Hierarchical Multitask Learning

by

Salim Malakouti

Bachelor of Science, Amirkabir University of Technology, 2013

Submitted to the Graduate Faculty of

the School of Computing and Information in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

DEPARTMENT OF COMPUTER SCIENCE

This dissertation was presented

by

Salim Malakouti

It was defended on

September 27, 2022

and approved by

Milos Hauskrecht,PhD, Professor, University of Pittsburgh

Greg Cooper, PhD, Professor, University of Pittsburgh

Adriana Kovashka, PhD, Associate Professor, University of Pittsburgh

Erin Walker, PhD, Associate Professor, University of Pittsburgh

Hierarchical Multitask Learning

Salim Malakouti, PhD

University of Pittsburgh, 2022

Traditionally, machine learning research has adopted methods that were designed to learn one or a set of machine learning tasks independently. However, motivated by our brain's learning mechanism to transfer knowledge from past and other related experiences, recent research has developed and studied methods incorporating target task relationships in the learning algorithms. The area of machine learning in which multiple target tasks are solved simultaneously while exploiting their similarities and underlying structures is known as multi-task learning. Multi-task learning methods (MTL) have proven effective in learning improved machine learning models by facilitating the transfer of knowledge through simultaneously learning a set of target tasks.

However, the success of existing multi-task learning methods depends on the extent of the similarity between the target tasks. When tasks are not sufficiently similar, the negative transfer that impacts the quality of the learned models may occur. Therefore, new techniques were adopted that took advantage of task clusters, task-task relatedness, or an asymmetric knowledge transfer. However, none of these techniques are adequate when applied to a large number of heterogeneous tasks organized in a complex hierarchical structure. The abundance of such hierarchies in many domains, including health-care, document classification, and image classification, motivates the development of a new class of multi-task learning methods that can take advantage of these complex hierarchical task relationships.

In this thesis, we explore and develop supervised multi-task learning methods that leverage existing task hierarchies to guide the transfer of knowledge between related tasks and evaluate these methods in the context of healthcare applications. First, we propose a simple, yet flexible, approach for learning low-dimensional representations of patients' electronic health records data that are able to overcome challenges related to learning of the models for multiple target tasks from such data. Second, we propose new hierarchical multi-task learning methods that enable the transfer of knowledge in the form of parameter transfer. Third,

we study and present new feature-based hierarchical multi-task learning methods that utilize feature transfer instead of parameter transfer solutions to further improve the performance of the models. Finally, we discuss the open questions and problems, and provide ideas for future research directions.

Table of Contents

List of Tables

List of Figures

Preface

The work presented in this thesis would not have been possible without many people's help, support, and guidance. I want to take this opportunity to acknowledge the people who supported me throughout this journey and my life in Pittsburgh.

First and foremost, I would like to extend my deepest gratitude to my Ph.D. advisor Prof. Milos Hauskrecht who guided me throughout this process and without whose support the work presented in this dissertation would not have been possible. I was given a unique opportunity to work with Milos throughout the past nine years on my thesis and some of the most exciting applications of artificial intelligence and machine learning in Healthcare, taking on challenges at the forefront of bringing AI to the bedside. This challenging but exciting journey granted me a unique opportunity that I hope will help set forth the foundation for a new journey I plan to embrace in the next few decades of my life, building a friendship and relationship with Milos beyond the scope of my work at the University of Pittsburgh.

I would also like to thank my dissertation committee members: Greg Copper, Adriana Kovashka, and Erin Walker, who guided, inspired, and patiently supported my work and dissertation. Their passionate guidance, kind support, and feedback guided me throughout this journey and challenged me to present my best work these past couple of years.

I also believe I owe a lot of my progress and experiences to those I had the opportunity to work with throughout these years, including Dr. Gilles Clermont, who was both a mentor and a friend. Gilles's guidance, advice, and support throughout my long but exciting journey was not only a fantastic resource but also often an energizing source of motivation. Prof. Greg Cooper, who beyond my dissertation, challenged me throughout research projects to bring forth my best work and encouraged me to go beyond expectations and pay attention to details. Furthermore, I would like to thank various people in our research group whom I had the opportunity to work with, discuss ideas and have friendly but exciting debates that helped achieve the best version of our work both in scientific and engineering problems, including Joo Heung Yoon, Zhipeng Luo, Mathew Barren, Siqi Liu, Charmgil Hong, Ankitkumar Joshi, Junhen Wang, Jeongmin Lee and Dr. Michael Pinsky, Chris Horvat, Andrew King, Edvin

Music, Dan Ricketts and others in the computer science department who were at times mentors and friends including Prof. Daniel Mosse, Keena M Walker, Karen Joll Dicks, Bob Hoffman, Matheus de Lima Barbosa, Terry Wood, the late Russ Howard, Adam Hobaugh, Walter Gibson and Brandi L Belleau.

It would also not be just if I did not thank my dear parents and family, Kazem Malakouti, Camelia Vahidi, and Sina Malakouti, who always supported me and guided me no matter where I was. And of course, I would like to thank my dearest friends who truly made Pittsburgh a new home for me and became my new family, including the Sixburgh©team, Navid Kazem, Reza Azimi, Milad Memarzadeh, Mohammad Ahmadpour, and Yashar Aucie. My dear friends Fattaneh Jabbari, Amin Tajgardoon, Colleen Marsh, Ali Behrang, Meena Mehdizadeh, Sareh Yousefzadeh, Mostafa Mirshekari, Drew Charles, Rachel Shockey, Coleman Drake, Pearl Nielson, Ida Chavoshan and Mike Sotace.

Last but not least, I would like to extend my deepest gratitude to my dear partner in life, Sadaf Tayefeh without whose support, this would not have been possible. If anything, this dissertation is our achievement. She has always been my best friend and warmest supporter, and a role model I looked up to as an example of kindness, hard work, and persistence. I have been lucky to have her in my life from the day we met till the day we said our vows, and today which I hope to begin a new chapter of our lives together.

1.0 Introduction

1.1 Motivation

Since the first time computerized solutions emerged in 1880 to process the census data, advancements and prevalence of technology have revolutionized the generation, storage, and extraction of data in ways that might have been unimaginable in the past. This has motivated research and development of machine learning algorithms that can automatically learn to perform classification, detection, and prediction tasks otherwise performed by humans.

1.1.1 Multi-task Learning: Promise and Shortcomings

Traditionally, machine learning research has adopted methods designed to learn one or a set of machine learning tasks independently. However, motivated by our brain's learning mechanisms to transfer knowledge from past experiences, recent work has developed methods that model target task relationships. The area of machine learning that incorporates underlying task structures and relationships to facilitate knowledge transfer between target tasks is called multi-task learning (MTL). In multi-task learning, multiple target tasks are solved simultaneously while exploiting their similarities. The goal of MTL methods is to train improved machine learning models for T target tasks from their input data $X_1, ..., X_T$ by facilitating the simultaneous transfer of knowledge between related tasks. Two tasks are considered related if they have similar behavior concerning some or all regions of input data.

One challenge in developing and adopting multi-task learning methods is ensuring that knowledge transfer only happens when it can result in improved model performance. Otherwise, we say a negative transfer has occurred [160, 175]. Negative transfer refers to a transfer of knowledge that results in models with reduced performance. A number of solutions have been designed to tackle this problem by either clustering tasks into subgroups based on task similarities, allowing the transfer only between tasks within the same cluster, or by defining and learning task-task relatedness weights.

However, neither of these methods has proven sufficient when solving problems that involve learning machine learning models for a large number of target tasks that are a part of a more complex hierarchical task structure [134]. First, by solely utilizing pairwise similarity measures to model task-task relatedness, we lose vital information about how tasks can be categorized into groups of closely relevant target tasks. Second, modeling of task groups as a set of flat categories results in a loss of details about different levels of similarities between tasks. Finally, the existing methods fail to consider the relationship between categories.

1.1.2 Inspirations from Hierarchical Learning Mechanisms in Human Brain

The human brain, which evolved over millions of years, has developed sophisticated and powerful means of processing information and making inferences, many of which remain unknown. However, one area in which researchers from neurology to psychology have come to similar understandings is the role abstractions and hierarchies play in our daily learning experiences. For example, imagine a child first introduced to the fruit Tangerine, perhaps after she had already experienced eating an Orange. She might even initially mistake the Tangerine for an Orange because of similarities such as their round shape, the texture of their zest, or even the fact that they are both pulpy. However, when told about her mistake, she automatically learns to identify similarities and differences between Tangerine and Orange. To learn new concepts, our brain somehow transfers the knowledge it accumulated about past similar concepts to the newly seen object. In fact, it uses our past experiences and knowledge to improve the process needed to learn what a Tangerine looks like. Imagine it would instead require being exposed to a new concept or experience often and in isolation to learn new things. How would that change our pace of learning complex information? Transferring information and knowledge to new experiences and concepts enables our brain to learn new notions without requiring many repetitions of the same experience. As a result, we can learn to identify a Tangerine much faster than we learned to recognize its cousin Orange.

In the field of psychology, this process is called transfer of learning, and it has been the subject of studies and debates for many years [196]. In fact, we now know that our brains

tend to transfer the knowledge it has already accumulated from far and near concepts in different ways to enhance the process of learning a new experience, object, or task [164].

However, learning is an asynchronous experience for human beings, simultaneously facing new concepts and tasks every day. Imagine the same child facing another similar fruit, such as a Grapefruit. Inevitably she'd find herself grouping the newly faced object into the same category as Orange and Tangerine. This is done based on their commonalities and how they differ from other much more variant types of fruits such as melons and apples. Eventually, as she learns about other similar fruits such as Lime, Lemon, and Mandarin, she will likely find herself not only adding them to the newly formed hypothetical group of citrus fruits but also automatically creating new sub-categories. Perhaps one such category is dedicated to various types of oranges, including Blood Orange, Juice Orange, and even Cava Cava. Her brain detects shared properties among the members of a group forming an abstract notion representing the entirety of its members. As a result, a future object she might encounter will automatically be placed into the group with the most similar abstract concept, and existing knowledge will be transferred from each group to the extent of the similarity of the new concept with such abstract notions. Our brain also identifies other features that allow it to differentiate members of a particular group from one another. For instance, this might be the dark red color of Blood Orange. However, differences within one group might sometimes be more subtle than those of other groups. For example, many might fail when trying to identify a Tangerine from its almost identical sibling Mandarin. In fact, they are so similar that in some languages, such as Farsi, they have the same exact name.

Although many aspects of our brain's functions are left to be discovered, there seems to be no dispute that hierarchies have an essential role in enhancing the performance of our brain's learning and thinking process [174, 173, 140, 183]. Thanks to this hierarchical way of creating an abstraction of the world, we often find ourselves needing only a few or sometimes precisely one example to learn a new concept, an experience, or a task. Thus, categorization and hierarchies are an essential underlying system of human brain function to render comprehensibly this otherwise bewildering diversity of concepts and experiences [19, 189]. This role has such a strength in human learning and thinking process that in some games such as "Twenty Questions" [207] we put that ability to test in which contestants

hope to guess a secret object, name, or concept from all that is known to us. Many of us often find ourselves opting in for a common winning strategy that relies on narrowing down the search space by repeatedly asking questions about the properties of the answer.

Despite these hierarchical relationships' impact on our brain's learning process, we don't always find ourselves starting from the top of the fruit hierarchy to identify an Orange. We would instead do that immediately. Moreover, it seems like our brain learns to take advantage of such hierarchical structures and abstractions aside from when learning new concepts and experiences, mostly when independent models of concepts fail to provide confident answers.

1.1.3 Hierarchical Multi-task Learning

These fascinating mechanisms of our brain motivate studying machine learning methods that can take advantage of complex hierarchical structures of target tasks the way the human brain does. Therefore, a natural and interesting question is how can we take advantage of such hierarchical structures to improve machine learning models? However, we first need to understand how target machine learning tasks could be organized within a hierarchy to answer this question..

In a hierarchy, target tasks can be restricted to the leaves, or they can cover both leaf nodes and higher-level categories. In the former case, the internal category nodes only help to define the task similarities and relationships by creating a hierarchical grouping of tasks. Figure 1 shows a part of a hierarchy for the scientific classification of animals in which target tasks are only defined by the leaves of the hierarchy.

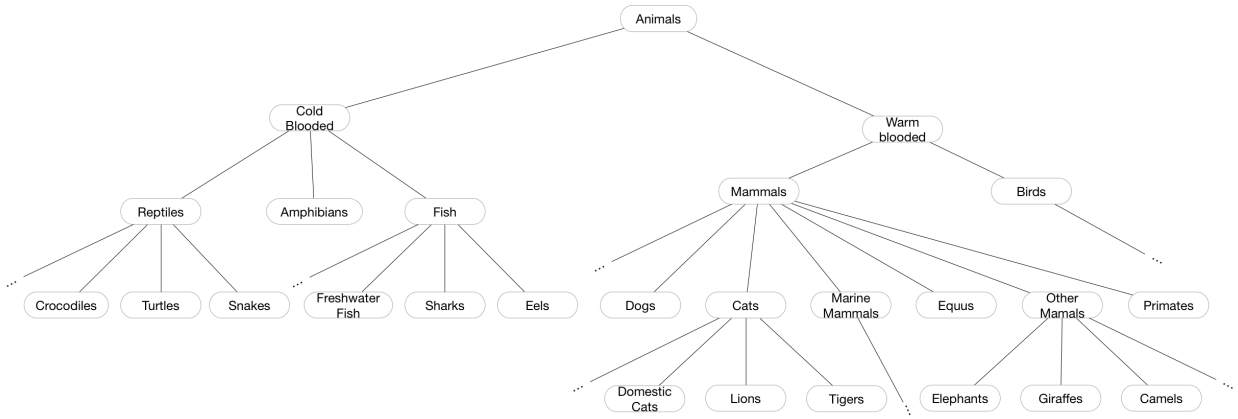


Figure 1: Part of a hierarchy for scientific classification of animals

Intuitively, in the hierarchical categorization of concepts, members of a group have certain commonalities that are not shared with members of other groups at the same or higher levels of the hierarchy. Furthermore, these similarities grow stronger as we move toward categories at the lower levels of the hierarchy. For instance, in Figure 1 various types of cats are more closely related than all members of the general family of mammals. Hierarchies enable us to find and utilize relations at different levels of the tree to improve the learning of machine learning models. For example, if we know how to classify mammals accurately, can we adopt this model to facilitate learning a more accurate model for identifying cats? Additionally, can we use the model for domestic cats to improve the classification of lions or tigers? Or can we learn a better model for elephants by using the model for classifying giraffes?

In this work, we explore and propose new multi-task learning methods that rely on and take advantage of hierarchical structures of target tasks to guide the transfer of knowledge between target tasks. We refer to such methods as hierarchical multi-task learning (HMTL). Hence, we will next discuss the potential benefits and challenges of incorporating task hierarchies into the training of machine learning models.

1.2 Benefits of Hierarchies

In general, multi-task learning methods are designed to facilitate knowledge transfer between a set of target tasks. Therefore, the development of new multi-task learning algorithms relies on answering two critical questions: (1) which target tasks should knowledge transfer happen between? And (2) how or in what format should such knowledge transfer take place?

Hierarchies form a multi-level arrangement of target tasks based on a relatedness measure (i.e. similarity). In general, a task hierarchy includes a set of categories and leaves. Each category is designed to group multiple closely related lower-level categories or leaf nodes. Thus, the relationship between two target tasks, A and B, in a hierarchy can be defined as "above", "below", or "at the same level". These relationships can link two tasks either directly or indirectly. For instance, task A can be directly "above" task B defining parent-child relationships. While two tasks "at the same level" of the hierarchy can be directly under the same parent representing a sibling relationship. Such relationships can help multi-task machine learning models by guiding how knowledge transfer should take place between target tasks. In the rest of this section, we review and study (1) different types of task-task relationships defined in a hierarchical task structure that can be used to facilitate the transfer of knowledge and (2) different forms of knowledge transfer that can take place between related tasks.

1.2.1 Types of Task Relationship in Hierarchies

1.2.1.1 Parent-child Relationship Facilitating Top-down Transfer

Each task category represents a more generalized or abstract version of the individual members of a group. By definition, these abstractions are designed to keep common signals among the lower level tasks while pruning detailed information particular to specific members of the group. Thus representing more generalized tasks can facilitate learning more accurate machine learning models. Next, these abstractions can be used in hierarchical multi-task learning models to transfer the captured knowledge to underlying tasks. This allows the modeling for lower-level target tasks to focus on learning additional signals that help identify

them from other members of the same group (siblings). Additionally, categorical tasks will have a higher number of positive examples since they aggregate the positive samples of all their children, which leads to lower class imbalance and can facilitate the learning of more accurate machine learning models.

In a top-down transfer of knowledge, parent-child relationships can be used to train better predictive models for target tasks by transferring useful knowledge learned for their groups(parent). One way to do this is first to learn a model for the parent task (source) and then learn to adapt its model parameters for its children by identifying how the child task differs from its parent. Alternatively, the parent and child can be learned simultaneously while allowing the child to learn and borrow useful features from the category. We will later outline our hypothesis and work focusing on each approach when we discuss this thesis's research goals.

1.2.1.2 Child-parent Relationship Facilitating Bottom-up Transfer

Although learning accurate machine learning models for a parent task might be easier since it represents a more abstract notation, it also introduces the risk of missing critical predictive signals that would otherwise be evident when learning a machine model for the more specific child tasks. Hence, it prompts the development of hierarchical multi-task learning models that can facilitate the bottom-up transfer of knowledge from child tasks to parents. Similar to the top-down transfer mechanism, bottom-up transfer of knowledge can either take place in a two-step process or it can take place in simultaneous learning algorithm that impose similarities and constraints that help parent models to capture such information from its children [55].

1.2.1.3 Sibling Relationship

Other types of relationships can also be used to improve machine learning models [231]. For example, the hierarchical multi-task learning methods can improve machine learning models for target tasks by allowing knowledge transfer between closely related siblings. However, incorporating sibling-sibling relationships also introduces unique challenges. For

example, the usefulness or knowledge transfer between two siblings may represent an entirely asynchronous relationship. That is, while knowledge transfer from a target task with a strong model to its siblings may result in improved models, forcing the stronger target task model to be similar to its weaker siblings can result in a negative transfer. Therefore, hierarchical multi-task learning methods that aim to leverage the relationship between sibling target tasks need to capture this asynchronous nature.

1.2.2 Types of Transfer of Knowledge

The second important question when studying multi-task learning methods is how and in what form knowledge transfer can happen? In general, transfer learning and multi-task learning methods are traditionally categorized based on the answer to "What to transfer?" to the following key groups: (1) parameters transfer, (2) feature transfer, or (3) instance transfer [160, 231]. We will study each approach in detail later in Chapter 2. However, we will briefly describe each method below:

1.2.2.1 Transfer of Model Parameters

In the transfer of model parameters, it is assumed that the machine learning models for related tasks share some model parameters or underlying prior distributions. In the context of MTL, this transfer is often happening by imposing similarities between the target task model parameters through a regularization term in the loss function. The regularization term is designed to minimize the difference between the trained parameters for each individual task's model, hence, preferring machine learning models that are closely related.

1.2.2.2 Transfer of Features

In contrast to the parameter-transfer methods, feature-transfer relies on learning a shared feature representation from the original input features that can maximize the expressiveness for all target tasks. This new feature representation will then be used to learn model parameters for each target task without imposing any similarities. As we review various

feature-transfer methods in Section 2.2.4, this new shared feature presentation can be learned through either a transformation or selection algorithms jointly with the model parameters to optimize the objective function.

1.2.2.3 Transfer of Instances

Ideas explored in the instance-based transfer of knowledge are closely aligned to standard boosting algorithms in which task-specific samples are adopted for similar tasks. Instance transfer methods aim to learn a re-weighting of the borrowed samples that can act as additional training data for related tasks. Thus, the final machine learning model is fine-tuned using a combination of the original target task data and weighted data samples from other related target tasks. While instance transfer methods can be used in MTL, they are more common in the context of traditional transfer learning techniques where a strong source auxiliary task is available, and we aim to learn an improved model for a target task that lacks sufficient data [160].

