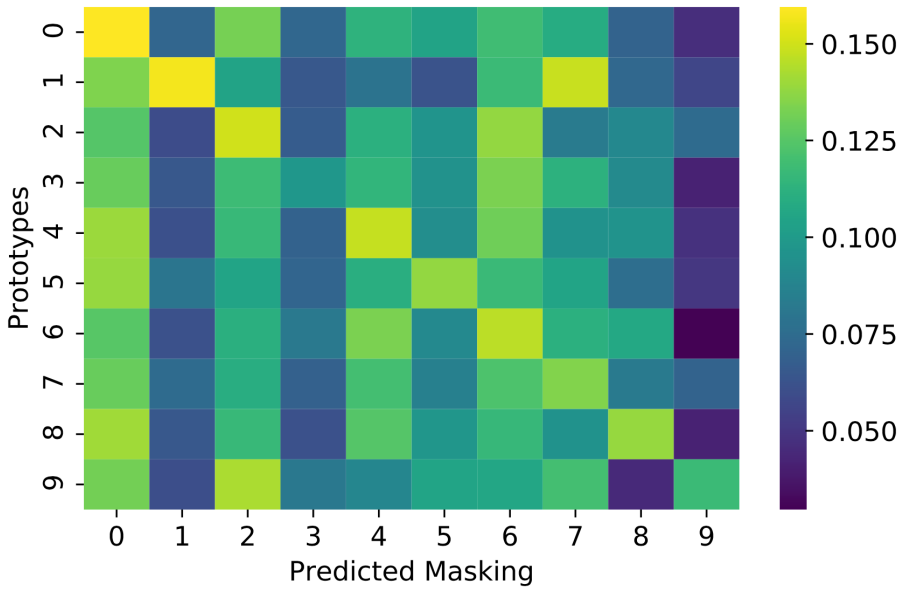


Figure 5.5: A heatmap for in-hospital mortality Top-k masking

6. CONCLUSION

This dissertation presents solutions of building flexible models for heterogeneous time-series biomedical data. Biomedical applications face various forms of data heterogeneity such as variations among subjects, irregular feature space, time domain variation, unlabeled data, and multiple modalities of data. These forms of data heterogeneity can occur either individually or simultaneously, presenting a significant challenge for model development. To tackle these challenges, we propose various flexible models that can be rapidly adapted using transfer learning, meta-learning, adversarial training, and semi-supervised learning techniques. This dissertation provides valuable contributions to the field of biomedical data analysis and can potentially lead to improved clinical decision-making and patient outcomes.

We first attempt to address the various data heterogeneity problems individually, focusing on three forms: heterogeneous data distribution, irregularly sampled time-series data, and time domain variation. We propose DANN to import other subjects' information in a personalized model, without adding personal bias to the model, and are able to obtain models meeting ISO standard with only three minutes of training data. For the irregularly sampled time-series data, we first apply clustering to analyze the irregularity, attempting to group patients with similar health conditions, and then apply meta-learning to build adaptive models for each group of patients. For the time domain variation, we propose DynEHR to address the various lengths of EHR data as a protocol for dynamic model adaptation. DynEHR uses meta-learning to train an optimized initialization and learning the optimization process, so that it can be easily adapted and applied to any duration of an ICU admission. In the future, we consider expanding the DANN approach to include more subject features and to find a metric for the subject-domain space in order to choose similar subjects prior to adapting a model given new subject with targeted minimal training data, and test our DynEHR on other data heterogeneity in EHRs, such as the sampling frequency of vitals.

Heterogeneity is a ubiquitous challenge in real-world applications that often complicates modeling, and time-series data is no exception. Time-series data faces the challenge of multi-source

heterogeneity, including heterogeneous features, uncertain labels, and time-varying factors. After addressing the individual heterogeneity problems, we then tackle the multi-source heterogeneity simultaneously. Traditional machine learning techniques struggle to address these heterogeneities simultaneously. To overcome this challenge, we propose a semi-supervised meta-learning (SSML) algorithm with an adversarial training mechanism that can handle the multi-source heterogeneity challenge in time-series data. Our SSML algorithm can simultaneously address the challenges of heterogeneous features and label uncertainty. Moreover, we introduce a time domain variation framework based on our proposed SSML and transfer learning to address the time-varying factor. We evaluate our proposed models on two real-world medical datasets: PhysioNet Challenge 2012 and MIMIC-III ICU dataset and demonstrate superior performance over all the baseline models.