In hierarchical multi-task learning methods, knowledge transfer can occur either as a parameter or feature transfer across any task relationship. For instance, model parameters can be transferred in a top-down fashion by imposing similarities between parent task and child task machine learning models, assuming that each child task represents a slight variation from its more general direct categorical parent. Similarly, one can also learn global feature representations for each target task category and permit top-down feature transfer by allowing child tasks to use such filters for model learning. This thesis primarily focuses on ideas and methods that leverage either parameters-based or feature-based techniques to guide knowledge transfer.

1.3 Challenges in Hierarchical Multi-task Learning

Hierarchical multi-task learning methods also face unique challenges that have not been addressed by the community. In the rest of this section, we study the existing challenges

toward effectively incorporating hierarchies into the learning algorithms.

1.3.1 Small Sample Sizes

Although a small sample size is a common problem in multi-task learning problems, it can take a slightly different form in HMTL settings. Similar to other MTL problems, a small sample size can be due to a small number of training samples in the overall data or due to the rare nature of the target task. However, in hierarchically structured problems, the issues becomes more extreme for tasks corresponding to the leaf nodes. In a hierarchy, while the number of positive examples for categorical tasks might be adequate, priors for the positive class for leaf tasks are often very low, negatively affecting our ability to learn their preditive models from imbalanced data.

1.3.2 Imperfect Real-world Hierarchies

One possible source of negative transfer in hierarchically structured tasks is the presence of imperfections in task hierarchies. Such flaws can be due to many reasons. First, many task categories (groups) may include outlier tasks. Outlier tasks refer to those target tasks that are not similar enough to the other members of the group. For example, when using the hierarchy of animals in Figure 1 to learn improved image classification models, marine mammals may not be as visually similar as other sub-categories of this group. Therefore, imposing similarities between the model parameters for marine mammals and the rest of the group may result in a negative transfer.

Second, a hierarchy may include groups that are too general. That is, the category may consist of a wide range of lower-level tasks that are not related enough. This category of hierarchy imperfections is usually not a shortcoming of the true underlying task structures but a flaw in the existing hierarchies, whether they are defined by domain experts or generated by algorithms. One way to address this problem is to improve the hierarchy by breaking such groups into two or more meaningful sub-groups. Therefore, an ideal hierarchical multi-task learning approach should be able to simultaneously enhance the task hierarchies in ways that can benefit the training of machine learning models.

Residual groups are another place where imperfections can appear. This is when we create categories that are not devoted to a meaningful sub-class of related tasks but instead they include all those target tasks that could not fit into other more specific categories. Residual groups usually exist when hierarchies are built for purposes other than data analytics, such as visualization or education. Often these groups are named with the prefix "other". For example, in our hierarchy of animals, the group "Other mammals" includes animals such as elephants and giraffes that are not significantly alike. Similar examples are available in health-care hierarchies such as the International Classification of Diseases - 9th revision (ICD-9) [188]. For example, the category "Other Disorders of Central Nervous System" in the ICD-9 hierarchy includes a broad range of patient conditions such as Migraines and Hemiplegia. The former refers to a recurring type of headaches due to genetic reasons. At the same time, Hemiplegia is the weakness or paralysis of half of the body, usually caused by strokes or tumors.

In contrast to the earlier types of hierarchy imperfections, residuals can not simply be solved by improving the group definition or treating group members as outlier tasks. The former approach may not be feasible since, by definition, a better group modification may not exist. On the other hand, treating all members of a residual group as an outlier task will result in learning machine learning models independently. However, being part of a residual group does not entail that the target task can not benefit from the broader group of relevant tasks (parents at a higher level of the hierarchy). Hence a simplistic approach can result in a loss of opportunity for improvement.

In general, task hierarchies can suffer from imperfects for various reasons. Therefore, it is important to develop HMTL methods that are either robust to such imperfections or can correct existing flaws in task hierarchies to prevent negative transfer.

1.3.3 Heterogeneous Relationships

In many domains, one can define various types of task relationships by considering different characteristics of tasks. As a result, multiple task hierarchies can be created. For instance, in our hierarchy of animals in Figure 1. An alternative hierarchy could be created

that considers the visual similarities of animals. In a hierarchy based on visual similarities, animals such as snakes, worms, eels, and caecilians might have been categorized together in a new group that could benefit from learning the classification models from images. Such heterogeneous task hierarchies are also available in many real-world applications. For instance, in patient diagnosis, diseases can be classified by etiological (causal), pathological (by the nature of the disease process), epidemiological (distribution and control), or other types of relationships. In medication, drugs can be classified based on their chemical compounds, mechanisms of action (biological target), mode of action (functional changes they induce), interactions, etc.

While this can represent a challenge in the effective application of HMTL methods, it also offers the opportunity to leverage additional sources of information in these heterogeneous hierarchies to facilitate the learning of more accurate machine learning models. Thus, such heterogeneous relationships between tasks prompt the research of hierarchical multi-task learning methods that can simultaneously use and combine multiple hierarchies to enhance knowledge transfer between target tasks.

1.4 Applications of Hierarchical Multi-task Learning

Today, task hierarchies are broadly available in many areas of science and technology, such as healthcare, computer vision, human activity, and document classification. For example, in healthcare, hierarchies have widely been used in medical ontologies aimed at categorizing clinical concepts such as diagnoses, medications, laboratory results, drug mechanisms of action, etc. Hierarchies are also available in other areas such as computer vision, natural language processing, document classification, and human activity. Ultimately, when standard hierarchies are not readily available, they can often be built from data using hierarchical clustering methods.

Many of these applications can benefit from adopting hierarchical multi-task learning methods while also facing a number of real-world challenges discussed in the earlier section. For example, learning machine learning models for the classification of patient diagnoses is a

hierarchical problem by nature. Clinicians can likely recognize or reject a high-level diagnostic category much earlier and with a higher certainty than more specific diseases that reside on the lower levels of the hierarchy. In fact, structuring the diagnostic process in a top-down manner based on a hierarchy often helps the clinician to make rapid progress in pursuing feasible diagnoses and arrive at diagnostic conclusions even while additional information is required for a final decision on the most reasonable lower-level assignment. Therefore, one would expect incorporating the relationship between parent-child diagnoses embedded in a hierarchy to also benefit learning improved diagnostic models. However, the existing diagnoses hierarchies are not perfect. In fact, disease hierarchies such as the International Disease Classification face many hierarchy imperfection challenges such as outlier tasks and residual groups.

Another problem that can significantly benefit from hierarchical multi-task learning methods is learning predictive models for future patient medication orders, as learning to predict the broad group of medications a patient needs might represent an easier task compared to identifying the exact sub-type. However, learning accurate predictive models for patients' medications can also face many of the challenges discussed in the previous section. For example, predicting future medication orders represents a time-series prediction problem. Thus, target medication tasks can represent significant imbalance problems, often with very few positive samples. Additionally, pharmaceutical drugs can be classified based on multiple characteristics such as their chemical compounds, mechanisms of action (biological target), mode of action (functional changes they induce), and interactions, creating multiple heterogeneous task hierarchies. Therefore, an ideal hierarchical multi-task learning method should be able to handle these challenges.

Similar motivations and challenges also exist in other research areas. For example, in computer vision, existing hierarchies are available in many real-world image classification datasets. ImageNet, a large-scale image classification dataset, used WordNet to obtain the semantically relevant hierarchical structure of target labels. Alternatively, others have attempted to create task hierarchies based on visual similarities between target tasks called visual trees representing heterogeneous hierarchies [57, 83]. Similarly, concepts and words have a hierarchical structure by nature. Therefore, we can often either find or develop hierar-

chical structures for them. For instance, the Wikipedia dataset contains a socially-annotated hierarchical classification of topics [95]. Other news classification datasets such as 10kGNAD [180] and Reuters [195] already provide hierarchical categories for news topics. Task hierarchies can also be found in popular datasets such as URL classification dataset DMOZ [43]. However, the existing hierarchies face a number of challenges discussed earlier. For example, Wikipedia's socially annotated topic classification often includes general categories representing a wide range of contents that can further be divided.

In summary, hierarchical multi-task learning methods can be adopted in a wide range of machine learning problems. However, this dissertation focuses on two novel applications of multi-task learning problems in healthcare: (1) classification of patient diagnoses and (2) prediction of future patient medication orders. First, we evaluate the proposed methods presented throughout the research goals in this thesis in the context of classification of patient diagnoses and diagnostic categories. Finally, in Chapter 6, we further choose the most promising approach and evaluate it in the context of modeling patient future medication orders as a second application of our proposed methods.

1.5 Research Goals and Hypotheses

As discussed earlier, hierarchical multi-task learning methods can offer multiple benefits and introduce new challenges toward learning improved machine learning by leveraging the structure of the hierarchical tasks to guide the transfer of knowledge in a wide range of applications. In this thesis, we generally focus on investigating and proposing new ideas and methods that leverage various types of task-task relationships to facilitate knowledge transfer and address some of these challenges in supervised problems. In particular, we aim to explore new ideas that can answer the following research questions in the context of healthcare applications that are this thesis's primary focus.

- **Question 1:** How can we learn a low-dimensional representation of a patient (patient state) from patient's Electronic Health Record data that can be effective in modelling a large number of prediction tasks including tasks organized in diagnostic and medication

hierarchies?

- **Question 2:** How can we leverage the hierarchical relationships between the tasks to allow transfer of parameters in both top-down and bottom-up fashion?
- **Question 3:** How can feature transfer approaches be adopted in hierarchical multi-task learning methods to improve learning of multiple task models?
- **Question 4:** Can we employ other task relationships such as relations among siblings in combination with the parent-child relationship to further improve machine learning models?

Next, we outline the primary research goals and hypotheses in this thesis, through which we aim to answer the above research questions.

1.5.1 Research Goal 1: Learning Feature Representation from Patient's Electronic Health Records

In this thesis, we focus on applications of supervised hierarchical multi-task learning in healthcare, such as learning machine models that can automatically assign patient diagnoses and diseases using their electronic health records. However, learning from patient's electronic health records is not an easy task and faces multiple challenges:

First, structured EHRs data are high dimensional and contain many diverse time series variables that represent a variety of labs, physiological measurements, symptoms, treatments, procedures, etc. Hence it is not easy to automatically associate the signals in these time series with specific target tasks such as diagnoses. This proves more challenging, especially since many of these signals might carry overlapping information for target task models. For instance, many patients' diagnoses might be confirmed by multiple subsets of patients' clinical data carrying. Therefore, any proposed algorithm must process, combine and incorporate a wide range of patient information that can be recorded as numerical and discrete values and capture underlying patient conditions critical for solving the target machine learning tasks.

Another critical challenge in patients' EHR data is missing and noisy values. There can be numerous underlying reasons resulting in missing values in EHR data. A common

reason for the presence of missing values is errors in data collection. However, missing values in healthcare can also be due to the patient's clinical requirements. For example, certain medications, laboratory tests, or examinations may only be prescribed for specific conditions. Hence, the values for such variables will remain missing if such intervention or tests are deemed unnecessary for a particular patient. In addition to missingness, other errors such as noisy measurements, delayed data entry, and other similar problems can also happen in electronic health records since the data collection for some clinical information may depend on the clinical staff's manual data entry.

Finally, patients' clinical data in the EHR are irregularly sampled, meaning that the frequency and timing of the data collection can vary significantly between different signals. In the case of data collected automatically by bedside monitors, this irregularity may depend on the patient's conditions or device capabilities. At the same time, other data collections, such as medication administration frequency or lab results, may entirely depend on patient conditions or medical staff's availability. For instance, the clinical team might be required to continuously monitor and measure the blood pressure of patients with critical conditions such as hypertension, while the same level of rigorous monitoring may not be necessary for average patients.

Any accurate machine learning approach that depends on learning from a wide range of patients' clinical data in electronic health records must be able to handle these challenges. Therefore, our first research goal in this thesis aims to answer our first research question: "How can we learn feature representations from patient's electronic health records that can be effectively used to model a large number of target tasks for hierarchical applications in this thesis?". We will attempt to answer this question in the context of the following hypothesis:

1.5.1.1 Hypothesis 1: Learning lower-dimensional Feature Representation

In Section 3.5, we hypothesize that patient's high-dimensional electronic health record data can be represented with a smaller set of underlying components that explain the patient's condition and information. To investigate this hypothesis, we propose both unsupervised and supervised techniques that learn a lower-dimensional representation of patients'

EHR from binary summarization of patients' clinical events. The unsupervised method uses an eigendecomposition technique based on singular value decomposition to learn a dense, orthogonal, and lower-dimensional representation of patient's EHR summarizations. On the other hand, the supervised method uses deep neural network architecture based on the Recurrent Neural Network (RNN) with lower-dimension representation sufficient to support task predictions. We provide extensive results by evaluating both techniques in the context of learning diagnostic models for a large set of patient diseases and disease categories. Later in Chapter 6, we evaluate the usefulness of such representations in predicting patients' medication orders when we assess the effectiveness of hierarchical multi-task learning methods for a second novel application in healthcare.

1.5.2 Research Goal 2: Development of Parameter-based Hierarchical Multi-task Learning Methods

In the first research goal of this thesis, we proposed a method aiming to learn expressive feature representations from patients' electronic health records and showed that such representations could capture underlying patient conditions needed to learn accurate diagnostic models. However, the proposed methodology does not consider the hierarchical relationships between target diagnostic tasks. One way the hierarchical task structure can be incorporated into multi-task learning is by facilitating the transfer of model parameters between related tasks within the hierarchy. Therefore, our second research goal aims to explore and develop new hierarchical multi-task learning approaches based on parameter-transfer techniques that can help improve the individual machine learning models.

1.5.2.1 Hypothesis 1: Top-down Transfer of Model Parameters from Parent to Child Target Tasks

Earlier in this chapter, we discussed that task categories are designed to represent a generalized abstraction of their children. Therefore, training a machine learning model for such categorical tasks will rely on capturing the important features common across the entire group of their children[186]. Our first hypothesis related to parameter-based hierarchical

multi-task learning methods is that transfer of model parameters in a top-down fashion can result in learning improved machine learning models for the lower-level target tasks. We study this hypothesis in Section 4.1 and devise a new iterative adaptive hierarchical multi-task learning algorithm that facilitates top-down parameter sharing by imposing similarities between parent and child diagnostic models. We evaluate our proposed method and demonstrate across two EHR datasets that top-down parameter transfer can result in learning more accurate machine learning models for child diagnostic tasks.

1.5.2.2 Hypothesis 2: Bottom-up Transfer of Model Parameters from Children to Parents Target Tasks

Our second hypothesis in this research goal is that similar benefits can also be gained by facilitating the bottom-up transfer of model parameters. The motivation behind this hypothesis is that the aforementioned generalized task categories may fail to capture essential features for accurately predicting sub-types of diagnostic groups. Therefore, by allowing bottom-up transfer of parameters, we allow such important information to be shared with the parent machine learning models. We explore this question simultaneously with hypothesis 1 in Section 4.1 and demonstrate through extensive quantitative and qualitative results that the transfer of model parameters was helpful in both the top-down and the bottom-up transfer.

1.5.2.3 Hypothesis 3: Asymmetric Class-Dependent Similarities Between Task Predictions Across Samples

Finally, we hypothesize that related tasks' (models) prediction scores organized in expert-defined hierarchies do not have the same level of similarity among different classes of samples. For example, when transferring model parameters in a top-down fashion by imposing similarities in model behaviors, such similarities might be stronger for negative samples. This is because, in a hierarchy, a negative parent class directly translates to a negative label for all of its children. While if the parent class is positive for a sample, it does not necessarily result in a positive label for all of its children. In fact, it only requires one of the children to

be positive. Thus, imposing similarities between parents and children should consider such conditions. To evaluate this hypothesis, we first study the behavior of our proposed hierarchical multi-task learning method in Section 4.1 and demonstrate that the proposed imposed similarities between model parameters are equivalent in learning from the weighted average of predicted scores of the auxiliary task. However, blindly imposing similarities in predicted scores of parent and child diagnostic models can be detrimental since such similarities may not always hold. For instance, a negative label for a parent diagnostic task will translate to a negative label for its children. However, this may not always be true for positive labels. In fact, by design, a positive label for the parent diagnostic model means that at least one of its children is positive and some not. We further investigate this hypothesis in Section 4.2 by proposing a new class-dependent of our hierarchical multi-task learning method and providing substantial results in the context of patient diagnoses assignment problems to evaluate our claims.

1.5.3 Research Goal 3: Development of Feature-based Hierarchical Multi-task Learning Methods

Another way learning can take place in hierarchical multi-task learning methods is via feature transfer. In contrast to parameter transfer methods, where knowledge transfer happens by imposing similarities between model parameters, model parameters are trained independently in feature-transfer methods. However, feature-transfer methods leverage the commonalities between related tasks to facilitate the co-learning of a number of shared feature representations used by each target task to learn separate machine learning models. Ideally, this common feature representation should capture features that help identify and differentiate closely related target tasks. In our example of learning to identify citrus fruits, this common feature representation might learn to capture information such as size, color, presence of pulp pulp color and taste that help the model first identify that the fruit is a citrus fruit and also facilitates differentiating an orange from a lime or lemon. In this research goal, we aim to answer this thesis's final two research questions. First, we explore ideas in the context of deep neural network models that facilitate knowledge transfer across

the hierarchy using feature-transfer techniques. Next, we investigate whether we can allow the model to further improve its prediction by learning to better differentiate itself from its siblings and answer the final research question in this thesis: "Can we employ other task relationships such as siblings in combination with the parent-child relationship?".

1.5.4 Hypothesis 1: Top-down Transfer of Shared Feature Representations

When target tasks are organized in a hierarchical structure, each categorical node represents a level of similarity between a subset of the tasks. Thus, the hierarchy can be used to identify target task groups with various levels of similarities. Here, we hypothesize that the hierarchical structure of tasks can be used to guide the co-learning of shared feature representations between various groups of the target tasks that can result in learning improved target task models. In other words, we hypothesize that by leveraging such a multi-level group of similar task, we can facilitate learning of common feature representations across task categories that contain important features for differentiating each target task from their direct and indirect siblings. In Chapter 5, we investigate this assumption and explore new ideas in the context of hierarchical multi-task deep neural networks that facilitate feature transfer by learning shared feature representations for each group of the diagnostic model. The proposed architecture allows each target task to either use the features from the group (parent), learn new task-specific features from the input, or combine these two features to learn improved target task models.

1.5.5 Hypothesis 2: Modeling Interactions Between Siblings

Next, we hypothesize that accurate learning of target task models may rely on capturing the sibling-sibling interactions in the context of hierarchical problems. Therefore, motivated by the field of differential diagnoses, we develop a new interaction learning deep neural network layer in Section 5.1.3. The proposed method uses the initial predictions of siblings to find additional helpful information in patients' clinical data and further improve the target task models.

Finally, we evaluate our proposed top-down transfer of features and interaction learning

layer extensively in the context of assigning patient diagnoses and diagnostic categories and show that the proposed methods can improve the final classifications significantly.

1.6 Outline

In this chapter, we described the motivation behind hierarchical multi-task learning and provided an overview of the benefits and challenges that will be introduced when incorporating hierarchical task structures in the learning algorithms. We organize the remaining chapters of this thesis as follows.

First, in Chapter 2, we review existing work related to hierarchical multi-task learning and learning from electronic health records and study the relationship between HMTL and other well-known machine learning problems such as multi-class classification, multi-label classification and hierarchical classification and finally demonstrate existing gaps in the fields of multi-task learning and transfer learning that motivates the study of hierarchical multi-task learning methods.

In Chapter 3, we propose a flexible approach for learning models and representations from a wide range of patient's clinical information stored in electronic health records and demonstrate the effectiveness of the proposed approach in context of patient diagnosis problem that assigns diagnoses and diagnostic categories to patient's EHR.