With the advancement of hardware and mobile devices, incorporating multiple modalities of data in real-world applications has become feasible. In this dissertation, we propose a novel macronutrient prediction model that integrates continuous glucose monitoring (CGM) and food image data. Our model employs a transformer to extract CGM data and a vision transformer to process image data. We also extract Gaussian AUC features from the CGM data to capture the accumulated glucose change. To combine the features from both modalities, we use the late fusion mechanism with a projector. We evaluate our model on two regression tasks: calorie and carbohydrate prediction. The experimental results demonstrate that our proposed model, leveraging multiple modalities of data, outperforms single-modality models and all baseline models, achieving significant improvements in both prediction tasks.

In clinical settings, it is crucial to understand how heterogeneity affects patients and how flexible models can be applied. Hospitalized patients often have complex health conditions, including multiple diseases, complications, and underlying conditions. A generalized model may not be able to account for the variability among different diseases, while personalized models are limited by the amount of training data and the laborious training process. To address this, we propose the use of meta-prototype networks, which utilize meta-learning to identify similarities among patients and train a prototype network to represent the prototypes. We evaluate the effectiveness of our

proposed method on predicting cardiovascular diseases in the MIMIC-III dataset, and demonstrate superior performance on all three prediction tasks, particularly in identifying high-risk patients.

REFERENCES

[1] W. V. Padula and M. J. Sculpher, "Ideas about resourcing health care in the united states: can economic evaluation achieve meaningful use?," *Annals of Internal Medicine*, vol. 174, no. 1, pp. 80–85, 2021.

[2] R. Gilgen-Ammann, T. Schweizer, and T. Wyss, "Rr interval signal quality of a heart rate monitor and an ecg holter at rest and during exercise," *European journal of applied physiology*, vol. 119, no. 7, pp. 1525–1532, 2019.

[3] B. Reeder and A. David, "Health at hand: A systematic review of smart watch uses for health and wellness," *Journal of biomedical informatics*, vol. 63, pp. 269–276, 2016.

[4] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[5] M. Basza, B. Krzowski, P. Balsam, M. Grabowski, G. Opolski, and L. Kołtowski, "An apple watch a day keeps the doctor away?," *Cardiology Journal*, vol. 28, no. 6, pp. 801–803, 2021.

[6] M. Smieszek, A. Kindermann, A. Amr, B. Meder, and C. Dieterich, "An apple watch dashboard for highmed heart insufficency patients," in *German Medical Data Sciences 2021: Digital Medicine: Recognize–Understand–Heal*, pp. 146–155, IOS Press, 2021.

[7] D. Ayata, Y. Yaslan, and M. E. Kamasak, "Emotion recognition from multimodal physiological signals for emotion aware healthcare systems," *Journal of Medical and Biological Engineering*, vol. 40, no. 2, pp. 149–157, 2020.

[8] E. Mbunge, B. Muchemwa, J. Batani, *et al.*, "Sensors and healthcare 5.0: transformative shift in virtual care through emerging digital health technologies," *Global Health Journal*, vol. 5, no. 4, pp. 169–177, 2021.

[9] Z. D. King, J. Moskowitz, B. Egilmez, S. Zhang, L. Zhang, M. Bass, J. Rogers, R. Ghaffari, L. Wakschlag, and N. Alshurafa, "Micro-stress ema: A passive sensing framework for detecting in-the-wild stress in pregnant mothers," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 3, no. 3, pp. 1–22, 2019.

[10] G. Ogbuabor and R. La, "Human activity recognition for healthcare using smartphones," in *Proceedings of the 2018 10th international conference on machine learning and computing*, pp. 41–46, 2018.

[11] A. Subasi, K. Khateeb, T. Brahimi, and A. Sarirete, "Human activity recognition using machine learning methods in a smart healthcare environment," in *Innovation in health informatics*, pp. 123–144, Elsevier, 2020.

[12] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognition*, vol. 108, p. 107561, 2020.

[13] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Scientific data*, vol. 6, no. 1, pp. 1–18, 2019.

[14] L. Zhang, Z. King, B. Egilmez, J. Reeder, R. Ghaffari, J. Rogers, K. Rosen, M. Bass, J. Moskowitz, D. Tandon, *et al.*, "Measuring fine-grained heart-rate using a flexible wearable sensor in the presence of noise," in *2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 160–164, IEEE, 2018.

[15] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, "Cuffless blood pressure estimation algorithms for continuous health-care monitoring," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 4, pp. 859–869, 2016.

[16] Y.-L. Zheng, B. P. Yan, Y.-T. Zhang, and C. C. Poon, "An armband wearable device for overnight and cuff-less blood pressure measurement," *IEEE transactions on biomedical engineering*, vol. 61, no. 7, pp. 2179–2186, 2014.

[17] S. S. Thomas, V. Nathan, C. Zong, K. Soundarapandian, X. Shi, and R. Jafari, "Biowatch: A noninvasive wrist-based blood pressure monitor that incorporates training techniques for posture and subject variability," *IEEE journal of biomedical and health informatics*, vol. 20, no. 5, pp. 1291–1300, 2016.

[18] P. Nabeel, S. Karthik, J. Joseph, and M. Sivaprakasam, "Arterial blood pressure estimation from local pulse wave velocity using dual-element photoplethysmograph probe," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1399–1408, 2018.

[19] Y. Wang, Z. Liu, and S. Ma, "Cuff-less blood pressure measurement from dual-channel photoplethysmographic signals via peripheral pulse transit time with singular spectrum analysis," *Physiological measurement*, vol. 39, no. 2, p. 025010, 2018.

[20] O. H.-Y. Shay and L. L. Dai, "System and method for biometric measurements," May 4 2017. US Patent App. 15/337,127.

[21] B. Ibrahim, J. McMurray, and R. Jafari, "A wrist-worn strap with an array of electrodes for robust physiological sensing," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4313–4317, IEEE, 2018.

[22] N. Luo, W. Dai, C. Li, Z. Zhou, L. Lu, C. C. Poon, S. Chen, Y. Zhang, and N. Zhao, "Flexible piezoresistive sensor patch enabling ultralow power cuffless blood pressure measurement," *Advanced Functional Materials*, vol. 26, no. 8, pp. 1178–1187, 2016.

[23] B. Ibrahim and R. Jafari, "Cuffless blood pressure monitoring from an array of wrist bioimpedance sensors using subject-specific regression models: Proof of concept," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 6, pp. 1723–1735, 2019.

[24] S. Yan, H. Song, N. Li, L. Zou, and L. Ren, "Improve unsupervised domain adaptation with mixup training," *arXiv preprint arXiv:2001.00677*, 2020.

[25] A. Moin, A. Zhou, A. Rahimi, A. Menon, S. Benatti, G. Alexandrov, S. Tamakloe, J. Ting, N. Yamamoto, Y. Khan, *et al.*, "A wearable biosensing system with in-sensor adaptive ma-

chine learning for hand gesture recognition," *Nature Electronics*, vol. 4, no. 1, pp. 54–63, 2021.

[26] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[27] O. of the National Coordinator for Health Information Technology, "Office-based physician electronic health record adoption," 2019.

[28] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Medical concept representation learning from electronic health records and its application on heart failure prediction," *arXiv preprint arXiv:1602.03686*, 2016.

[29] B. K. Beaulieu-Jones, C. S. Greene, *et al.*, "Semi-supervised learning of the electronic health record for phenotype stratification," *Journal of biomedical informatics*, vol. 64, pp. 168–178, 2016.

[30] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with lstm recurrent neural networks," *arXiv preprint arXiv:1511.03677*, 2015.

[31] Z. Che, D. Kale, W. Li, M. T. Bahadori, and Y. Liu, "Deep computational phenotyping," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 507–516, 2015.

[32] K. E. Henry, D. N. Hager, P. J. Pronovost, and S. Saria, "A targeted real-time early warning score (trewscore) for septic shock," *Science translational medicine*, vol. 7, no. 299, pp. 299ra122–299ra122, 2015.

[33] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. V. Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *arXiv preprint arXiv:1703.07771*, 2017.

[34] Y. Xu, S. Biswal, S. R. Deshpande, K. O. Maher, and J. Sun, "Raim: Recurrent attentive and intensive model of multimodal patient monitoring data," in *Proceedings of the 24th ACM*

SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2565–2573, 2018.

[35] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[37] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmark of deep learning models on large healthcare mimic datasets," *arXiv preprint arXiv:1710.08531*, 2017.

[38] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, "E2gan: End-to-end generative adversarial network for multivariate time series imputation," in *Proceedings of the 28th international joint conference on artificial intelligence*, pp. 3094–3100, AAAI Press, 2019.

[39] S. N. Shukla and B. M. Marlin, "Multi-time attention networks for irregularly sampled time series," *arXiv preprint arXiv:2101.10318*, 2021.

[40] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, pp. 1126–1135, PMLR, 2017.

[41] X. S. Zhang, F. Tang, H. H. Dodge, J. Zhou, and F. Wang, "Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2487–2495, 2019.