By leveraging the representations developed in Chapter 3, Chapter 4 develops and presents multiple hierarchical multi-task learning methods that utilize the transfer of model parameters in order to improve the performance of the ML models built for individual target tasks. We evaluate of our methods on the patient diagnosis problem and demonstrate the improved performance of the methods when compared to solutions from Chapter 3.

In Chapter 5, we continue our investigations of HTML methods by proposing a new hierarchical deep multi-task learning method that adopts a feature transfer approach to facilitate knowledge sharing instead of parameter transfer. Through experiments on the patient diagnosis problem we show that the new HTML feature transfer approach outperforms our previous solutions presented in both Chapters 3 and 4.

Encouraged by results obtained using methods in Chapter 5, in Chapter 6 we investigate the possibility of applying the feature transfer methods to a new healthcare-related problem, the medication order prediction problem, that aims to predict future patient medications orders from EHRs. The hierarchy used in this problem organizes and abstracts the underlying medications into different medication categories.

Finally, Chapter 7 summarizes the achievements of the new methodologies presented in the thesis, and discuss challenges and open problems that can be the topics of future research investigations.

2.0 Background

This chapter provides a detailed overview of background work related to this thesis. We start with providing basic notations used throughout this work, followed by a detailed review of related work in two primary groups: (1) existing research related to the hierarchical multi-task learning methodology, and (2) past and recent research closely aligned with learning patient feature representations from electronic health records.

First, we describe how multi-task and hierarchical multi-task learning are closely aligned with other well-studied machine learning problems, including multi-class classification, multi-label learning, and hierarchical classification. Next, we describe each domain in detail, review some of the well-known approaches proposed by the respective research communities, and discuss how they are connected to multi-task learning methods. Finally, after reviewing standard techniques in transfer learning and multi-task learning, we describe the shortcomings in these areas that motivate the adoption of hierarchical multi-task earning approaches and close this section with a review of the existing research.

In the second half of this chapter, we explore existing work in the field of learning dense feature representations from patient's electronic health records.

2.1 Notation

The following notation will be used in the rest of this document:

- We denote vectors using lower-case Latin letters (i.e. a), and use subscripts to denote individual elements of the vectors (e.g. a_i is the i_{th} element of vector a). Additionally, we use upper-case letters for matrices. Similarly to vectors, we denote an individual item in matrix A at row i and column j as A_{ij}. Finally, for both vectors and matrices, the superscript ^{T} denotes the transpose (i.e. A^T) and the inverse of the matrix is shown with superscript $^{-1}$ (i.e. A^{-1}).

- Special norms used throughout this work include: $||.||_2$ and $||.||_1$ which correspond to the l_2 and l_1 norms.
- In the multi-task learning settings we use T to refer to the total number of target binary classification tasks $t_1, t_2, ..., t_T$. These are organized in a hierarchical structure H. We use $\phi(t, H)$ and $\rho(t, H)$ to refer to the set of parent and the set of children of task t. Additionally, we use $\phi^*(t, H)$ and $\rho^*(t, H)$ to refer to the set of all ancestors and children of task t while $\phi(t, H) \subset \phi^*(t, H)$ and $\rho(t, H) \subset \rho^*(t, H)$.
- We use $D_t : \{X_t, P(X_t)\}$ to denote the domain of task t in which X_t refers to its feature space of task t and $P(X_t)$ is the marginal probability distribution. However, in the majority of the sections of this thesis, we assume all tasks have the same feature space.
- Finally, the common objective of methods proposed in this dissertation is to learn T discriminant functions $f_1, f_2, ..., f_T$ in which $f_t : X : \mathbb{R}^D \rightarrow \mathbb{R}$, where, D corresponds to the dimensionality of feature space for individual tasks. A key assumption here is that tasks do share the same feature space while it is not necessary for them to have the overlapping samples.

2.2 Hierarchical Multitask Learning

Hierarchical thinking in humans is believed to be facilitating more efficient learning of new experiences [19, 189]. Additionally, the hierarchical categorization of concepts has become a crucial part of many aspects of our lives. In health care, hierarchical classification of diseases, medications, and medical procedures is used for decision making or public health research [99, 21]. On the other side, the hierarchical classification of animals and other living beings has become an integral part of research and education in biology, and animal studies [178]. Motivated by the abundance of existing task hierarchies and our brain's efficient learning processes, we seek to answer the following question: Can we use hierarchical task structures to improve the learning of classification models that cover a large number of tasks? We seek to learn tasks that can be either linked to leaf nodes of the hierarchy or both leaf and higher level category nodes.

Hierarchical multi-task learning is directly related to the following areas of machine learning research:

- **Multi-class Classification:** The simplest way that task hierarchies can be used in machine learning is to solve multi-class learning problems with a large number of classes. Multi-class classification (MCC) algorithms aim at assigning exactly one class for each input example [45, 1]. That is our goal is to learn a function $f : X \rightarrow \{1, ..., T\}$, in which X corresponds to the input features and T to the number of classes we want to use in the classification. Hierarchical class structures can be used to model the class dependencies in multi-class problems.

  For example, In the animal classification problem in Figure 1, our goal is to classify each image into exactly one animal class located at one of the leaves of the hierarchy. Intuitively we can use the given hierarchy to narrow down the search space for the classification of a new image by starting from the root and recursively classifying at each intermediate node corresponding to the sub-category that best matches the given sample.
- **Multi-label Classification:** In contrast to multi-class classification, multi-label classification (MLC) is the area of machine learning in which our goal is to learn a function $f : X \rightarrow \{0, 1\}^T$ from data that assigns to each instance a binary vector of T class labels [200, 78].

  In general, class labels assigned to an instance can be dependent. These class label dependencies can be modeled using hierarchical structures [201, 24]. In this case, classification labels can be included either in the hierarchy's leaf nodes or both leaf and internal categorical nodes. In hierarchical multi-label classification, similarly to MCC methods, we could assign an instance to multiple labels by recursively using classifiers at the internal nodes until leaf nodes are reached.
- **Multi-task Learning:** Multi-task learning (MTL) denotes to the area of machine learning research that aims to learn a set of T binary target task models $\{X_t \rightarrow \{0, 1\}\}^T$ simultaneously by leveraging the similarities between the tasks. The main difference between MTL and MLC is that in MTL the input data X_t for the task t can be different from the other target tasks. In hierarchical multi-task learning, hierarchical task structures can be used to guide the transfer of knowledge between tasks to improve individual

model performances and prevent negative transfer.

In the rest of this chapter, we will first introduce the notation that will be used in the remaining text. Next, after reviewing existing methods for incorporating hierarchical task structures into multi-class and multi-label problems, we will introduce transfer learning in Section 2.2.3 as the fundamental idea behind multi-task learning methods. Finally, Section 2.2.4 provides a formal definition of the multi-task learning problem and reviews the existing techniques. Lastly, we review the recent work incorporating task hierarchies in multi-task learning methods.

2.2.1 Multi-class Classification

The objective of multi-class classification (MCC) is to learn a function $f : X \rightarrow \{1, ..., T\}$, that assigns exactly one class label to each input sample x from data[45, 1].

Hierarchical multi-class classification refers to a sub-type of multi-class classification problems where classes form a hierarchical structure [185]. Therefore, in contrast to the more standard ways of handling multi-class problems such as learning the discriminant functions directly for all classes from data or One-vs-Rest and One-vs-One settings where each class is compared to all other classes as a whole or in pairs, the hierarchical classification incorporates the knowledge about the hierarchical structure of tasks to derive these comparisons. Consequently, at each node, a simple classifier makes the determination between different child class categories. Therefore, following a top-down path from the root of the tree to a leaf node determines the class of a new sample.

The most common method for defining classification models within a hierarchy is to use the top-down approach [96]. In this case, a classifier on a low level of the hierarchy is defined using a decision or the signal generated by its parent classifier. There are different versions of the top-down approach that place various consistency constraints on predictions of the parent and child tasks and their classifier outputs, most frequently assuring the probability of a parent task is higher than the probability of a low-level class or class category [51, 209].

Kumar et al. introduced a binary hierarchical classifier (BHS), a hierarchical classification algorithm that automatically learns a binary tree by iteratively splitting the set of classes

that best discriminate the two groups according to the Fisher discriminant. At the time of inference, binary classifiers are used in a top-down fashion to determine the final class of the new sample [98]. A similar method, hierarchical support vector machines (HSVM) was proposed by Chen et al. that automatically learns a binary tree structure for the set of target classes [28]. Then HSVM trains a binary SVM model for each internal node of the tree. Their clustering approach to learn the tree hierarchy of classes consisted of two steps. First, they created an undirected graph of task-task similarities using Kullback-Leibler distance and then employed the max-cut algorithm to split the graph into sub-clusters.

An alternative group of methods was proposed that made use of the hierarchical structure of the tasks by training machine learning models for each node of the hierarchy and by imposing constraints between parent and child models [172, 66, 57]. Xiao et al. have developed an interesting approach based on the HSVM algorithm that added an orthogonal regularization term between parent and child models into the objective function. To understand their method, we first review the standard support vector machine (SVM) algorithm for binary classification tasks [181, 25].

A linear classifier aims to learn a hyperplane that can separate the two classes. One reasonable choice is to find the hyperplane that creates the maximum separation. In order to do this SVM algorithm tends to find the maximum-margin hyperplane that has the maximum distance from the nearest data points called support vectors. Therefore, SVM is often formulated as a constrained optimization problem shown in Equation 1 in which w refers to the model parameters and $y \in \{-1, 1\}$. The constraint $y_i w^T x_i \geq 1$ ensures that each sample i is located on the right side of the hyperplane.

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}||w||_2 \\ s.t. \quad & y_i w^T x_i \geq 1 \quad \forall i \in \{1, ..., N\} \end{aligned} \tag{1}$$

In order to extend SVM to cases in which the data are not linearly separable, we introduce the soft-margin SVM algorithm in which we relax the constraints by allowing an error variable ε_i, for instance, i. Equation 2 shows the constraint optimization problem for soft-margin SVM. The soft-margin SVM aims to find the maximum-margin hyperplane that minimizes

error variables. In the rest of this document, we use SVM to refer to the soft-margin SVM algorithm.

$$\begin{aligned} \min_{w,\varepsilon} \quad & \sum_{i}^{N} \varepsilon_i + \frac{1}{2}||w||_2 \\ s.t. \quad & y_i w^T x_i \geq 1 - \varepsilon_i \quad \forall i \in \{1, ..., N\} \\ & \varepsilon_i \geq 0 \qquad\qquad\qquad \forall i \in \{1, ..., N\} \end{aligned} \tag{2}$$

The soft-margin SVM can also be formulated as an unconstrained optimization problem using the hinge loss function $max(0, 1 - y_i w^T x_i)$. This function is zero for each sample if it is correctly classified on the right side of the hyperplane. Otherwise, it is equal to the error variable ε_i and proportional to the data point's distance to the hyperplane. The unconstrained formulation for SVM's objective function is shown in Equation 3.

$$\min_{w} \quad \frac{1}{2}||w||_2 + \sum_{i}^{N} max(0, 1 - y_i w^T x_i) \tag{3}$$

However, in the rest of this document, we commonly use the constrained optimization formulation since it is easier to understand its extensions that tackle transfer learning and multi-task learning problems by introducing new constraints.

The optimization function for Xiao et al.'s method is formulated as a constrained SVM optimization problem as follows:

$$\begin{aligned} \min_{\varepsilon,\ w_1, w_2, ..., w_{|Y|}} \quad & \frac{C}{N}\sum_{i}^{N} \varepsilon_i + \frac{1}{2}\sum_{y \in Y} ||w_y||^2 + \sum_{y \in Y} \sum_{a \in \phi^*(t,H)} w_y^T w_a \\ s.t. \quad & w_y^T x_i - w_j^T x_i \geq 1 - \varepsilon_i, \\ & \qquad \forall j \in S(j), \quad \forall y \in \phi(y_i, H), \quad \forall i \in \{1, ..., N\} \\ & \varepsilon_i \geq 0, \qquad\qquad\qquad\qquad\qquad\qquad \forall i \in \{1, ..., N\} \end{aligned} \tag{4}$$

Where Y is the set of classes in a multi-class problem setting and $S(y)$ refers to the set of class siblings y. The goal of the optimization problem in Equation 4 is to learn all model

parameters $\{w_1, ..., w_Y\}$ simultaneously. The first term of the minimization function belongs to SVM's slack variables that minimize the hinge loss and the second term is regularizing the complexity of the model parameters. Finally, the third term $\sum_{y \in Y} \sum_{a \in \phi(t,H)} w_y^T w_a$ is imposing orthogonality between model parameters w_y for class y and model parameters of all of its ancestors defined as $\phi^*(t, H)$.

The main problem with the top-down approaches is that learning higher-level class models from data may omit details that only low-level class models can capture. For example, some of the findings for a patient may point expressly and with high accuracy to a low-level diagnosis, while the higher-level class model may ignore the same conclusions and, as a result, may not include it in the model. In such cases, the probability of a lower-level class may be higher than the probability of a higher-level class category violating the constraint consistency.

One way to correct for child-to-parent effects is to define and add a bottom-up process that assures positive lower-level class predictions aggregate properly in the parent tasks [202]. However, pure bottom-up approach would require the presence of accurate classifier models on the leaf classification layer, which is hard to achieve in practice when datasets of a limited size are used to train such models and the count of positive instances for such classes are very low.

There exists a variety of hierarchical classification methods that try to account for both the top-down and bottom-up classification processes. One example is the Bayesian aggregation method by [41] that compiles the hierarchy into a Bayesian belief network and uses inferences to support the classification on different levels of the hierarchy.

The limitation of the vast majority of current methods is that the classification models are dependent on related models both during the learning and the application stage. One advantage of the hierarchical multi-task learning methods we develop in this work is that while it considers the model interactions during the training stage, it leads to separate models that can be applied independently.

2.2.2 Multi-label Classification

The goal of the multi-label classification methods (MLC) is to learn a function $f : X \rightarrow \{0, 1\}^T$ that assigns to each instance x a binary vector of T class labels [200, 78].

Similar to multi-class problems, the existing multi-label classification methods can be divided into two categories. The early work in multi-label classification methods assumed that target labels are independent of each other [20, 33]. However, in general, labels can be dependent, and assuming independent relationships may not produce correct results [12]. The second group, on the other hand, sought to utilize or learn the underlying dependencies between target labels [63, 227].

One way to model label relationships is the two-level methods that independently learn models for each label at the first level and use the outputs of those models as extensions of original features to learn the second level of models [63]. Multi-label extensions of k-nearest neighbor methods have also been proposed [227]. Another group of methods tackles this problem using multi-dimensional Bayesian networks [204, 17]. An alternative approach to MLC is based on the error-correcting output coding (or simply output coding) [81, 193]. The idea is to encode the output values into a codeword, learn how to predict the codeword, and then recover the correct output from noisy predictions. However, one shortcoming of output coding methods is that they can only predict the single best label for each sample instead of the probability for all labels. Despite the efforts to incorporate label dependencies, the majority of these approaches do not consider complex hierarchical task structures while considering such complex hierarchical relationships can be vital for learning accurate machine learning models [109]

More recently, new methods have been proposed that consider the hierarchical structure of labels. A well-known hierarchical multi-label classification method is the HOMER algorithm proposed by Tsoumakas et al. for document classification [201]. HOMER takes advantage of the task hierarchy by recursively training a multi-label model at each internal node. Other methods were also proposed for hierarchical multi-label problems. For example, Bianchi et al. presented a hierarchical extension of the hamming loss function for multi-label classification problems. The idea of hierarchical loss is based on the notion that whenever a

classifier makes a mistake at any node in a given hierarchy, no further loss should be counted for any error in the subtree rooted at that particular node [24] . This is analogous to the top-down recursive algorithm of using internal node classifiers to find the final set of labels for a new example.

Decision tree-based algorithms were also proposed to solve hierarchical multi-label problems [206, 46]. For instance, Vens et al. studied three different ways of learning decision trees for hierarchical multi-label classification problems [206]. Additionally, Dimitrovski et al. exploited the classification hierarchy by building an ensemble of predictive clustering trees (PCT) that can simultaneously predict all different levels in the hierarchy [46].

Another family of multi-label classification methods that can be adopted for hierarchical label structures is classifier chain algorithms (CC) [171, 226]. The classifier chain methods first find an ordering of labels. Next, they train the classification models for each task in order and feed the output of preceding labels' classifiers to the classifiers that depend on them. This helps improve the results of some dependent classifiers by using the output of stronger models as input features. In order to extend these methods to hierarchical problems, one can take advantage of existing hierarchical structures to find an appropriate order for the classifiers.

2.2.3 Transfer Learning

The motivation behind transfer learning methods, as famously discussed in an article with the title "Learning to Learn", is the need to study methods that enable the reuse of previously trained models in the learning process [197]. For the sake of simplicity in the rest of this section, we will often assume that only one source task is available.

2.2.3.1 Definition and Types of transfer learning

We borrow the following definition of transfer learning from [160]:

Definition 2.2.1. Transfer Learning: Given a source domain D_s and learning task f_s, a target domain D_t and learning task f_t, transfer learning aims to help improve the learning

of the target predictive function f_t in D_t using the knowledge in D_s and f_s, where $D_s \neq D_t$, or $f_s \neq f_t$.

In transfer learning $D_s \neq D_t$ happens when either the feature spaces X_t and X_s are different or the tasks have different marginal probabilities. An example of the case feature spaces are different between source and target tasks can be a document classification problem in which source and target tasks are in other languages. Another group of transfer learning methods is inductive transfer learning. In inductive transfer learning, the difference between source and target tasks relies on underlying f_t and f_s functions [160].

An alternative categorization of machine learning methods is based on the answers to the question "What to transfer?". Hence, transfer learning methods are divided into four groups: parameter transfer, instance transfer, feature representation transfer, and relational knowledge transfer. Parameter transfer methods often assume that related tasks should have similar model parameters. Therefore, existing methods tend to adapt and fine-tune model parameters trained for the source task to train the target task model. Instance transfer approaches, which also assume common feature space between target and source tasks, are often used when the marginal probabilities for the two tasks are different. Therefore, they often tend to improve the target task model by reusing some samples from the source task. Existing instance-transfer methods operate similarly to boosting algorithms by choosing which samples need to be transferred [37, 218, 161]. Feature transfer methods attempt to learn efficient feature representations for the source task that can be adopted by the target task [130]. This idea has also been adopted in recent deep learning algorithms. We will review these techniques in more details in Section 2.2.3.4. The last category of techniques is transferring relational knowledge. Methods under this category are usually used in problems that i.i.d assumptions do not hold, and samples have relations and dependencies. Common examples can be found in social network and graph problems in which relations are usually transferred from one task to another [40]. The methods proposed in this dissertation fit into the model parameter transfer learning category. Hence, in the rest of this section, we review some of the existing inductive transfer learning methods that rely on the transfer of model parameters. Generally, methods in the model parameter transfer category are closely related to the multi-task learning domain. However, in the next subsection, we only focus

on methods designed for transferring parameters between source and target tasks and keep multi-task learning methods for Section 2.2.4.

2.2.3.2 Model Parameter Transfer Methods

The first group of parameter transfer methods relies on the hierarchical Bayesian framework [176, 175, 36, 167, 138, 90]. The underlying idea of these methods is to obtain a strong and informative prior from source tasks to train the target task model. Rosenstein et al. proposed a transfer learning method based on the Naive Bayes algorithm that imposed similarities between the target and the source task by encouraging their model parameters to be similar. This is done by assuming model parameters are drawn from the same hyperparameter distribution with an unknown mean, but a small variance [175]. Dai et al. proposed an EM algorithm for fine-tuning the pre-trained Naive Bayes model based on a single source task's data to fit the target task [36]. Raina et al. proposed a new logistic regression-based transfer learning approach by imposing an informative prior over the parameters. Their new algorithm automatically constructs a multivariate Gaussian prior with a full covariance matrix for a given target task by using other known similar tasks [167]. Similarly, Marx et al. used pre-trained logistic regression models for source tasks to obtain a Bayesian prior for the target task by averaging their model parameters [138]. Other Bayesian methods impose similarities by assuming model parameters of the source tasks and the target task are drawn from the same distribution [90, 176].