[42] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[43] P. Gupta, P. Malhotra, J. Narwariya, L. Vig, and G. Shroff, "Transfer learning for clinical time series analysis using deep neural networks," *Journal of Healthcare Informatics Research*, vol. 4, no. 2, pp. 112–137, 2020.

[44] T. Desautels, J. Calvert, J. Hoffman, Q. Mao, M. Jay, G. Fletcher, C. Barton, U. Chettipally, Y. Kerem, and R. Das, "Using transfer learning for improved mortality prediction in a data-scarce hospital setting," *Biomedical informatics insights*, vol. 9, p. 1178222617712994, 2017.

[45] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," 2005.

[46] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5982–5991, 2019.

[47] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

[48] Y. He and D. Zhou, "Self-training from labeled features for sentiment analysis," *Information Processing & Management*, vol. 47, no. 4, pp. 606–616, 2011.

[49] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, p. 896, 2013.

[50] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," *Advances in neural information processing systems*, vol. 27, pp. 3365–3373, 2014.

[51] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko, "Semi-supervised learning with ladder networks," *arXiv preprint arXiv:1507.02672*, 2015.

[52] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *Advances in neural information processing systems*, vol. 29, pp. 1163–1171, 2016.

[53] J. Pereira and M. Silveira, "Learning representations from healthcare time series data for unsupervised anomaly detection," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 1–7, IEEE, 2019.

[54] H. Yu and A. Sano, "Semi-supervised learning and data augmentation in wearable-based momentary stress detection in the wild," *arXiv preprint arXiv:2202.12935*, 2022.

[55] J.-R. Jiang, J.-B. Kao, and Y.-L. Li, "Semi-supervised time series anomaly detection based on statistics and deep learning," *Applied Sciences*, vol. 11, no. 15, p. 6698, 2021.

[56] H. Gweon and H. Yu, "A nearest neighbor-based active learning method and its application to time series classification," *Pattern Recognition Letters*, vol. 146, pp. 230–236, 2021.

[57] Y. Wang, J. Guo, S. Song, and G. Huang, "Meta-semi: A meta-learning approach for semi-supervised learning," *arXiv preprint arXiv:2007.02394*, 2020.

[58] T. Xiao, X.-Y. Zhang, H. Jia, M.-M. Cheng, and M.-H. Yang, "Semi-supervised learning with meta-gradient," in *International Conference on Artificial Intelligence and Statistics*, pp. 73–81, PMLR, 2021.

[59] C. A. Webb, Z. D. Cohen, C. Beard, M. Forgeard, A. D. Peckham, and T. Björgvinsson, "Personalized prognostic prediction of treatment outcome for depressed patients in a naturalistic psychiatric hospital setting: A comparison of machine learning approaches.," *Journal of Consulting and Clinical Psychology*, vol. 88, no. 1, p. 25, 2020.

[60] F.-Y. Cheng, H. Joshi, P. Tandon, R. Freeman, D. L. Reich, M. Mazumdar, R. Kohli-Seth, M. A. Levin, P. Timsina, and A. Kia, "Using machine learning to predict icu transfer in hospitalized covid-19 patients," *Journal of clinical medicine*, vol. 9, no. 6, p. 1668, 2020.

[61] Y. Raita, T. Goto, M. K. Faridi, D. F. Brown, C. A. Camargo, and K. Hasegawa, "Emergency department triage prediction of clinical outcomes using machine learning models," *Critical care*, vol. 23, no. 1, pp. 1–13, 2019.

[62] S. Sajjadi, A. Das, R. Gutierrez-Osuna, T. Chaspari, P. Paromita, L. E. Ruebush, N. E. Deutz, and B. J. Mortazavi, "Towards the development of subject-independent inverse metabolic models," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3970–3974, IEEE, 2021.

[63] Z. Huo, B. J. Mortazavi, T. Chaspari, N. Deutz, L. Ruebush, and R. Gutierrez-Osuna, "Predicting the meal macronutrient composition from continuous glucose monitors," in *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pp. 1–4, IEEE, 2019.

[64] K. J. Bell, C. E. Smart, G. M. Steil, J. C. Brand-Miller, B. King, and H. A. Wolpert, "Impact of fat, protein, and glycemic index on postprandial glucose control in type 1 diabetes: implications for intensive diabetes management in the continuous glucose monitoring era," *Diabetes care*, vol. 38, no. 6, pp. 1008–1015, 2015.

[65] Q. Suo, F. Ma, Y. Yuan, M. Huai, W. Zhong, A. Zhang, and J. Gao, "Personalized disease prediction using a cnn-based similarity learning method," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 811–816, IEEE, 2017.

[66] K. Liu, X. Zhang, W. Chen, S. Alan, J. A. Kellum, M. E. Matheny, S. Q. Simpson, Y. Hu, and M. Liu, "Development and validation of a personalized model with transfer learning for acute kidney injury risk estimation using electronic health records," *JAMA Network Open*, vol. 5, no. 7, pp. e2219776–e2219776, 2022.

[67] E. K. Oikonomou, E. S. Spatz, M. A. Suchard, and R. Khera, "Individualising intensive systolic blood pressure reduction in hypertension using computational trial phenomaps and machine learning: a post-hoc analysis of randomised clinical trials," *The Lancet Digital Health*, vol. 4, no. 11, pp. e796–e805, 2022.

[68] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *arXiv preprint arXiv:1703.05175*, 2017.

[69] F. Boniolo, G. Boniolo, and G. Valente, "Prediction via similarity: Biomedical big data and the case of cancer models," *Philosophy & Technology*, vol. 36, no. 1, p. 8, 2023.

[70] J. Crabbé, Z. Qian, F. Imrie, and M. van der Schaar, "Explaining latent representations with a corpus of examples," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12154–12166, 2021.

[71] Z. Fu, X. He, E. Wang, J. Huo, J. Huang, and D. Wu, "Personalized human activity recognition based on integrated wearable sensor and transfer learning," *Sensors*, vol. 21, no. 3, p. 885, 2021.

[72] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.

[73] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 120–128, 2006.

[74] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and information systems*, vol. 36, no. 3, pp. 537–556, 2013.

[75] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*, pp. 2208–2217, PMLR, 2017.

[76] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[77] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[78] M. H. Olsen, S. Y. Angell, S. Asma, P. Boutouyrie, D. Burger, J. A. Chirinos, A. Damasceno, C. Delles, A.-P. Gimenez-Roqueplo, and D. Hering, "A call to action and a lifecourse strategy to address the global burden of raised blood pressure on current and future generations: the lancet commission on hypertension," *The Lancet*, vol. 388, no. 10060, pp. 2665–2712, 2016.

[79] E. O'Brien, R. Asmar, L. Beilin, Y. Imai, G. Mancia, T. Mengden, M. Myers, P. Padfield, P. Palatini, and G. Parati, "Practice guidelines of the european society of hypertension for clinic, ambulatory and self blood pressure measurement," *Journal of hypertension*, vol. 23, no. 4, pp. 697–701, 2005.

[80] T. G. Pickering, G. D. James, C. Boddie, G. A. Harshfield, S. Blank, and J. H. Laragh, "How common is white coat hypertension?," *Jama*, vol. 259, no. 2, pp. 225–228, 1988.

[81] T. G. Pickering, G. A. Harshfield, H. D. Kleinert, S. Blank, and J. H. Laragh, "Blood pressure during normal daily activities, sleep, and exercise: comparison of values in normal and hypertensive subjects," *Jama*, vol. 247, no. 7, pp. 992–996, 1982.

[82] J. R. Banegas, L. M. Ruilope, A. de la Sierra, E. Vinyoles, M. Gorostidi, J. J. de la Cruz, G. Ruiz-Hurtado, J. Segura, F. Rodríguez-Artalejo, and B. Williams, "Relationship between clinic and ambulatory blood-pressure measurements and mortality," *New England Journal of Medicine*, vol. 378, no. 16, pp. 1509–1520, 2018.

[83] G. Stergiou, P. Palatini, R. Asmar, A. de la Sierra, M. Myers, A. Shennan, J. Wang, E. O'brien, and G. Parati, "Blood pressure measurement and hypertension diagnosis in the 2017 us guidelines: first things first," *Hypertension*, vol. 71, no. 6, pp. 963–965, 2018.

[84] P. K. Whelton, R. M. Carey, W. S. Aronow, D. E. Casey, K. J. Collins, C. D. Himmelfarb, S. M. DePalma, S. Gidding, K. A. Jamerson, and D. W. Jones, "2017 acc/aha/aapa/abc/acpm/ags/apha/ash/aspc/nma/pcna guideline for the prevention, detection, evaluation, and management of high blood pressure in adults: a report of the american col-

lege of cardiology/american heart association task force on clinical practice guidelines," *Journal of the American College of Cardiology*, vol. 71, no. 19, pp. e127–e248, 2018.