The second approach of transferring model parameters enforces similarities by using regularization techniques [225]. The general idea is that we can derive the learning of the parameters of the target task model by penalizing their difference from those of the source task. Adaptive Support Vector Machine (A-SVM), first proposed by Yang et al. [215] is a transfer learning algorithm that learns the function f_t for a target task t by taking advantage of pre-trained models for a set of auxiliary tasks. The idea of an adaptive support vector machines (A-SVM) algorithm is to learn an enhanced SVM model for target task t using both the features and prediction scores of the set of related source or auxiliary tasks as input. In other words, A-SVM learns a new function Δf_t to predict how much the predictions for

target task t should differ from the predicted scores of its auxiliary tasks. Therefore, it defines $f_t = \sum_{a \in A} \tau_a f_a + \Delta f_t$ in which τ_a determines the contribution of an auxiliary task a while $\sum_{a \in aux(t)} \tau_a = 1$. The generalized version of A-SVM for multiple auxiliary tasks can be formulated as a constrained SVM optimization problem (see 2) as shown below:

$$\begin{aligned} \min_{v_t, \varepsilon} \quad & \sum_{i}^{N_t} \varepsilon_i + C||v_t|| \\ s.t. \quad & y_i \sum_{a} \tau_a f_a(x_i) + y_i v_t^T x_i \geq 1 - \varepsilon_i \quad \forall i \in \{1, ..., N\} \\ & \varepsilon_i \geq 0 \qquad\qquad\qquad\qquad\qquad\quad \forall i \in \{1, ..., N\} \end{aligned} \tag{5}$$

In 5, $\Delta f_t = v_t x_i^T$, C determines the balance between minimizing the regularization term $||v_t||$ and the loss function, while, τ_a denotes to the weight of auxiliary task a expected to be provided as input or tuned using hyperparameter optimization techniques. Larger values of C result in stronger regularization of model parameters which forces more similarities between f_t and $\sum_a \tau_a f_a(x_i)$. On the other hand, smaller values of C allow f_t to differ from its auxiliary models. Although A-SVM is able to use any arbitrary auxiliary model as input, if all f_a functions are linear SVM models one can calculate $w_t = \sum_{a \in aux(t)} \tau_a w_a + v_t$. This allows us to use task t's model independently from its source models. Later, Duan et al. proposed a series of multiple kernel learning methods by learning the target task model as a combination of multiple source models based on A-SVM. [49, 50, 48].

The third group of methods, commonly known as hierarchical transfer learning, was also proposed [199]. The idea in hierarchical transfer learning is to learn a composite or higher-level target task category by first learning multiple simpler source tasks and combining them to perform a target task. One way to do this is by feeding the prediction scores of the source models as features of the target task [191].

2.2.3.3 Negative Transfer

Negative transfer happens when transfer learning methods contribute to reduced performance of target task models [160]. This usually occurs when imposed similarities during the

learning of the target task model did not truly exist [175]. Past work in transfer learning has tackled this problem using various methods, including learning task relatedness and limiting transfer only from related source tasks. After reviewing recent work in multi-task learning, we will discuss these methods in more detail in Section 2.2.4.

2.2.3.4 Transfer learning for Deep Neural Networks

Recently transfer learning has also gained the attention of the deep learning community. This is primarily due to the fact that training in deep learning methods often requires large-scale datasets. However, adequately large datasets are not always available. Therefore, adapting pre-trained deep learning models for new tasks or re-using data from existing larger datasets is often the only feasible solution.

More recent work has proposed transfer learning methods for deep learning models based on instance-based transfer, mapping-based transfer, network-based transfer, and adversarial-based transfer [194]. Instance-based transfer methods similar to the standard transfer learning methods and boosting algorithms find a re-weighting of the source dataset that can be used for training the target task [37, 121]. The mapping-based methods find an intermediate shared representation between the source and the target task. The idea is that these two tasks or their domains might be more similar than they appear to be [128, 131]. On the other hand, network-based methods rely on the idea of re-using pre-trained networks [220]. Similar approaches have been proposed in Natural Language Processing community [82, 157, 60]. For instance, Huang et al. proposed a cross-language knowledge transfer by splitting the network into two language-independent and language-dependent parts [82]. Yosinisky et al., in their article "How transferable are features in deep neural networks?" studied the relationship between the network architecture and transferability [220]. They transferred pre-trained deep learning networks to the target task by re-using early layers of Convolutional Neural Network models and fine-tuning or re-training the later layers for a target task. Their study examined the transferability of different layers of deep neural network models. The results demonstrated that since earlier deep learning layers tend to learn more straightforward and generalizable visual features such as colors, lines, and corners [224], we can adopt them for

a new target task. Therefore, instead of having to train a very complex neural network from scratch, we can only fine-tune or re-train the later task-specific layers.

2.2.4 Multi-task Learning

Multi-task learning (MTL) aims to learn multiple related tasks simultaneously. The motivation is to exploit task relationships, their commonalities, and differences. This has shown promising results in improving models of individual tasks compared to the standard methods of learning each task independently [231]. At the same time, many multi-task learning methods have been proposed for various categories of machine learning problems, such as: supervised tasks, unsupervised tasks, semi-supervised tasks, reinforcement learning and etc. This thesis focuses on multi-task learning for supervised machine learning problems. The intuition behind MTL is both related to that of transfer learning and the fact humans can learn multiple tasks jointly.

More formally we define multi-task learning as follows:

Definition 2.2.2. Multi-task Learning: Given T supervised learning tasks $t_1, t_2, ..., t_T$ with similar or different domains, multi-task learning aims to help improve the learning of machine learning models $f_1, f_2, ..., f_T$ by using the information from all related tasks.

The methods of multi-task learning can be categorized into three categories based on the question "what to share?". These categories are instance based, feature-based, and parameter-based. In the rest of this section, we first review in more detail the existing categories of supervised MTL methods. Next, we visit the problem of negative transfer in the context of MTL and discuss ideas of existing methods to prevent it. Finally, we review existing methods for hierarchical multi-task learning (HMTL).

2.2.4.1 Instance-based Multi-task Learning

Similar to the transfer learning methods in the same category, Instance-based methods learn improve target tasks by learning to use samples from other tasks and re-weighting them. However, very few publications fit into this category. Most notably, Bickel et al.

proposed an instance-based method that first matches each sample from all tasks to the target distribution of any given task. Finally, it uses the weighted samples to train models for each target task separately [16].

2.2.4.2 Multi-task Feature Learning

Multi-task feature learning (MTFL) tackles the multi-task learning problem by learning a shared feature representation that allows better learning of the target task models and facilitates sharing between the tasks. Argyriou and Evgeniou proposed a multi-task feature learning method that attempts to learn a lower-dimensional feature space that is shared across all tasks [2]. Their underlying assumption was that related tasks share a common feature space. Hence f_t can be represented as $f_t = w_t h(x_i)$ in which w_t is the model parameters for task t and $h(.)$ is a shared feature transformation function for all tasks that leads to a new lower-dimensional space, that is smaller than the original feature space. The multi-task feature learning method is formulated as a regularization problem with the following objective function:

$$\min_{w_1,w_2,...,w_T} \quad \sum_{t}^{T}\sum_{i}^{N_t} L(y_{ti}, w_t U^T x_{ti}) + \gamma \sum_{t}^{T} ||w_t|| \qquad s.t. UU^T = I \tag{6}$$

in which U is the feature transformation matrix, x_{ti} and y_{ti} correspond to features and a label assigned to sample i from task t and finally γ is the model parameter to set the trade-off between regularization of w_t and loss function L. The MTFL method in Equation 6 aims to jointly learn the target task models by minimizing classification losses of all tasks $\sum_t^T L(y_{ti}, w_t U^T x_{ti})$ and regularizing the model parameters in the second term. MTFL models each task function as $f_t(x_{ti}) : w_t U^T x_{ti}$ in which $U \in \mathbb{R}^{d \times D}$ and d represents the dimensions of learned feature space while D refers to the dimensions of the original input data (d can be smaller than D). The $UU^T = I$ constraints insure orthogonality of matrix U which is designed to avoid learning of redundant features.

Following the introduction of the MTFL method, Argyriou et al. proposed a convex formulation for the multi-task feature learning problem by minimizing the trace of square matrix $V^T UV$ in which $v_i = Uw_t$ [3] . Moreover, other multi-task sparse coding methods

have been proposed to learn the linear transformation of features [139]. An alternative approach to MTFL is learning the task parameters and shared lower-dimensional features using matrix factorization methods. For instance, Jin et al. proposed an MTFL method that learns a shared feature representation for multiple tasks with heterogeneous feature spaces [86]. Gong et al. proposed rMTFL method that attempts not only to learn the shared feature space but also to identify the outlier tasks. They do this by imposing a group Lasso penalty not only on the feature transformation matrix but also on columns of model parameter weights to regularize entire feature vectors for outlier tasks [65].

Multi-task feature selection methods have also been proposed to choose features collectively important to all the tasks. Obozinski et al. proposed to co-regularize columns and rows of the matrix of model parameters matrix for all tasks in W to achieve this goal [155]. Equation 7 shows the objective function for MTFS method.

$$\min_{w_1,w_2,...,w_T} \quad \sum_{t}^{T}\sum_{i}^{N_t} L(y_{ti}, w_t x_{ti}) + \gamma||W||_{2,1} \tag{7}$$

MTFS uses a $l_{2,1}$ norm regularization regularization term on the matrix $W \in \mathbb{R}^{D\times T}$ which jointly selects the important features by first applying an l_1 norm on rows of matrix W. Then, it learns the importance of each selected feature using an l_2 norm regularization on the columns of the matrix or the individual tasks' model parameters. Later multiple learning algorithms were proposed to solve the problem in 7 [156, 114].

2.2.4.3 Parameter-based Multi-task Learning

In the parameter-based methods, the transfer of knowledge is done by encouraging model parameters to be similar to each other. Evgeniou and Pontil proposed regularized multi-task learning (RMTL) method based on the Support Vector Machines (SVM) algorithm that imposes task similarities by regularizing the differences between the target tasks' model parameters and their group [55]. RMTL defines the function for task t as $f_t = f_0 + \Delta f_t$ in which f_0 is the the category model and Δf_t is learning how f_t differs from f_0. Equation 8 shows RMTL's objective function by extending the constrained optimization formulation for

a single task SVM algorithm (see section 2.2.3.2). RMLT's objective function is simultaneously learning a set of maximum margin hyperplanes for all T target tasks by learning how each target task differs from the average model.

$$\begin{aligned} \min_{\varepsilon, w_0, v_1, v_2, \ldots, v_T} \quad & \sum_{t}^{T} \sum_{i}^{N_t} \varepsilon_{ti} + \lambda_1 ||w_0||_2 + \frac{\lambda_2}{T} \sum_{t}^{T} ||v_t||_2 \\ s.t. \quad & y_{ti}(w_0 + v_t) x_{ti}^T \geq 1 - \varepsilon_{ti} \\ \forall t, i \quad & \varepsilon_{ti} \geq 0 \end{aligned} \tag{8}$$

In Equation 8 w_0 represents the model parameters for the group model while v_t corresponds to the parameters of Δf_t. Hence, we can obtain the final model parameters for task t as $w_t = w_0 + v_t$. Additionally, λ_1 and λ_2 are model hyperparameters that determine our tendency of learning the tasks independently ($\frac{\lambda_2}{\lambda_1} \to \infty$) or together as a one size fits all solution ($\frac{\lambda_2}{\lambda_1} \to 0$)

RMTL's objective function is designed to learn the model parameters for the group model as the average of the parameters of all tasks. They show that by writing the Lagrangian of the objective function one can derive that $w_0^* = \frac{\lambda_1}{\lambda_2 + \lambda_1} \frac{1}{T} \sum_{t=1}^{T} w_t$ where w_0^* and w_t^* refer to the optimal solutions of the final model parameters. In other words, model parameters for the group model are learned as the average of model parameters for individual tasks.

They further prove that the objective function in 8 is in-fact enforcing a trade-off between regularizing model parameters w_t and their difference from the average of all model parameters. This can be shown by re-writing the regularization framework in 8 as shown below:

$$\begin{aligned} \min_{\varepsilon, w_0, v_1, v_2, \ldots, v_T} \quad & \sum_{t}^{T} \sum_{i}^{N_t} \varepsilon_{ti} + \rho_1 ||w_t||_2 + \rho_2 \sum_{t}^{T} ||w_t - \frac{1}{T} \sum_{s}^{T} w_s||_2 \\ s.t. \quad \forall t, i \quad & y_{ti}(w_t) x_{ti}^T \geq 1 - \varepsilon_{ti} \\ \forall t, i \quad & \varepsilon_{t,i} \geq 0, \end{aligned} \tag{9}$$

In Equation 9, the ρ_1 and ρ_2 are defined based on values of λ_1 and λ_2.

While existing methods outperform individually learned models in many problems, one common shortcoming of these methods is that they fail to prevent negative transfer when tasks are not similar enough. The next section will study the negative transfer phenomena in multi-task learning settings.

2.2.4.4 Negative Transfer in Multi-task Learning

As discussed in Section 2.2.3.3 negative transfer happens when transfer learning and multi-task learning result in machine learning models with reduced performance. This is often due to enforcing similarities that do not exist. When a large number of tasks are available, task relationships are often much more complex. For example, pairs of tasks might have different degrees of similarities. Alternatively, the task pool might be comprised of groups of more or less similar tasks. This motivates research and study of methods that incorporate task similarity weights, task groupings, and clusters or the hierarchical structure of tasks. While in this thesis, we study the last group, where task relationships form a hierarchical structure, in the rest of the section, we will briefly review the existing ideas for all three approaches.

Task Relationship Learning Approach: An underlying assumption of multi-task learning methods is the relatedness of target tasks. Some early work took advantage of existing task similarities to develop regularizers that guide the learning of tasks so that stronger transfer happens between more similar task pairs [54, 91]. However, such similarities are not always available a priori. This prompted research and development of MTL methods that can learn task relationships via task similarities, correlations, and so on. A Multi-task Gaussian process (GP) that directly captures task correlations by placing a GP prior over task functions f_t was proposed in [18]. Ben-David et al. developed a formal framework for task relatedness [15]. They tried to determine under what circumstances one can expect a group of tasks to be related in a way that helps improve learning and hence provided a formal definition of related tasks. Zhang and Yeung proposed a regularized multi-task relationship learning (MTRL) method that learned task relationships by placing a matrix variate normal prior $MN(0, I, \Omega)$ with zero mean on model parameter matrix of all tasks W in which I is an

identity matrix and represents the row covariance and Ω denotes to the column covariance. Equation 10 shows MTRL object function in which $||W||_F^2$ is regularizing model parameters for the T tasks's, $\lambda_2 tr(W\Omega W^T)$ is due to the GP prior and the loss function L is can be any classification loss.

$$\begin{aligned} \min_{w_1,w_2,...,w_T} \quad & \sum_{t}^{T}\sum_{i}^{N_t} L(y_{ti}, w_t x_{ti}) + \lambda_1 ||W||_F^2 + \lambda_2 tr(W\Omega W^T) \\ s.t. \quad & \Omega > 0, \qquad tr(\Omega) \leq 1 \end{aligned} \tag{10}$$

More recently extensions of MTRL method where proposed to handle multi-task boosting [232], multi-label classification [233] and sparse task relationships [230]. Multi-task k-nearest neighbors (kNN) method was also proposed to consider task similarities when using samples from related tasks that fit into the k closest neighbors of an unlabeled sample [229].

Task Clustering Approach: Another approach for preventing negative transfer is to consider the underlying group structures of related tasks. While hierarchical MTL methods are closely related to clustering techniques, in this section, we only review MTL methods that assume a flat task cluster structure. We review the existing hierarchical clustering techniques in Section 2.2.5.

Kang et al. proposed a clustered multi-task learning method that jointly learns target tasks' model parameters and G task clusters [89]. The idea is that similarities should only be imposed between similar tasks within the same cluster. Equation 11 shows the objective function.

$$\begin{aligned} \min_{w_1,...,w_T \;\; Q_1,...,Q_G} \quad & \sum_{t}^{T}\sum_{i}^{N_t} L(y_{ti}, w_t x_{ti}) + \gamma \sum_{g}^{G} ||WQ_g||^2 \\ s.t. \quad & \sum_{g=1}^{G} Q_g = I \\ & \forall g, t \qquad q_{g,t} \in \{0, 1\} \end{aligned} \tag{11}$$

where the first term $\sum_t^T L(y_{ti}, w_t x_{ti})$ is minimizing the classification loss for all tasks and the second term $\sum_g^G ||WQ_g||^2$ is enforcing similarities between tasks within cluster g using an l_2

norm regularization on columns of tasks' parameter matrix W. The matrix Q_g is a $T \times G$ diagonal matrix that indicates membership of task t in cluster g. Finally, the minimization constraint $\sum_g^G Q_g = I$ insures that each target task t only belongs to one and only one cluster.

Other methods proposed for cluster based multi-task learning include Xue et al. who deployed a Dirichlet process as the prior on model parameters to do clustering on the task level [210]. Han and Zhang devise a new regularization term that indirectly uses underlying task clusters by promoting similarity between pairs of tasks as shown in Equation 12 [236]. Their method minimizes within cluster distance of the model parameters during learning by creating a trade-off between minimizing the classification loss and minimizing the distance of the model parameters between task pairs. Therefore, the exact structure of clusters is not directly learned during the minimization algorithm. But, it can be obtained by comparing the task model parameters afterward.

$$\min_{w_1,\ldots,w_T} \quad \sum_{t}^{T} \sum_{i}^{N_t} L(y_{ti}, w_t x_{ti}) + \lambda \sum_{t>s}^{T} ||w_t - w_s||^2 \tag{12}$$

2.2.4.5 Multi-task Learning for Deep Neural Networks

In recent years, multiple hard parameter sharing (feature transfer) methods have been proposed by the research community [34]. Deep MTL frameworks typically permit knowledge transfer using either hard or soft parameter sharing of latent layers (Figure 2). Hard parameter sharing methods are closely related to the multi-task feature learning techniques[2, 3]. A shared latent feature layer is learned jointly for a set of closely related tasks that facilitates the training of improved machine learning models. In contrast, soft parameter sharing is comparable with the parameter transfer methodologies of learning task-specific latent feature layers and using regularization techniques to impose similarities between them (Constrained Layers) [52, 216, 217].

Early hard parameter sharing methods relied only on one or a set of global shared representation learning layers, which were used by task-specific prediction layers[34]. Others then extended this simple approach to facilitate learning both shared feature representation

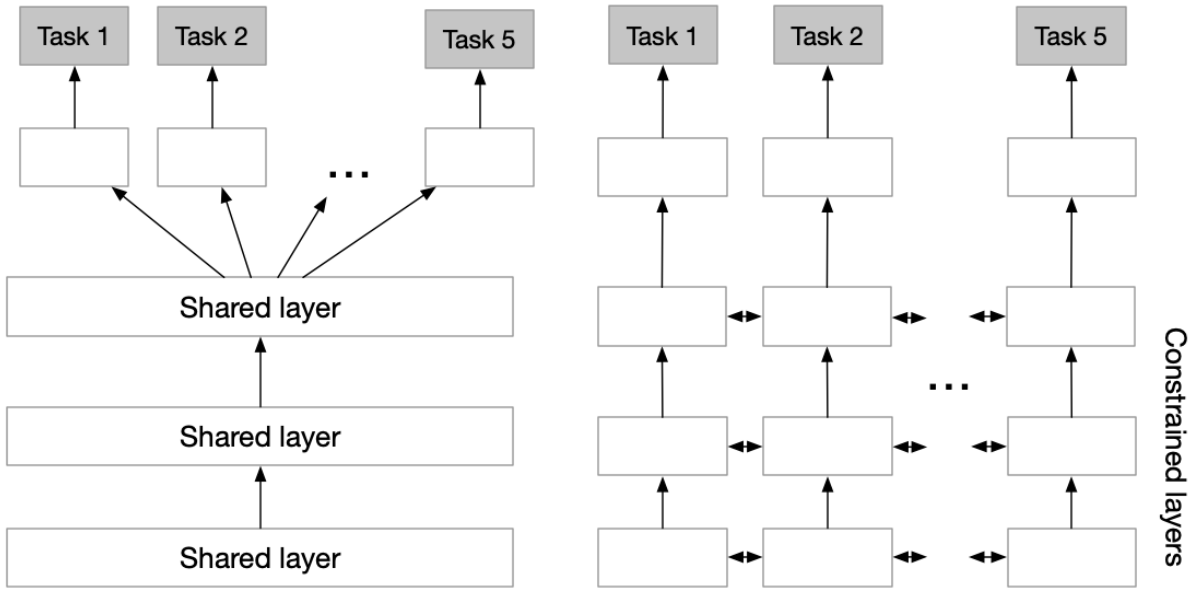


Figure 2: Comparison of hard parameter sharing (left) and soft parameter sharing (right) multi-task learning methods

shared across all tasks, and separate task-specific feature layers that were then used by a final set of independent prediction layers [234, 115]. For example, Zhao et al. introduce task-specific layers as linear projections of the share feature representation. The authors also propose a modulation framework that encourages shared feature representations between task-specific layers while disentangling the gradient directions, thus allowing optimized task-independent training. This has proven helpful in preventing negative transfer [234]. In another work, Liu et al. proposed a deep MTL learning method that used CNN layers to learn a joint feature representation layer for a set of target computer vision tasks[115]. However, to facilitate task-specific feature learning, independent attention layers were adopted to allow each target task to attend to the most important parts of the shared feature representation, assuming the task might require to use of different parts of the input to provide an accurate prediction. The task-specific attention layers allow each task model to pick up essential features from the shared CNN layers while downplaying the impact of less useful ones.

Others utilized multiple fully connected layers following a set of shared feature learning layers in AlexNet (CNN layers) to facilitate learning of task-specific features from the global feature representations (Figure 5) [129].

In soft parameter sharing techniques, sharing is designed to take place through regularization of task-specific layers also refereed to as constrained layers and includes a number of well known solutions including cross-stitch networks[97] and Sluice Networks[177]. Figure 3 shows the architecture of Cross-stitch networks on a multi-task application of AlexNet model[97] for image classification. The authors facilitated transfer of knowledge by using cross-stitch units after each feature learning block that allowed each target task to use a weighted linear combination of all task-specific feature resulting in information flow between target tasks. The weights of the linear combination are task-specific. Thus allowing each target task to choose how it wants to use the information of the related tasks. Later, [58] proposed a convolution based generalization of the cross-stitch networks(NDDR-CNN). The proposed method combined the output of each task-specific feature learning layer by first concatenating them and then combining them through a 1x1 convolution block (Figure 4). Following the approach in the cross-stitch networks, the 1x1 convolution blocks are task-specific allowing each target task to decide how features from relevant tasks should be combined. The convolution blocks can be learned in such a way to mimic the linear combination behavior of the cross-stitch networks, hence, offering a generalization of the earlier solution.

Sluice networks were also proposed in [177] as a generalization of the cross-stitch networks. In the Sluice network, each layer consists of both task-specific and shared feature learning components. Therefore, the model decides at each layer whether to prioritize using shared components to learn a joint representation or learn features for each target task independently.

Often multi-task learning architectures have utilized a combination of hard sharing and soft sharing approaches. Notable, Long et al. utilized a tensor network prior between task-specific fully connected layers to impose similarities. Furthermore, the tensor network priors acted as matrix priors on the fully connected layers, which allowed the model to learn the relationship between tasks, similar to some of the Relatedness-based MTL learning

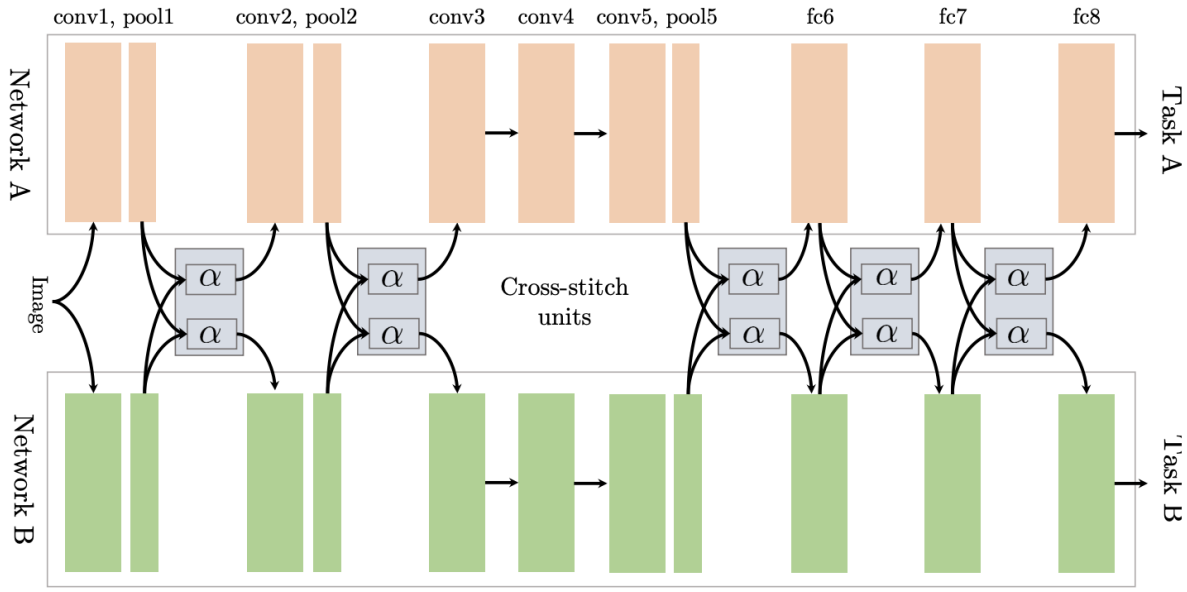


Figure 3: Cross-stitch network architecture uses a set of linear weights dedicated to each target task in order to learn a weighted linear combination of task-specific features at each fertilization block.

approaches we have looked at earlier in this section (Figure 5).

A critical shortcoming of early deep MTL methods is that they relied heavily on the relatedness of target tasks; hence negative transfer could happen when tasks are not sufficiently similar. Various methods have been proposed to prevent negative transfer that leverage underlying task clusters [85, 132], task-task relatedness [15, 89, 128], or facilitate an asymmetric transfer of knowledge [102, 103].

2.2.4.6 Remaining Shortcomings

Despite the advances in MTL methods and clustering techniques to prevent the negative transfer, a shortcoming of the existing work is that they have assumed that tasks reside in a flat cluster structure and do not consider their underlying hierarchy. In Section 2.2.5 we review some of the existing work in the field of hierarchical multi-task learning. However,

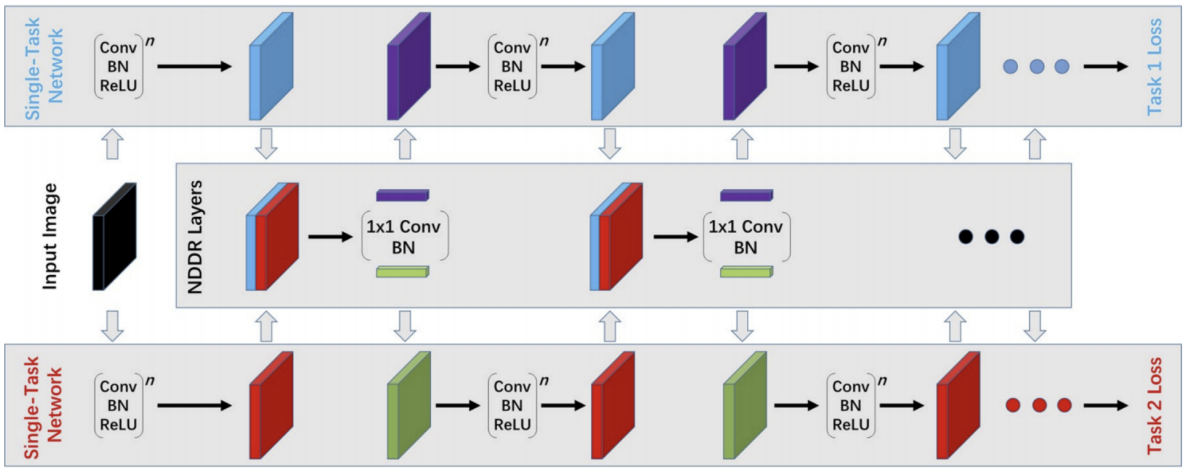


Figure 4: NDDR framework concatenates feature outputs of each CNN Batch Normalization ReLU block and combines them through a 1x1 convolution module

the area has yet to benefit from methods that address all challenges mentioned in 1.3.

2.2.5 Hierarchical Multi-task Learning

Recent work on hierarchical multi-task learning has attempted to take advantage of hierarchies by imposing regularization on tasks based on groups formed at different levels of the hierarchy and by assuming that the target tasks only reside in the leaf nodes of the hierarchy [113, 94]. For instance, Kim and Xing developed a regularized regression algorithm called tree-guided group lasso by assuming that the tree structure is available a-priori. The idea behind the tree-guided regularization is to recursively apply the group lasso regularization to enforce similarities amid tasks' parameters that belong to the same internal node. Assuming we have V nodes from which T tasks are linked to the T leaf nodes of the hierarchy, s_v and g_v at a specific node reflect the trade-off between preferring higher similarities between tasks at that node or at its child nodes. Equation 13 shows the

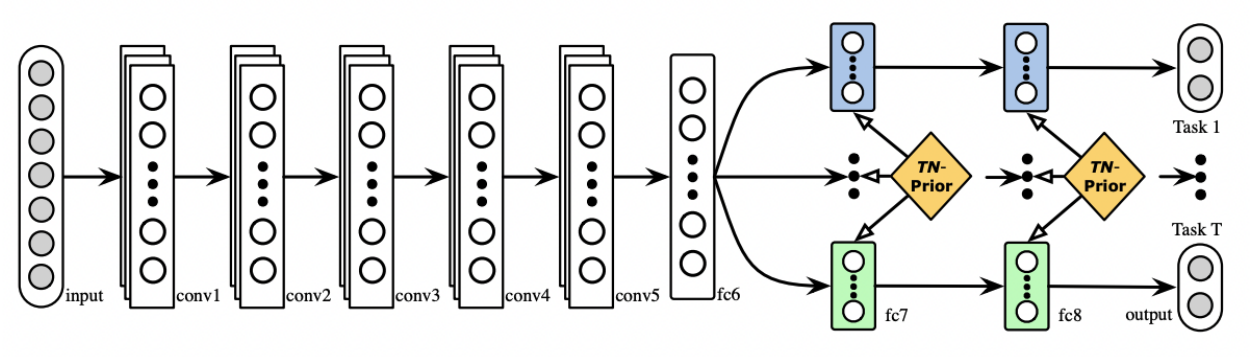


Figure 5: The proposed MTL solution by Long et al. combines hard and soft parameter sharing. It adopts the first 5 CNN-based feature learning layers from AlexNet to facilitate hard parameter sharing through a set of global feature learning layers. Next, multiple task-specific, fully connected layers are designed to learn independent features for each target task. Finally, the model allows soft-parameter sharing between the task-specific fully connected layers through a matrix prior between target tasks designed to learn the task-task relatedness and an MTL regularization term in the loss function.

minimization objective for their method.

$$\min_{w_1,\ldots,w_T \;\; s_1,\ldots,s_V \;\; g_1,\ldots,g_V} \quad \sum_{t}^{T} \sum_{i}^{N_t} L(y_{ti}, w_t x_{ti}) + \lambda \Gamma(v) \tag{13}$$

in which $\Gamma(v)$ defines the regularization terms at node v and is described below:

$$\Gamma(v) = \begin{cases} s_v \sum_{c \in children(v)} |\Gamma(c)| + g_v ||W_v||_2 & \text{if } v \text{ is an internal node} \\ ||w_v||_2 & \text{if } v \text{ is an leaf node} \end{cases} \tag{14}$$

where W^v corresponds to the matrix of model parameters that are under node v. The recursive regularization algorithm in Equation 14 allows the model to find and impose similarities at each branch of the tree. However, one shortcoming of this method is that it does not allow the regularization of model parameters at both the group level and individual tasks equally since they assume $s_v + g_v = 1$. Additionally, the regularization terms at the lower levels of

the tree will inevitably have a lower impact in practice due to recursive multiplications of g_v parameters.

An alternative group of hierarchical multi-task learning methods approaches this problem by applying multi-level regularization techniques. The general assumption behind this approach is that the hierarchical structure of the task is not available. However, by using multi-level regularization techniques, we allow tasks to have different degrees of similarities at different levels, hoping each layer will learn model parameters that are shared at that level [39, 237, 68].

The shortcomings of current hierarchical multi-task learning methods are three fold. First, the research community has yet to propose solutions that support transfer of knowledge between tasks with heterogeneous relationship types. Second, in order to prevent negative transfer it seems vital to propose methods that consider both task hierarchies and relationships. For instance, we think that task relations may need to be asymmetric for the effective use of hierarchy. The Transfer of knowledge between siblings may need to be asymmetric to prevent the negative transfer from a task with a weaker classification model to its siblings so that a stronger model could be independently trained for them. Finally, we need methods that can handle imperfect hierarchies. This happens because of outlier tasks, groups that are too general, and residual categories, as discussed in Section 1.3.

2.2.5.1 Hierarchical Multi-task Learning for Deep Neural Networks

The existing research incorporating hierarchical task structure in deep multi-task learning methods has remained relatively limited to a handful of interesting approaches introduced in recent years. HD-MTL proposed to replace the traditional softmax layer with a tree-based classification layer that incorporated the hierarchical structure of the target task embedded in a visual tree [56]. The proposed architecture used: (1) A set of common CNN layers to learn shared feature representations followed by (2) multiple sets of separate task-specific CNN layers followed with fully connected layers to learn disjoint sets of feature representations and finally, a tree-based classification layer that would allow each target task model to choose to use features from each of the disjoint CNN-based feature

learning blocks (Figure 6). The tree-based classification layer was designed to enforce a tree-based constraint which would guarantee that the classification of a sample to a target class would result in also assigning it to the parent classes according to the visual tree.

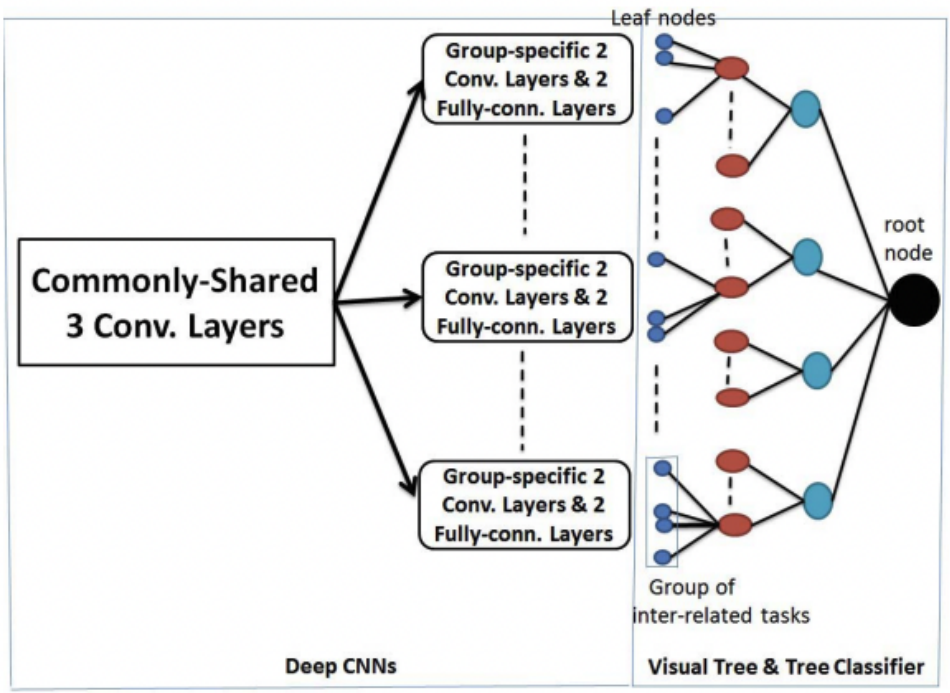


Figure 6: The model architecture for HD-MTL method

In another work, Sanh et al. proposed a top-down hierarchical multi-task learning approach to jointly train a set of carefully selected NLP tasks, including named entity recognition, relation extraction, and entity mention detection [179]. While an intrinsic hierarchical relationship did not exist between the proposed target tasks, the proposed model achieved promising improvements in all target tasks by introducing a hierarchical inductive bias between the tasks by learning low-level tasks (that are assumed to require less knowledge and language understanding) at the bottom layers and learning higher-level tasks at higher layers. In other words, the hierarchical model would facilitate feature transfer in a top-down approach allowing the lower-level tasks to use the feature representations learned for top-level

target tasks.

2.3 Learning Patient Representations from Electronic Health Records

Patient's electronic health records (EHRs) are an integral part of today's clinical workflows. A patient's EHR data are formed by complex multivariate temporal sequences of events that cover a wide range of patient related clinical information, including demographics, medical history, vital signs, physiological measures, medication administration, and laboratory results. The complexity and temporal character of the EHR make the problem of learning accurate machine learning models directly from such data very challenging. Therefore, it is often desirable to develop feature representations of the patient and the patient state that are smaller and more compact, and that are at the same time capable of summarizing the information in EHR important for building accurate machine models.

The different solutions one may use for building patient representations from patients' EHR can be divided into three main groups: unsupervised, supervised, and hybrid representation learning techniques. The unsupervised methods aim to learn lower-dimensional representations of patient data that are able to recover (reconstruct) in some way the original data and their key characteristics. The methods are usually trained by minimizing the reconstruction error $L(X, X')$ which can be modeled using an euclidean norm such as the l2-norm (Frobenius distance) between the original input X and the reconstructed matrix X' from the low-dimensional representation.

While unsupervised approaches offer a patient representation that can be often adopted to solve a wide range of tasks, they may not necessarily offer the best performance on the specific task. Supervised techniques aim to learn a patient representation that is optimized for a specific target task or a set of target tasks. The representations are usually trained in a supervised fashion by minimizing the suitable predictive loss for the target machine learning problem. Finally, hybrid approaches combine and leverage both unsupervised and supervised representations in different ways.

In the following we review in greater depth methods and solutions that have been devel-

oped and applied to represent patient's EHR data in order to facilitate the model learning process.

2.3.1 Template-Based Feature Representation

Early work in this area extracted the patient representations from the electronic health records by converting patients' clinical time-series data to a vector space representation of a patient's state by defining and extracting a set of features for each time-series in the EHR and by merging them into one vector. [72, 203]. The extracted feature vectors were then used to learn machine learning models for the different target tasks.

In general, template maps featurizing the different time series in EHRs can be of different complexity. In the simplest case the features can be formed by simple indicators of occurrence of clinical events of different types in the EHR. More complex solutions, such as maps used in the work of Hauskrecht et al [72, 71, 70] may cover a broad variety of temporal and non-temporal statistics characterizing the time series such as last value, recent trend, and slope, apex, nadir values, etc.