[85] H.-Q. Fan, Y. Li, L. Thijs, T. W. Hansen, J. Boggia, M. Kikuya, K. Björklund-Bodegård, T. Richart, T. Ohkubo, and J. Jeppesen, "Prognostic value of isolated nocturnal hypertension on ambulatory measurement in 8711 individuals from 10 populations," *Journal of hypertension*, vol. 28, no. 10, pp. 2036–2045, 2010.

[86] G. S. Stergiou, B. Alpert, S. Mieke, R. Asmar, N. Atkins, S. Eckert, G. Frick, B. Friedman, T. Graßl, and T. Ichikawa, "A universal standard for the validation of blood pressure measuring devices: Association for the advancement of medical instrumentation/european society of hypertension/international organization for standardization (aami/esh/iso) collaboration statement," *Hypertension*, vol. 71, no. 3, pp. 368–374, 2018.

[87] L. L. Raket, J. Jaskolowski, B. J. Kinon, J. C. Brasen, L. Jönsson, A. Wehnert, and P. Fusar-Poli, "Dynamic electronic health record detection (detect) of individuals at risk of a first episode of psychosis: a case-control development and validation study," *The Lancet Digital Health*, vol. 2, no. 5, pp. e229–e239, 2020.

[88] C. G. Walsh, K. B. Johnson, M. Ripperger, S. Sperry, J. Harris, N. Clark, E. Fielstein, L. Novak, K. Robinson, and W. W. Stead, "Prospective validation of an electronic health record–based, real-time suicide risk model," *JAMA network open*, vol. 4, no. 3, pp. e211428–e211428, 2021.

[89] M. Mori, T. J. Durant, C. Huang, B. J. Mortazavi, A. Coppi, R. A. Jean, A. Geirsson, W. L. Schulz, and H. M. Krumholz, "Toward dynamic risk prediction of outcomes after coronary artery bypass graft: Improving risk prediction with intraoperative events using gradient boosting," *Circulation: Cardiovascular Quality and Outcomes*, pp. CIRCOUTCOMES–120, 2021.

[90] Z. C. Lipton, D. C. Kale, R. Wetzel, *et al.*, "Modeling missing data in clinical time series with rnns," *Machine Learning for Healthcare*, vol. 56, 2016.

[91] S. Hu, J. Tomczak, and M. Welling, "Meta-learning for medical image classification," 2018.

[92] N. Banluesombatkul, P. Ouppaphan, P. Leelaarporn, P. Lakhan, B. Chaitusaney, N. Jaimchariya, E. Chuangsuwanich, W. Chen, H. Phan, N. Dilokthanakul, *et al.*, "Metasleeplearner: A pilot study on fast adaptation of bio-signals-based sleep stage classifier to new individual subject using meta-learning," *IEEE Journal of Biomedical and Health Informatics*, 2020.

[93] T. Naren, Y. Zhu, and M. D. Wang, "Covid-19 diagnosis using model agnostic meta-learning on limited chest x-ray images," in *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 1–9, 2021.

[94] G. Wilson, J. R. Doppa, and D. J. Cook, "Multi-source deep domain adaptation with weak supervision for time-series sensor data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1768–1778, 2020.

[95] M. Javeed, A. Jalal, and K. Kim, "Wearable sensors based exertion recognition using statistical features and random forest for physical healthcare monitoring," in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, pp. 512–517, IEEE, 2021.

[96] J. Zhang, Y. Wang, C. Wang, and M. Zhou, "Symmetrical hierarchical stochastic searching on the line in informative and deceptive environments," *IEEE transactions on cybernetics*, vol. 47, no. 3, pp. 626–635, 2016.

[97] J. Zhang, Y. Wang, and M. Zhou, "Fast adaptive search on the line in dual environments," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pp. 1540–1545, IEEE, 2017.

[98] P. Macadam, J. B. Cronin, A. M. Uthoff, and E. H. Feser, "Effects of different wearable resistance placements on sprint-running performance: A review and practical applications," *Strength & Conditioning Journal*, vol. 41, no. 3, pp. 79–96, 2019.

[99] T. Plötz and Y. Guan, "Deep learning for human activity recognition in mobile computing," *Computer*, vol. 51, no. 5, pp. 50–59, 2018.

[100] L. Zhang, X. Chen, T. Chen, Z. Wang, and B. J. Mortazavi, "Dynehr: Dynamic adaptation of models with data heterogeneity in electronic health records," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–4, IEEE, 2021.

[101] A.-A. Kafy, R. M. Shuvo, M. N. H. Naim, M. S. Sikdar, R. R. Chowdhury, M. A. Islam, M. H. S. Sarker, M. H. H. Khan, M. A. Kona, *et al.*, "Remote sensing approach to simulate the land use/land cover and seasonal land surface temperature change using machine learning algorithms in a fastest-growing megacity of bangladesh," *Remote Sensing Applications: Society and Environment*, vol. 21, p. 100463, 2021.

[102] E. Brynjolfsson, T. Mitchell, and D. Rock, "What can machines learn, and what does it mean for occupations and the economy?," in *AEA Papers and Proceedings*, vol. 108, pp. 43–47, 2018.

[103] L. Zhang, A. Ebrahimi, and D. Klabjan, "Layer flexible adaptive computation time," in *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, IEEE, 2021.

[104] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," *arXiv preprint arXiv:1803.00676*, 2018.

[105] S. Ding, Z. Chen, T. Zheng, and J. Luo, "Rf-net: A unified meta-learning framework for rf-enabled one-shot human activity recognition," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pp. 517–530, 2020.

[106] T. Bansal, R. Jha, T. Munkhdalai, and A. McCallum, "Self-supervised meta-learning for few-shot natural language classification tasks," *arXiv preprint arXiv:2009.08445*, 2020.

[107] X. Wang, L. Zhang, and D. Klabjan, "Keyword-based topic modeling and keyword selection," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 1148–1154, IEEE, 2021.

[108] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," *arXiv preprint arXiv:1801.08930*, 2018.

[109] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine, "Meta-learning with implicit gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[110] Z. Nowroozilarki, A. Pakbin, J. Royalty, D. K. Lee, and B. J. Mortazavi, "Real-time mortality prediction using mimic-iv icu data via boosted nonparametric hazards," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–4, IEEE, 2021.

[111] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.

[112] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[113] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," *arXiv preprint arXiv:1706.05208*, 2017.

[114] X. Song, W. Gao, Y. Yang, K. Choromanski, A. Pacchiano, and Y. Tang, "Es-maml: Simple hessian-free meta learning," *arXiv preprint arXiv:1910.01215*, 2019.

[115] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[116] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *Plos one*, vol. 16, no. 7, p. e0254841, 2021.

[117] I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark, "Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012," in *2012 Computing in Cardiology*, pp. 245–248, IEEE, 2012.

[118] T. Ahmad, M. J. Pencina, P. J. Schulte, E. O'Brien, D. J. Whellan, I. L. Piña, D. W. Kitzman, K. L. Lee, C. M. O'Connor, and G. M. Felker, "Clinical implications of chronic heart failure phenotypes defined by cluster analysis," *Journal of the American College of Cardiology*, vol. 64, no. 17, pp. 1765–1774, 2014.

[119] D. Neil, M. Pfeiffer, and S.-C. Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences," *Advances in neural information processing systems*, vol. 29, 2016.

[120] S. L. Murphy, K. D. Kochanek, J. Xu, and E. Arias, "Mortality in the united states, 2020," 2021.

[121] W. Sami, T. Ansari, N. S. Butt, and M. R. Ab Hamid, "Effect of diet on type 2 diabetes mellitus: A review," *International journal of health sciences*, vol. 11, no. 2, p. 65, 2017.

[122] Y. Zhou, H. Peng, H. Xu, J. Li, M. Golovko, H. Cheng, E. C. Lynch, L. Liu, N. McCauley, L. Kennedy, *et al.*, "Maternal diet intervention before pregnancy primes offspring lipid metabolism in liver," *Laboratory Investigation*, vol. 100, no. 4, pp. 553–569, 2020.

[123] H. Peng, H. Xu, J. Wu, J. Li, Y. Zhou, Z. Ding, S. K. Siwko, X. Yuan, K. L. Schalinske, G. Alpini, *et al.*, "Maternal high-fat diet disrupted one-carbon metabolism in offspring, contributing to nonalcoholic fatty liver disease," *Liver International*, vol. 41, no. 6, pp. 1305–1319, 2021.

[124] H. Peng, J. Li, H. Xu, X. Wang, L. He, N. McCauley, K. K. Zhang, and L. Xie, "Offspring nafld liver phospholipid profiles are differentially programmed by maternal high-fat diet and maternal one carbon supplement," *The Journal of Nutritional Biochemistry*, vol. 111, p. 109187, 2023.

[125] M. Farooq and E. Sazonov, "A novel wearable device for food intake and physical activity recognition," *Sensors*, vol. 16, no. 7, p. 1067, 2016.

[126] K. Ishihara, N. Uchiyama, S. Kizaki, E. Mori, T. Nonaka, and H. Oneda, "Application of continuous glucose monitoring for assessment of individual carbohydrate requirement during ultramarathon race," *Nutrients*, vol. 12, no. 4, p. 1121, 2020.