While the template-based method are able to replace complex multivariate time series with a vector representation of the patient state suitable for a variety of ML methods it also introduces new challenges. One of the challenges is a high dimensionality of the feature vector generated by the template-based method and our ability to learn high-quality models for the target task from such a data. To overcome this problem, feature selection techniques or expert supervision to choose features or relevant feature blocks were needed. The feature selection methods in combination of with template-based data were successfully applied for predicting patient medication and lab orders in work of [71, 70]

Whilst template-based feature methods rely on predefined feature maps, another group of methods attempted to learn the temporal features representing the patient's EHR data automatically using pattern mining techniques [11, 13, 14, 10, 9]. Pattern mining methods can solve the high-dimensionality problem by choosing the top most informative patterns to create the patient state representations. Therefore, the performance of the target task machine learning models will rely on the accuracy of the interestingness measures used to

filter the most expressive patterns. The significance of this issue has motivated numerous past studies by the pattern mining community [59]. For instance, Batal et al. proposed a Bayesian scoring-based framework that relied on Bayesian inference to evaluate the quality and structure of the rules to filter out spurious ones[8].

2.3.2 Matrix Decomposition Methods

EHR data cover a large number of time series and many of these may be dependent. These relations often translate to and exist also in high-dimensional vector-based patient representations based on feature templates. This allows us to adopt unsupervised matrix-decomposition techniques to learn more compact lower-dimensional representations of patients' vectors generated from electronic health records. The most common approach for reducing the dimensionality of complex feature vectors is singular value decomposition (SVD). The algorithm learns orthogonal eigenvectors representing non-overlapping underlying patient conditions and information. The method has been successfully applied in multiple EHR data analysis works [144, 137, 221, 7]. We note that the SVD approach is closely aligned with our data analysis method in Chapter 3. Other matrix decomposition methods, such as non-negative matrix factorization, can be applied to EHR data. For example, Ho et al. proposed Marble which adopted a sparse non-negative matrix factorization-based approach to directly learn lower-dimensional representations from patients' EHR data [75].

2.3.3 Sequential time-series models

In general, EHRs are defined by complex multivariate time series of events associated with patient condition and patient managements. One way to represent the patient state and its feature vector at a specific time is by maximizing its ability to predict future events. In such a case, the state is referred to as a Markov state of the process. Briefly, the *Markov property* assumes that the current state captures all necessary information relating the future and past. Multiple models and methods may be used for this purpose. For example, one may attempt to model the time series and their states using point processes [100, 84, 170, 101]. The point process models have been applied to various event sequence problems including clinical event

prediction [117, 142, 118]. However, these models are very hard to optimize directly when covering many events, and because of this, the event time series are often converted to discrete-time models segmented using a window spanning a fixed period of time, such that events within the window are considered to co-occur in the discretized time. A broad range of statistical models covering discrete time models capable of predicting future events exist. The most common ones include Markov Models defined by probabilistic transitions among discrete sets of states, and Hidden Markov Models that can model discrete-time time series using discrete hidden state representation [166]. When clinical time series record real-values, autoregressive models [67], linear dynamical systems [88, 61, 123, 125, 124, 126] or Gaussian processes [169, 127, 122] are often the models of the choice.

2.3.4 Autoencoder networks

In recent years, advances in deep neural networks redefined the landscape of machine learning solutions one can apply to represent patient's EHR. One such example are autoencoder networks that effectively replace linear methods for defining low-dimensional representations based on SVD with non-linear models. Auto-encoder (see Figure 7) is a deep neural network architecture designed to learn a low-dimensional representation of the original input with the help of the middle restricted neural network layer. The auto-encoder model is usually trained by minimizing the reconstruction loss, aiming to learn a general-purpose representation that best retains the vital information in the EHR data.

One of the early approaches that adopted auto-encoder networks was Deep Patient [144]. The authors used a denoising auto-encoder model to learn unsupervised patient representations over time segments of patient clinical data that could be adopted by various machine learning problems[144]. Later, Katsuki et al. extended this work by proposing a convolution-based auto-encoder architecture. The proposed method was developed to overcome the challenges related to irregular sampling times of EHR data and to capture time-shift invariant correlations between clinical measurements [92].

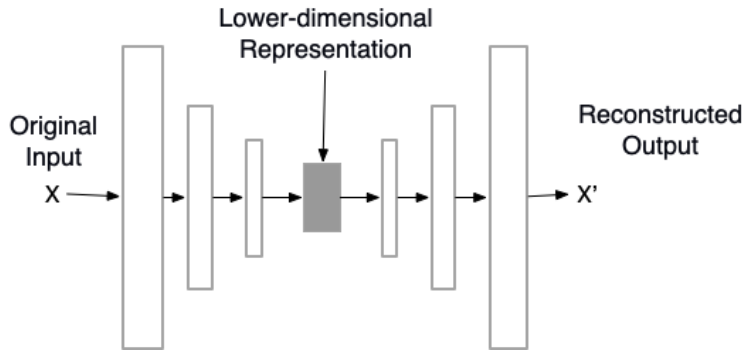


Figure 7: The general architecture for auto-encoder networks

2.3.5 Recurrent Neural Networks

Similarly to autoencoders, sequential models based on recurrent neural networks (RNN) and their clones have been introduced to alleviate the shortcommings of statistical time series models. Early works by Lipton et al. [112] and Rajkomar et al [168] used RNN-based architecture with Long Short Term Memory (LSTM) units to predict patient discharge diagnoses as a multi-label classification problem from underlying clinical variables. On the other hand, in Doctor AI [29], the authors adopted a somewhat simpler GRU-based model to encode the sequential patient medical data from past hospitalization into a lower-dimensional representation and could be used to predict patients' future visit diagnoses. In a more recent work, [104] argued that while RNN-based models are able to learn feature representations that combine useful information from past and recent timestamps, accurate prediction of clinical events depend on the model's capability to combine these representations with patient's current clinical context. Thus they proposed a context-aware LSTM-based method that learned to combine the temporal feature representations from LSTM network with a linear embedding of the recent patient state (context) and showed the new context-aware features can help learn improved machine learning models. Additional refinements of this work added temporal mechanism for representing and modeling periodic events and their frequencies [105, 108].

In Section 3.6.2, we will propose a similar solution to the methods discussed here that

use LSTM networks to capture temporal feature representations of patient's EHR that can be used for classification of patient's diagnoses.

2.3.6 Attention Based Methods

While RNN-based methods were able to capture temporal patterns in patients' EHR data important for the target task, they also introduced a new set of challenges. First, when applied to long sequences, RNN models can suffer from the vanishing gradient problem [76]. While the LSTM networks were designed to solve this problem using activation gates that allow the model to choose between past and new information, past research has shown that they can perform poorly when facing extremely long sequences. Second, RNN models can require significantly longer training times since they are applied to timestamps sequentially, thus preventing parallel processing in Graphical Processing Units (GPU).

In order to address the above challenges, new solutions adopted an attention mechanism. The attention mechanism is a neural network module that mimics human cognitive attention and is designed to allow the network to allocate more focus on parts of the data that are more important. For example, in the context of temporal clinical data, attention is mainly used to allow the model to focus on important timestamps during a patient's hospitalization. Learning to identify which times during the hospitalization are more critical than the others depends on the context. Thus, attention modules are trained as part of the original network and learn to assign a set of "soft attention weights" to each segment of the input features. Finally, attention output is obtained as the weighted average of the input segments.

In recent years, various methods have been proposed that leverage attention mechanisms to facilitate learning of more expressive feature representations of patients' EHR data. These approaches can be classified into three primary categories:

The first group uses an attention mechanism to allow the deep neural models to focus on the critical segments of patients' data to help learn better feature representations. For instance, the Reverse Time Attention Model (RETAIN) proposed to use two pairs of RNN networks and attention mechanisms to learn two sets of weights, one for time (α_i) and another for feature variables at each timestamp (β_i)[31]. The attention weights α_i and β_i would

be trained based on the hidden states of the respective RNN networks. Finally, the patient representation c_i will be calculated as $c_i = \Sigma_j = i^1 \alpha_j \beta_j v_j$ in which v_j represents the EHR embedding of data at time j (See Figure 8).

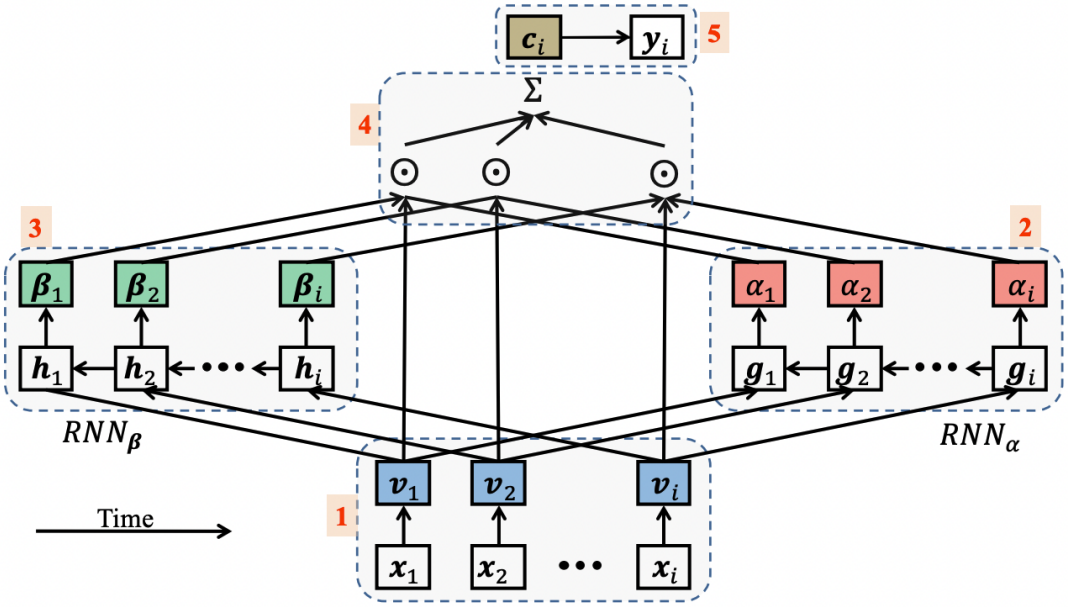


Figure 8: The general architecture for the RETAIN model. In Step 1, a fully connected layer is adopted to learn a dense embedding of the patient's input state at each timestamp. Step 2 uses an RNN network followed by a soft attention module to learn the time attention weights α. Similarly, Step 3 uses a separate RNN and attention pair to learn variable attention weights β at each timestamp. Finally, the final feature representations are calculated in Step 5 as the weighted combination of features according to the feature-level and time-level attention weights.

Lee et al. adopted an attention-based extension of the context-aware LSTM models to learn a multi-scale feature representation from patients' EHR data. The proposed method uses an attention mechanism to allow the model to attend to and combine important patient information from "distant past", "intermediate past" and "recent past" that facilitate learning of improved machine learning models for the prediction of clinical events. Patient2Vec

learns a feature representation of patients' EHR data across multiple visits using multiple levels of attention mechanisms to obtain a multi-scale combination of patient information within multiple hospital visits [64]. Patient2Vec uses the word2vec method originally introduced for natural language processing problems by treating patients' EHR data within a visit as a sentence and clinical events as the words within that sentence to create a dense embedding of the patient's EHR data(Figure 9). Next, it defines subsequences of patient hospital visits by grouping subsequent patient visits within a predefined time window. Since the information within each visit may not be equally important, the authors adopted a within-subsequence self-attention mechanism to obtain a feature representation for each subsequence as the weighted average for each patient visit's embedding. Finally, an RNN network followed by an attention module was used to combine the subsequence embedding and learn a final feature presentation that embeds a patient's EHR throughout multiple hospital visits and captures critical information to predict future diagnoses.

The second group takes advantage of attention modules to learn enriched patient feature representations by modeling the relationships between patient's clinical information. For instance, GRAM supplements the raw EHR data with the hierarchical information in medical ontologies to learn better embedding of clinical concepts [30]. GRAM achieves this goal by representing a low-level medical concept (leaf) as a combination of the embeddings of its ancestors in the ontology via an attention mechanism. Hence, when a medical concept is less frequent in the data, more weight will be given to its ancestors as they can be learned more accurately (Figure 10. GRAM, adopted the global vectors for word representations (GLOVE) to learn the initial embeddings for clinical concepts based on the global co-occurrence matrix of words [163].

Similar to GRAM, Choi et al. later proposed a deep neural network model that learned multi-level embeddings of EHR data (MiME) [32]. MiME defined the final feature representation for a patient visit by combining the embeddings of the associated diagnoses codes to that visit. Consequently, the clinical embeddings for the diagnoses code embeddings were defined according to the embeddings of their associated treatments (medications and procedures). MiME achieved this goal by learning the multi-level associations between clinical concepts as auxiliary tasks while simultaneously using the EHR representation of the

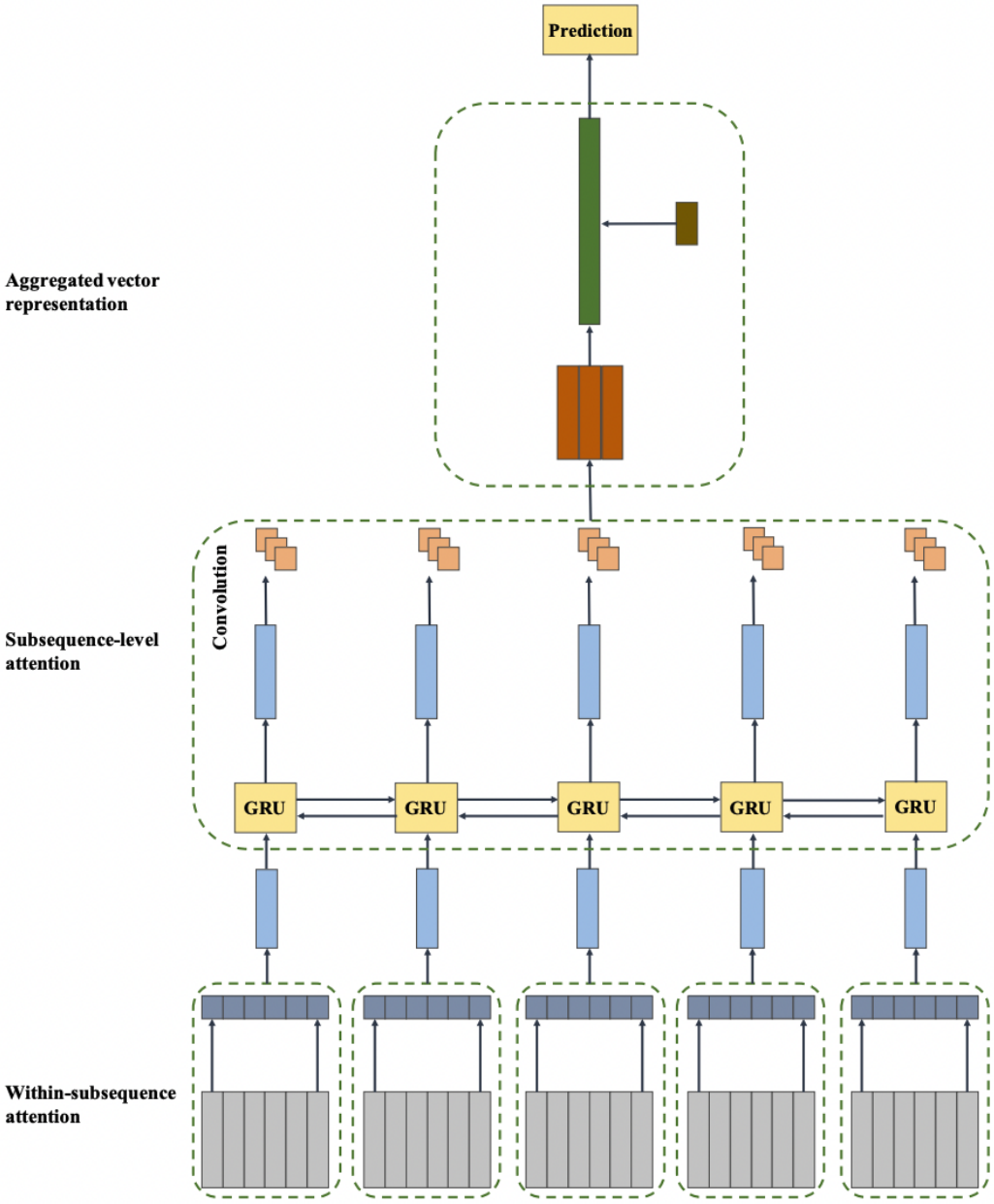


Figure 9: Patient2Vec learns a feature representation of patients' EHR data across multiple visits using multiple levels of attention mechanisms to obtain a multi-scale combination of patient information within multiple hospital visits.

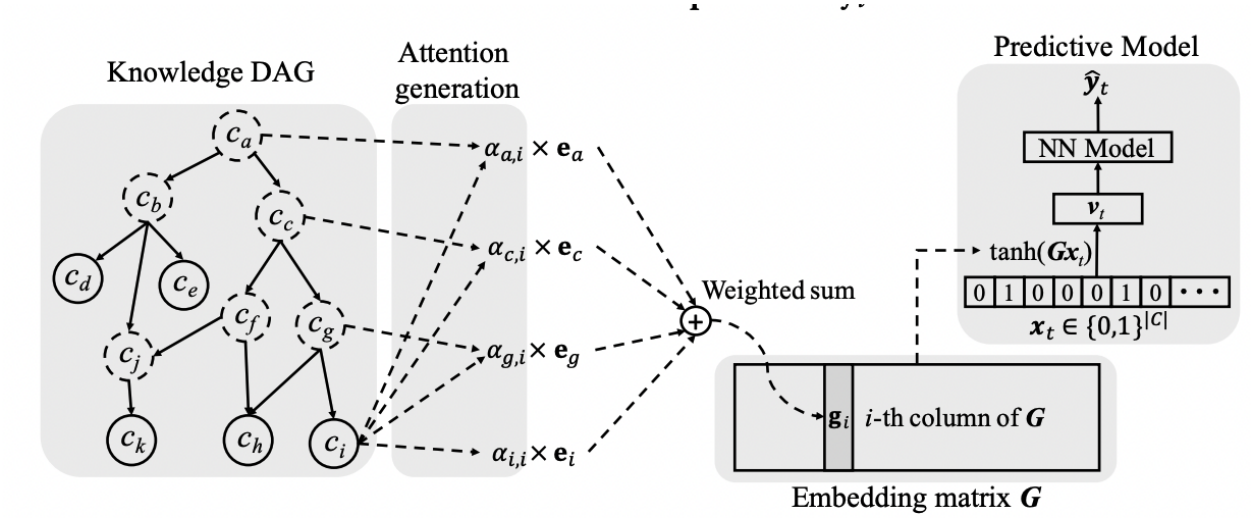


Figure 10: GRAM method uses attention to define the final embedding of the leaf clinical events as a weighted combination of its ancestors. This facilitates learning a better representation from the patients' EHR data, especially by defining events with low occurrences based on their more common parents.

patient's visit to learn predictive models for a set of downstream tasks.

The third group of attention-based methods aimed to tackle the slow training problem associated with the RNN-based models. This was mainly motivated by a well-known paper, "All you need is attention" that proposes a new simple network architecture, the transformer, based solely on a set of parallel multi-head attention mechanisms, dispensing with recurrence and convolutions entirely [205]. Their experiments on two machine translation tasks show the transformer model to be superior in quality while being more parallelizable and requiring significantly less time to train. Later Devlin et al. proposed Bidirectional Encoder Representations from Transformers method (BERT) as a multi-layer bidirectional transformer encoder architecture that learns a deep bidirectional transformer-based embedding from natural language text. The authors pre-trained the model based on a set of well-defined auxiliary tasks [44] and demonstrated that it could be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks without

substantial task-specific architecture modifications to solve arbitrary target tasks.

The promising results achieved by the BERT architecture motivated numerous efforts that explored ideas that would adopt BERT models to learn transferable representations from electronic health records. Most notable, [110] proposed BEHRT, designed to pre-train deep bidirectional representations of medical concepts by jointly conditioning on both left and right contexts in all layers. The pre-trained representations can be simply employed for a wide range of downstream tasks, e.g., predicting the next diseases and disease phenomapping.

3.0 Modeling Patient Diagnoses using Electronic Health Records

The widespread adoption of electronic health records (EHRs) has introduced the opportunity to process and extract valuable knowledge from massive data warehouses of real-time and diverse clinical data recorded during patient's hospitalizations. One interesting problem is the automatic assignment of diagnostic codes to patients' hospital stays.

3.1 Problem Significance

The problem of automatic assignment of diagnostic codes to patients' hospital stays is an interesting application of hierarchical multi-task learning. If the problem is solved successfully, it can help improve several hospital workflows related to both clinical decision-making and administration of healthcare systems. First, these codes are commonly used for hospital reimbursement, usually assigned to patients by a human annotator (a trained nosologist) after discharge. An effective solution can help speed up the annotation process and alleviate costs. Second, an automated diagnostic system could help physicians by providing a concise, automated, and easily accessible summary of patients' conditions and problems at the time of discharge and during the patient's hospital stay. Hence, it can act as a decision support tool that can recommend and bring to the attention of physicians possible patient diagnoses that have not yet been considered. Given the importance of these applications, recent years have witnessed an increased interest in developing machine learning methods that can automatically assign diagnoses to patient stays based on the information in their electronic health records(EHR) [144, 168, 137]. However, despite recent advancements, multiple challenges making the solutions more practical remain to be solved.

First, learning models from structured EHR data to automatically classify the diagnoses the patient suffered from during the hospital stay is not trivial. Structured EHRs consist of a large number of time series that represent a variety of labs, physiological measurements, symptoms, treatments, procedures, etc. Hence it is not easy to automatically associate

the signals in these time series with specific diagnoses, especially when the diagnoses can be defined by multiple alternative combinations of these signals. This is because multiple clinical data in a patient's electronic health records may often indicate the presence of a particular underlying patient condition. This problem is even more challenging when data is sparse and many time series for the patient cases are unknown are missing. Therefore, learning accurate diagnostic models will depend on learning useful feature representations from patients' electronic health records that can capture and summarize the crucial underlying patient conditions and signals. This problem is closely aligned with the first research goal in this thesis. Therefore, in this chapter, we study and propose a new approach for learning simple yet flexible lower-dimension representations of patient EHR data that can be used to learn machine learning models for a wide range problems including automatic assignment of patient diagnoses.

3.2 Contributions and Outline

The main objective of this chapter is to study the first research goal in this thesis which is to propose a simple yet flexible framework to learn dense feature representations from patients' electronic health records and evaluate them in the context of learning diagnostic models. Therefore, in the rest of this chapter, we first review existing standard patient disease hierarchies that help define the target diagnoses and diagnostic categories and their relationships. Next, we provide a brief overview of related work to classify and predict patients' diseases and other attempts to use patient disease hierarchies. Next, in Section 3.5, we propose both unsupervised and supervised techniques for learning feature representations from patients' electronic health records. Briefly, the proposed unsupervised method uses an eigendecomposition technique based on singular value decomposition to learn a dense, orthogonal, and lower-dimensional representation of patients' EHR summarizations. On the other hand, the supervised method uses deep neural network architecture based on the Recurrent Neural Network (RNN) with lower-dimension representation sufficient to support task predictions.

Throughout this chapter, we evaluate the proposed patient representations and provide extensive results in the context of assigning patients' diagnoses. As such, in Section 3.6, we model patient discharge diagnoses using standard machine learning techniques. First, we use support vector machines [74] and learn machine learning models for each target diagnosis independently using the unsupervised EHR representations to assign diseases to the entire patient visit automatically. Later, in Section 3.6.2, we propose deep recurrent neural network (RNN) [223] architectures that leverage the supervised feature representation learning method described in Section 3.5. The proposed deep RNN solutions extended the earlier work by also capturing the temporal patterns in patients' clinical data and formulate the diagnostic assignment problem as a sequence classification problem.

Finally, in Section 3.8 we summarize the contributions and conclusions in this section and discuss motivational findings for research into new hierarchical multi-task learning methods that can help better leverage task-task relationships among patient diseases to learn improved classification models.

3.3 Standard Patient Diseases Hierarchies

Diseases and conditions that patients suffer from can be categorized according to various factors, including etiological (causal), pathological (by the nature of the disease process), epidemiological (distribution and control), and physiological or symptom(s). Another way of classifying diseases is based on affected organs, called topographical classification. However, this can become complicated since many diseases impact multiple organ systems in our bodies. Today, many classifications of diseases have been developed and are actively maintained [188, 158, 47, 111]. However, the most popular and widely used one is the World Health Organization's (WHO) International Classification of Diseases (ICD) [188, 158]. The ICD hierarchy was created and published worldwide based on mortality statistics and used for public health and epidemiological research. Therefore, due to its statistical nature, it is a great candidate for use in machine learning research. Figure 11 visualizes a small subset of the ICD-9 disease hierarchy.

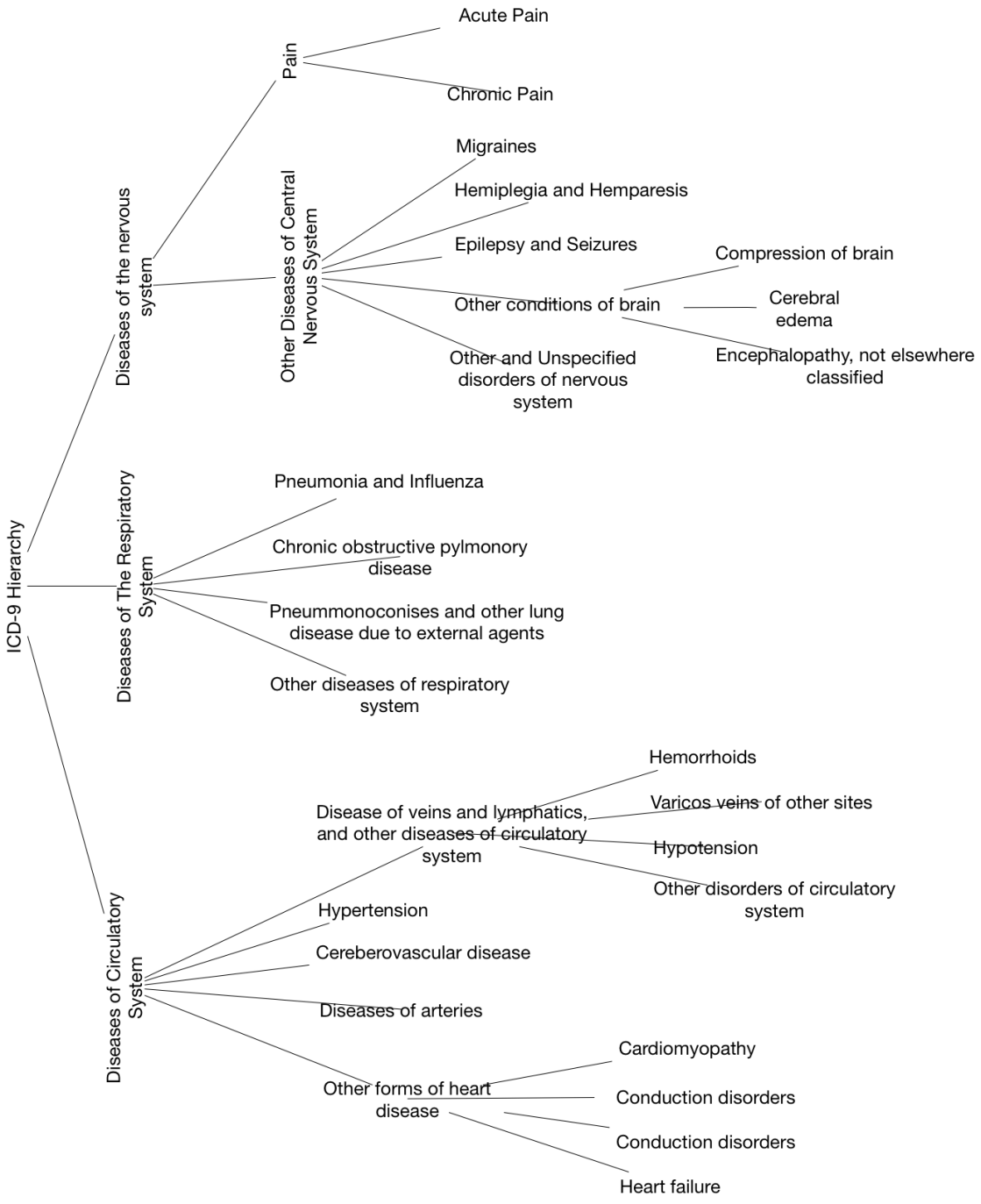


Figure 11: A subset of ICD-9 disease hierarchy

In the United States, ICD-10, the most recent version of ICD codes, is adopted by the Centers for Medicare and Medicaid Services (CMS) for medical coding and reporting.

However, its earlier version (ICD-9) is also often used in electronic health record systems, and open medical datasets available for research [87]. Both the ICD-10 (70,000 codes) hierarchy and ICD-9 (17,000 codes) include diagnoses code for a wide range of patient diseases and conditions designed to capture many common and even unlikely circumstances. While many of these diagnosis codes may never be used, the prevalent diagnostic codes can still represent a large-scale hierarchical multi-task problem. However, in this chapter, we mainly focus on exploring ideas for learning expressive feature representation of patient's EHR that can be used for learning accurate diagnostic models and leave the study of hierarchical multi-task learning for this problem to chapters 4 and 5.

3.4 Related Work

Patients' diagnoses and diagnostic codes in disease hierarchies have been the subject of various research in recent years:

The first group of existing works studied the problem of automatic assignment of ICD diagnostic codes. However, most of the existing research in this space trained diagnostic models using patients' clinical notes and limited the target diagnostic tasks to only the leaf nodes from the hierarchy since patient diagnoses are normally only assigned by the hospitals using the lower level ICD codes and not the internal categories. For example, Pakhomov et al. proposed a technique based on learning lower-dimensional embedding from clinical notes using autoencoder networks to tackle this problem [159]. More recently, Deep Patient adopted a similar denoising auto-encoder architecture to learn a general-purpose unsupervised encoding of patients' EHR trained on 700,000 patients' EHR. The learned patient representation was then used to train the classification model for patient diagnoses [144].

Separately, a number of existing research attempts to predict patients' diagnoses for future hospital visits using clinical data from previous visits. For example, Lipton et al. proposed a Recurrent Neural Network (RNN) architecture based on Long Short Term Memory (LSTM) units to predict future patient visit diagnosis from a collection of 13 clinical

variables [112]. In addition, Choi et al. proposed to RETAIN [31], a reverse time RNN network with both visit and variable level attention to learn from the patient's medication, procedures, and problems.

More recently, researchers have also attempted to leverage the hierarchical disease classifications to learn enriched feature representations of patients' clinical data. Most notably, Singh et al. proposed and evaluated different ways of embedding the hierarchical structure of diagnosis in their feature vector [187]. Similarly, GRAM [30], was presented as an attention-based network that uses the disease hierarchies to learn more expressive BoW representation of patients' clinical data.

Our work in this dissertation differs from the current work in multiple ways. First, we aim to learn diagnostic models from patients' structured data in contrast to the existing work that used a lower-dimensional representation of patients' clinical notes. To do so, we develop techniques that adopt similar ideas to those used for natural language processing problems and show that they can be used to learn expressive feature representation of patients' structured EHR data. Second, most existing work in learning diagnostic models either focuses on lower-level diagnoses codes or uses the diagnoses hierarchy to learn more expressive feature representations from patients' clinical data. On the contrary, in this thesis, we define diagnostic tasks as both leaf and categorical nodes. We believe that although diagnosis categories are not directly used for medical billing purposes, accurate classification of such categorical codes can be informative for both clinical and billing applications. Additionally, instead of using the task hierarchy on the feature learning aspect, we aim to leverage the hierarchical task structures to facilitate the transfer of knowledge between target tasks and thus learn improved classification models. We will defer the study of the last problem to chapters 4 and 5.

3.5 Learning Feature Representations from Electronic Health Records

In this section, we propose a general framework for learning an expressive representation of patients' electronic health records that can be adopted for various applications of hier-

archical multi-task learning in healthcare. Briefly, the proposed method is designed to (1) extract an event representation of a patient's EHR data and (2) leverage similar techniques adopted in natural language processing methods by treating such events analogous to words and the entire or segments of patient hospitalization as documents to obtain bag-of-word summarization of patient's clinical data, and (3) to learn dense lower-dimensional representations of the bag-of-word data that capture underlying patient conditions and be suitable for modeling various machine learning tasks.

3.5.1 Basic notation

Let V_i denote a patient visit i and let $D = \{V_1, V_2, ..., V_{|D|}\}$ be a set of all patient visits in our data. A visit can be defined as $V_i = \{X_i, Y_i\}$ where X_i and Y_i are respectively a set of clinical data recorded and target task labels assigned to the patient during the hospitalization. A time-series segmentation algorithm with regular samples divides V_i into a sequence of discrete segments with regular intervals, hence, visit i can be defined as $V_i = \{X_i, Y_i\}$, while $X_i = \{x_i^1, x_i^2, ..., x_i^{l_i}\}$, $Y_i = \{y_i^1, y_i^2, ..., y_i^{l_i}\}$ and l_i refers to the number of segments created during patient i's hospitalization. Our goal is to learn a function f_{embd} that can learn expressive feature representation from patient's clinical data X_i or from the segmented inputs $\{x_i^1, x_i^2, ..., x_i^{l_i}\}$. In the rest of this work we use X^i and Y^i to refer to the entire set of patient's clinical records and target task labels for visit i, while, x_t^i and y_t^i will correspond to clinical data records and labels associated with the patient during a particular time segment t of patient visit i for simplicity.

3.5.2 Binary Bag-of-Word Representation

Physicians often summarize patients' conditions and overall clinical pathways by providing a general overview of what has happened during the hospitalization. For example, an ICU patient may have multiple low blood pressure recordings (both Systolic and Mean blood pressure) followed by multiple administrations of vasopressors, indicating that the patient likely experienced hypotension shock during their ICU stay. Such summarizations might abstract away details of dosage or measurement values in favor of informative discretizations

based on appropriate critical ranges such as systolic blood pressure (SBP) lower than 90 represented as low SBP. Motivated by this example, we propose a preprocessing method $f_\beta(x) : \mathbb{R} \to \{0, 1\}$ that generates a binary clinical event representation E_i from patients' clinical data X_i to summarize patient information from a diverse and wide range of clinical variables. As shown in Figure 12 these clinical events are created by utilizing either standard range thresholds or indicator functions that determine whether the clinical variable (i.e. administration of a particular medication) had taken place during the patient's stay. Despite its simplicity, the proposed method provides a flexible framework that can easily be extended to new data types and variables. When time segmentation is appropriate, such segments can also be created from binary events in similar manners as it would have been done for the raw patient's clinical data. In this work we will use $x_i^t \in \{x_i^1, x_i^2, ..., x_i^{l_i}\}$ to refer to patient i's raw EHR data at time segment t and $e_i^t \in \{e_i^1, e_i^2, ..., e_i^{l_i}\}$ to represent its corresponding binary event representation.

Finally, we create a normalized bag-of-word (BoW) representation of a patient's clinical events to summarize the entire hospitalization or a specific time segment during the visit. A bag-of-words is a vector representation of patient EHR data that describes the occurrence of words within a document. Figure 12 summarizes the pre-processing steps for creating such BoW representations for different EHR time segments. The intuition behind using the BoW representation is very simple. Patients with similar diagnoses will also experience similar symptoms, have similar lab results, and receive comparable care plans throughout their hospitalization.

The advantage of using the bag-of-word representation is that it is straightforward to understand and implement and offers much flexibility for customization for various types of clinical data. However, it also continues to face some of the challenges in EHR data that we discussed earlier. First, the bag-of-word representations inherit the sparsity of the EHR data. Sparse representations are harder to model both for computational reasons and from the viewpoint of capturing the correct predictive signals which of the information in patients' EHR data are interrelated, conveying interchangeable or opposing information regarding patient conditions. For example, various medications are used to treat blood pressure-related conditions, including diuretics, beta-blockers, and alpha-1-agonist medications, while each

group includes many specific drugs. Therefore, if the accurate assignment of blood pressure diagnoses depends on capturing the presence of such medications, the model must learn to include all possible variants. However, this can be a challenging problem. Therefore, an ideal feature representation from patients' EHR data should be able to learn to combine such overlapping information in ways that can be easily consumed by the machine learning models.

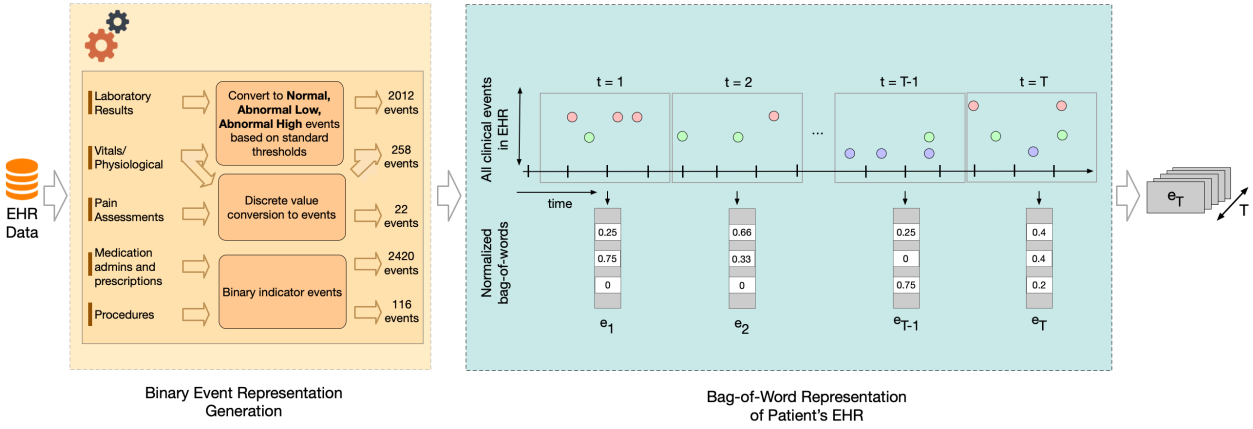


Figure 12: Steps to obtain a normalized bag-of-word representation of patient's EHR data and the corresponding number of events for each category in the MIMIC-III dataset which will be explained later in Section 3.6.1.1

3.5.3 Learning Dense Representation of Patient's EHR Data

To address the shortcomings in patients' bag-of-word representations, we propose to adopt lower-dimensional representation methods which can learn compact representations of patient data that a simple bag-of-word model fails to do. Thus, the key to our approach is to obtain lower-dimensional representations that learn to summarize (compress) patients' data into key principal components representative of patients' information and conditions. We define a low dimensional embedding as a mapping $E \mapsto \mathbb{R}^k$ that maps a patient's bag-of-words representation u to a new lower dimension dense vector $v \in \mathbb{R}^k$ while $k << |E|$,

and $|E|$ refers to the total number of binary events. Throughout this work, we utilize both unsupervised and supervised techniques to achieve this goal.

3.5.3.1 Unsupervised Method

To learn an unsupervised lower-dimensional representation of a patient's BoW vectors, we utilize latent semantic indexing (LSI) [42]. LSI is a statistical method for analyzing the relationship between a set of documents and terms used in information retrieval by finding underlying concepts. This is done by finding a Singular Value Decomposition(SVD) of the original term-document matrix A in which each row corresponds to the BoW representation of one patient's EHR. Alternatively, when segmentation is required, the rows will refer to a patient's EHR segment. In SVD, the underlying concepts are, in fact, eigenvectors of the symmetric matrix $U^U X$ and are represented in the left singular vector matrix in $A = Z\Sigma V^T$. Therefore, rank k Singular value decomposition of patient bag-of=word matrix $U_{|D|,|E|}$ can be obtained as:

$$U_{|D|,|E|} = Z_{|D|,k}\Sigma_{kk}{V^T}_{k,|E|} \tag{15}$$

Thus, the lower dimensional representation of v_i can be obtained as $v_i = u_i V\Sigma^T$.