[127] S. Mezgec, T. Eftimov, T. Bucher, and B. K. Seljak, "Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment," *Public health nutrition*, vol. 22, no. 7, pp. 1193–1202, 2019.

[128] K. Motoki, T. Saito, S. Suzuki, and M. Sugiura, "Evaluation of energy density and macronutrients after extremely brief time exposure," *Appetite*, vol. 162, p. 105143, 2021.

[129] Z.-Y. Ming, J. Chen, Y. Cao, C. Forde, C.-W. Ngo, and T. S. Chua, "Food photo recognition for dietary tracking: System and experiment," in *MultiMedia Modeling: 24th International Conference, MMM 2018, Bangkok, Thailand, February 5-7, 2018, Proceedings, Part II 24*, pp. 129–141, Springer, 2018.

[130] A. Chakrabarty, F. J. Doyle, and E. Dassau, "Deep learning assisted macronutrient estimation for feedforward-feedback control in artificial pancreas systems," in *2018 Annual American Control Conference (ACC)*, pp. 3564–3570, IEEE, 2018.

[131] M. B. Gillingham, Z. Li, R. W. Beck, P. Calhoun, J. R. Castle, M. Clements, E. Dassau, F. J. Doyle, R. L. Gal, P. Jacobs, *et al.*, "Assessing mealtime macronutrient content: patient perceptions versus expert analyses via a novel phone app," *Diabetes technology & therapeutics*, vol. 23, no. 2, pp. 85–94, 2021.

[132] Y.-J. Ko, A. Putkonen, A. S. Aydin, S. Feiz, Y. Wang, V. Ashok, I. Ramakrishnan, A. Oulasvirta, and X. Bi, "Modeling gliding-based target selection for blind touchscreen users," in *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*, pp. 1–14, 2021.

[133] Y. Lu, T. Stathopoulou, M. F. Vasiloglou, L. F. Pinault, C. Kiley, E. K. Spanakis, and S. Mougiakakou, "gofoodtm: an artificial intelligence system for dietary assessment," *Sensors*, vol. 20, no. 15, p. 4283, 2020.

[134] X. Chen, Y. Zhu, H. Zhou, L. Diao, and D. Wang, "Chinesefoodnet: A large-scale image dataset for chinese food recognition," *arXiv preprint arXiv:1705.02743*, 2017.

[135] N. Hnoohom and S. Yuenyong, "Thai fast food image classification using deep learning," in *2018 International ECTI northern section conference on electrical, electronics, computer and telecommunications engineering (ECTI-NCON)*, pp. 116–119, IEEE, 2018.

[136] S. Giovany, A. Putra, A. S. Hariawan, L. A. Wulandhari, and E. Irwansyah, "Indonesian food image recognition using convolutional neural network," in *Artificial Intelligence Methods in Intelligent Algorithms: Proceedings of 8th Computer Science On-line Conference 2019, Vol. 2 8*, pp. 208–217, Springer, 2019.

[137] J. Martinez, Z. Nowroozilarki, R. Jafari, and B. J. Mortazavi, "Data-driven guided attention for analysis of physiological waveforms with deep learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 11, pp. 5482–5493, 2022.

[138] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[139] S. N. Shukla and B. M. Marlin, "Integrating physiological time series and clinical notes with deep learning for improved icu mortality prediction," *arXiv preprint arXiv:2003.11059*, 2020.

[140] A. Sagheer and M. Kotb, "Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems," *Scientific reports*, vol. 9, no. 1, pp. 1–16, 2019.

[141] P. Funk and N. Xiong, "Case-based reasoning and knowledge discovery in medical applications with time series," *Computational Intelligence*, vol. 22, no. 3-4, pp. 238–253, 2006.

[142] P. Funk and N. Xiong, "Extracting knowledge from sensor signals for case-based reasoning with longitudinal time series data," *Case-Based Reasoning on images and signals*, pp. 247–284, 2008.

[143] Y. Wang, G. Borca-Tasciuc, N. Goel, P. Fodor, and M. Kifer, "Knowledge authoring with factual english," *arXiv preprint arXiv:2208.03094*, 2022.

[144] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[145] Z. Huo, L. Zhang, R. Khera, S. Huang, X. Qian, Z. Wang, and B. J. Mortazavi, "Sparse gated mixture-of-experts to separate and interpret patient heterogeneity in ehr data," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–4, IEEE, 2021.

[146] L. Phased, "Accelerating recurrent network training for long or event-based sequences," 2016.