3.5.3.2 Supervised Method

Our main goal is to propose a supervised alternative to the unsupervised technique that can be adopted in end-to-end neural network architectures later proposed in Section 3.6.2 and Chapter 5. Hence, we adopt a simple solution as a baseline method that consists of learning a single fully connected layer to learn a linear dense, and lower-dimensional representation of a patient's data. This approach has shown promising results for predicting a wide range of clinical events [107].

3.6 Classification of Patient Discharge Diagnoses and Diagnostic Categories

3.6.1 Independent Learning of Standard Machine Learning Models

In order to study the usefulness of the proposed unsupervised methods, we learn classification models for patient diagnostics from the entire patient EHR for each diagnostic task independently. Here, we learn one model per y_i (diagnosis or diagnostic category) using the support vector machine (SVM) algorithm with an L2 regularization term to capture the input-output relations. To address the low imbalance ratios for diagnostic target tasks, random over-sampling and under-sampling were applied to increase the training prior to the training data to a minimum threshold. Note, that using over/under-sampling techniques on the test set will not be appropriate.

All models use the apriori trained low-dimensional representations using the unsupervised technique as a preprocessing step as their inputs. If the lower-dimensional representation is successful in capturing important information about the patient visit in a compact form, we expect it to be sufficient. Finally, we note that this approach is not optimized to capture the hierarchical relations among different diagnoses and their categories and leave the study of these models to future chapters.

3.6.1.1 Experiments

To evaluate our methods we used two different electronic health record datasets:

MIMIC-III Dataset: We experiment with our models on the MetaVision part of MIMIC-III [87], an open-access EHR dataset obtained over a 12-year time span that covers more than 22,000 patient visits or hospitalization to ICU. MIMIC-III encodes patients' diagnoses using standard ICD-9 codes.

NOMA Dataset: The second dataset in this research was extracted from 15 Intensive Care Units (ICU) from the University of Pittsburgh Medical Center's electronic health records. The dataset includes more than 89,000 ICU admissions admitted during 2009 - 2018. However, in this study, we limited the data to patient visits before 2016 that continued to use ICD-9 diagnosis codes and included 45,257 ICU admissions to be consistent with the

MIMIC-III dataset.

Defining Study Population, Target Tasks, and Clinical Variables: We enrich the ICD-9 codes with diagnostic categories defined by the ICD-9 hierarchy. We limited our experiments to ICD-9 codes with at least 100 positive samples chosen to guarantee sufficient positive labels for learning and cross-validation. The input (feature) clinical variables used in learning patient representations included patients' vital signs, respiratory settings (noma), and other standard physiological data, medication orders, medication administrations, laboratory results, procedures, and surgeries. Table 1 depicts the statistics that define the scope of the data. Patient EHR data records may include many noisy and invalid data resulting from mistakes in data entry. It may also include many rare data recordings. For instance, clinical studies can record many data types that are only designed for a particular study. Therefore, aiming to exclude such clinical variables from the input features, the final set of input clinical events was limited only to those that were at least recorded for 100 patients. Finally, the study population was defined based on the length of stay (LOS), excluding any admissions with LOS less 4 hours and more than 31 days as a cutoff threshold.

Table 1: Basic information about each EHR dataset used in this study.

Dataset	Admissions	Diagnostic Tasks	Clinical Events
MIMIC-III	22,046	1165	6274
NOMA	43,788	2006	8507

Selecting the Optimal Number of Dimensions: Latent semantic indexing relies on the SVD algorithm to learn a lower-dimensional representation of the original data in which the dimensions represent orthogonal eigenvectors. An ideal lower-dimensional representation should retain all critical information needed for reconstructing the original data while using the minimum number of dimensions possible. Hence, we use the matrix reconstruction error to guide the selection of the optimal number. Our goal is to identify the smallest number

of eigenvectors that offer minimal information loss. This is usually done by plotting the reconstruction error as a function of the number of dimensions and finding the elbow of the curve, which means that adding additional dimensions does not result in a significantly lower reconstruction error. We calculate the reconstruction error as the L-2 norm ($\frac{1}{N}||X - X_k||^2$) of the difference between the original input and the reconstructed matrix X_k using a latent semantic indexing model with k dimensions. Figure 13 plots the reconstruction curve for both NOMA and MIMIC-III datasets and suggests that the common ideal number of dimensions should range between 500 - 600. Therefore, we use 500 as the final optimal number of dimensions.

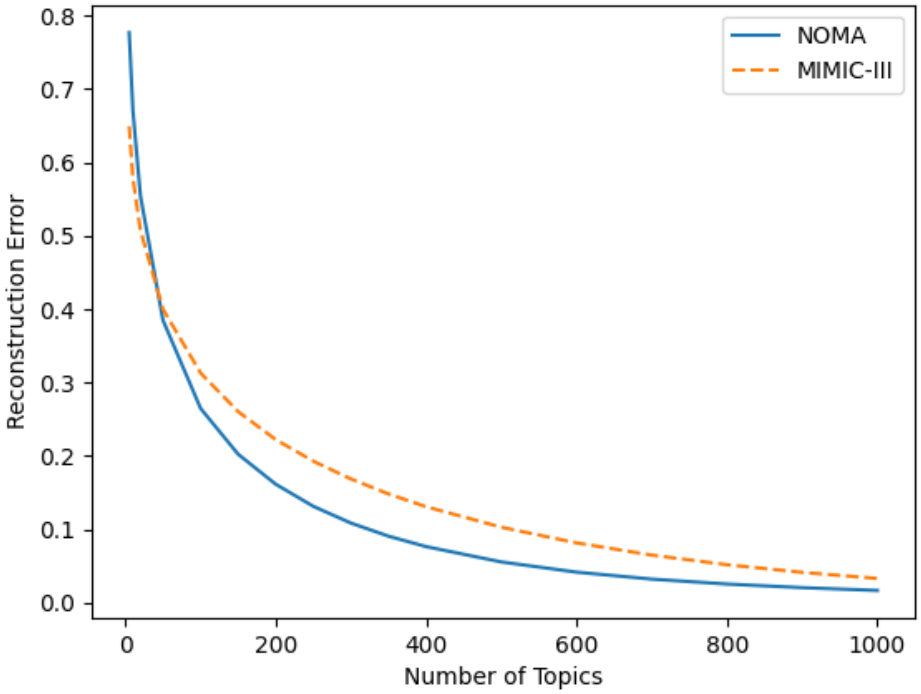


Figure 13: Reconstruction error of the lower-dimension representations as a function of number of dimensions.

Evaluation Metrics: We evaluate the performance of our models on the post-discharge diagnostic assignments expressed in terms of ICD-9 diagnoses and their categories using the

area under the receiver operating characteristics curve (AUROC) and the area under the precision-recall curve (AUPRC). The latter metric are known to be more appropriate in the presence of imbalanced data [73].

Quantitative Results: The overall results of the experiments are demonstrated in Table 2 for both datasets. Two sets of results are reported for the NOMA dataset separately: (1) the complete set of target variables as designed by the inclusion criteria, and (2) the equivalent set of target tasks to those included in the MIMIC-III experiments. The higher performance can be attributed to two major factors. First, the NOMA dataset includes a wider range of clinical variables used for learning unsupervised EHR representations, which can result in capturing patients' conditions and information more accurately. Second, the dataset is much larger than the MIMIC-III dataset leading to a higher number of positives for many diagnostic task, which in turn can result in training more accurate models.

Table 2: Average AUROC and AUPRC of diagnostic models for NOMA and MIMIC-III datasets trained using the SVD + SVM algorithm. Two sets of results are reported for the NOMA dataset separately: (1) the complete set of target variables as designed by the inclusion criteria, and (2) the equivalent set of target tasks to those included in the MIMIC-III experiments. SVD Features are trained using the unsupervised technique as a preprocessing step. All SVM models use the an unsupervised feature representation trained apriori.

	All Nodes		**Category Nodes**		**Leaf Nodes**	
Dataset	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
MIMIC-III (1165 tasks)	0.759	0.162	0.756	0.215	0.761	0.118
NOMA (2010 tasks)	0.778	0.133	0.782	0.188	0.776	0.095
NOMA (1165 tasks)	0.776	0.183	0.777	0.247	0.775	0.131

Furthermore, Table 3 depicts the diagnostic model performance for two sub-branches of the ICD-9 hierarchy for Heart Failure and Chronic skin ulcer. Many diagnostic models are

able to improve the classification precision when compared to the task's prior by multiple folds representing significant improvements indicating that the feature representations were effective in learning target diagnoses models.

Additionally, while we can't directly compare the performance of target tasks with one another, they can be compared from the viewpoint of clinical usability. The clinical usability of a machine learning model depends on two critical factors. First, sensitivity is important because the system should be able to identify all outcomes to help avoid the under-recognition of problems. Second, precision becomes important in preventing false positives. A high number of false positives often becomes important in determining the cost and risk effectiveness of a certain intervention. A high false positive ratio can result in lowering the cost-effectiveness of certain clinical decisions, thus, preventing physicians from adopting them. Another critical factor is false positive rates. Given a decision threshold to alert medical teams when high-risk patients are identified by machine learning risk models, high false positive rates can result in high false alert rates, thus, resulting in increased alarm fatigue. Therefore, an ideal clinical decision support (CDS) tool would be able to identify all outcomes while simultaneously minimizing the false positive rate. However, this may not be practical in many real-world problems. Therefore, we often reside on assessing the clinical usability of machine learning models according to the tradeoff between the model's precision and recall at different decision thresholds. This tradeoff can be clearly illustrated in the precision-recall curve. Figure 14 includes the precision-recall curve for a number of the diagnoses models classified under Heart failure. The plot shows that the machine learning models trained for high-level diagnostic categories provided higher precision levels within the low-recall regions. In other words, the high prediction scores by those models could provide a considerably highly confident recommendation. Therefore, we were able to learn better and more useful machine learning models for the higher level categories.

3.6.2 Modeling Patient Diagnoses using Recurrent Neural Networks

In the previous section, we proposed a general framework for independently learning classification models for patient diagnostic tasks. However, this approach suffers from a

Table 3: The model performance using the NOMA dataset for select subsets of the diagnoses hierarchy

Task Name	Prior	AUROC	AUPRC	Task Name	Prior	AUROC	AUPRC
Heart failure	0.192	0.849	0.616	Chronic ulcer of skin	0.053	0.818	0.232
Systolic heart failure	0.032	0.871	0.225	Pressure ulcer	0.041	0.827	0.213
Systolic hrt failure NOS	0.007	0.752	0.028	Pressure ulcer, site NOS	0.005	0.643	0.015
Ac systolic hrt failure	0.007	0.850	0.050	Pressure ulcer, low back	0.026	0.832	0.178
Diastolic heart failure	0.040	0.835	0.223	Pressure ulcer, hip	0.002	0.808	0.066
Diastolc hrt failure NOS	0.016	0.728	0.043	Pressure ulcer, buttock	0.008	0.797	0.045
Ac diastolic hrt failure	0.005	0.784	0.026	Pressure ulcer, heel	0.004	0.725	0.014
Ac on chr diast hrt fail	0.010	0.862	0.093	Pressure ulcer, site NEC	0.004	0.717	0.024
Cmbd sys & dias hrt failure	0.011	0.835	0.082	Ulcer of lower limbs, except.	0.013	0.801	0.064
Syst-diast hrt fail NOS	0.002	0.710	0.005	Ulcer of lower limb NOS	0.005	0.745	0.015
Ac-chr syst-dia hrt fail	0.006	0.832	0.058	Ulcer of heel & midfoot	0.003	0.719	0.023

number of shortcomings. First, the proposed method did not incorporate the temporal patterns in patients' clinical data. Therefore, making the trained models incapable of capturing important signals for the classification of some diagnostic codes that rely on recognition of such temporal patterns. Second, by treating the entire patient hospitalization as one unit(document), the BoW summarizing a patient's clinical data is doomed to combine contradictory or counter-intuitive signals. For instance, hypotension and hypertension are two clinical diagnoses referring to high and lower blood pressure conditions. A lengthy patient hospitalization might include symptoms of both conditions at different times. In such cases, the BoW representation will show signs of both conditions in one summarizing, which is contradictory. This motivates the need to adopt methods that capture and incorporate such temporal behaviors and heterogeneity in learned feature representations.

One popular way to learn models capable of capturing such temporal signals is using recurrent neural network (RNN) architectures. Here, we follow the notation described in Section 2.1 to obtain equal length time segments of patient's EHR hospitalization and train

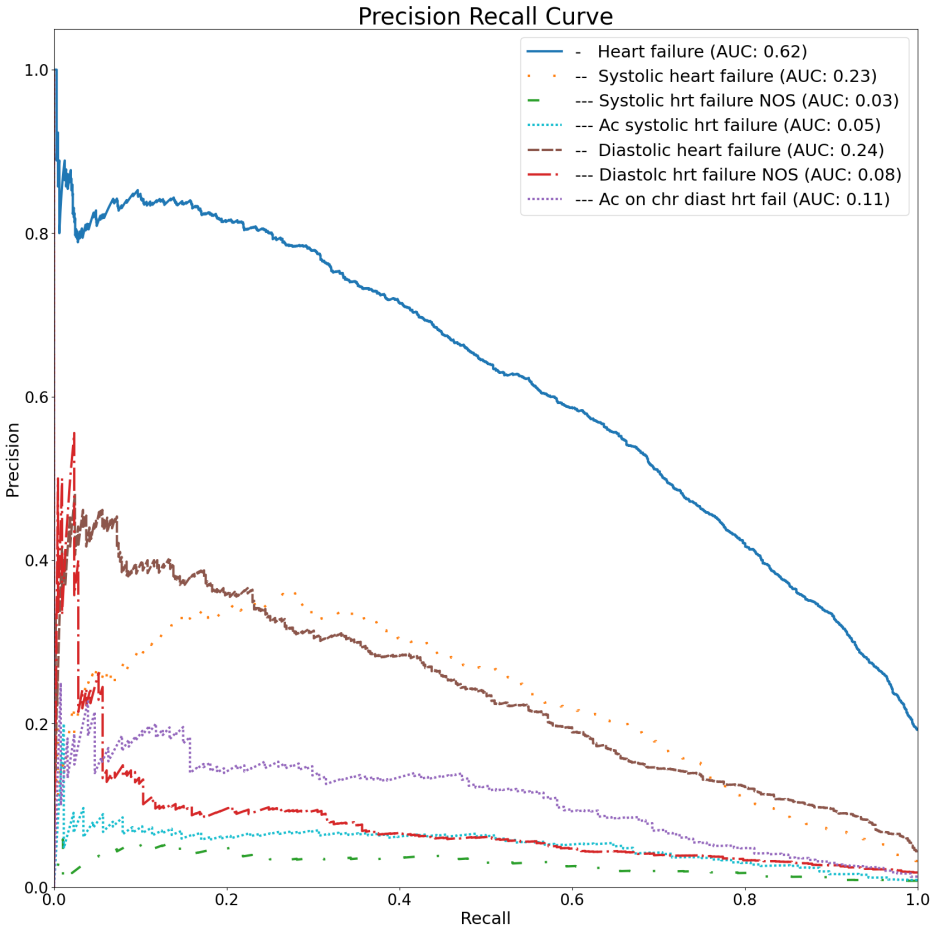


Figure 14: The precision-recall curve for the diagnostic tasks under the heart failure group

RNN models that utilize the low dimensional representation of the patient's clinical data during these time segments. To capture the temporal features from a broad range of patient clinical data, we adopt multiple well-known recurrent deep neural network architectures that systematically facilitate processing multivariate time-series data. Therefore, by formulating automatic assigning of patient discharge diagnoses as a sequence classification problem, we learn neural network models that capture from feature representations of segments of patient hospitalization and combine such features using recurrent models to learn a final dense representation that multiple target tasks can then use.

3.6.2.1 Preliminary Information

Methods adopted in this section include long short-term memory(LSTM), attention-based, and multi-head attention models, which differ in how they summarize the patient information when used for diagnostic predictions. In the rest of this section, we briefly summarize these architectures.

Long Short Term Memory Networks(LSTM) are a type of recurrent neural network architecture proposed to address the problem of long-term dependencies in recurrent neural networks [77] by taking advantage of extra network structures that are particularly responsible for deciding to remove or keep the information, often called gates. Like other RNN models, LSTM networks are sequentially applied to each time stamp t to combine and learn from two inputs: (1) the input features x_t for timestamp t, and (2) the hidden state of the previous timestamp h_{t-1} (Figure 15) to produce a new hidden state h_t.

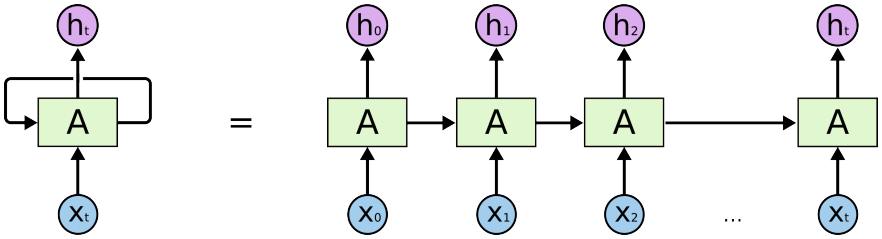


Figure 15: Application of recurrent neural network units to timeseries and sequence data

However, in contrast to standard RNN networks, LSTMs are designed to leverage gate modules that learn to decide either to emphasize on the input features or the previous hidden states. This allows the model to retain critical information from previous timestamps. The output or hidden state of time t in an LSTM network is defined as in equation 16 (see Figure 16:

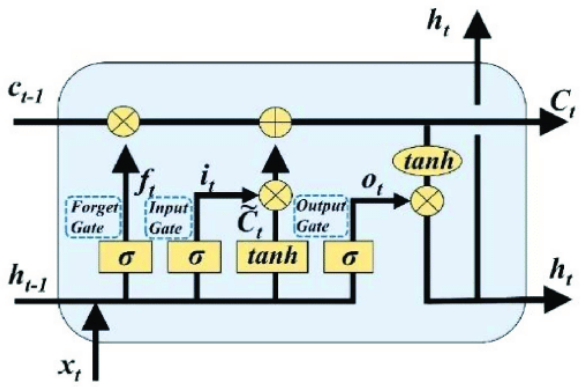


Figure 16: Long-term short-term memory unit architecture

$$
\begin{aligned}
i_t &= \sigma(W_i u_t + U_i h_{t-1} + b_i) & g_t &= \sigma(W_g u_t + U_g h_{t-1} + b_g) \\
f_t &= \sigma(W_f u_t + U_f h_{t-1} + b_f) & c_t &= f_t \circ c_{t-1} + i_t \circ g_t \\
o_t &= \sigma(W_o u_t + U_o h_{t-1} + b_o) & h_t &= o_t \circ tangh(c_t)
\end{aligned} \tag{16}
$$

in which i_t,f_t,o_t are sigmoidal control gates that determine how much of information each gates passes. The forget gate or f_t controls the amount of past information to to be copied to c_t and the input gate (i_t) controls the contribution of g_t, while, g_t function similar to a standard RNN. Finally, o_t controls how much of the current output c_t will be active. Today, LSTM networks have been widely adopted for various clinical time series. However, despite the motivation, it is shown that LSTM units can fail to retrain information in very long sequences [4].

Attention The shortcomings of LSTM networks to capture information from distanced past in very long sequences motivated the idea behind the attention mechanism. Attention layers are designed to learn an attention weight α_i for each step in a sequence that determines the importance of that times step for the prediction task. Attention is designed to address the problem of long sequences by creating direct shortcuts to all past timestamps. Finally, by learning a set of attention weights for each time stamp, it obtains a weighted average of all

input time-stamp to produce the final hidden state. One way to learn the attention weights is to use an additive mechanism where attention weights are defined as:

$$\alpha_{ti} = softmax(v_a^T tanh(W_a[s_t; h_i]) \tag{17}$$

where α_{ti} refers to the attention weight of timestamp i at step t and W_a and v_a are the attention weights. Finally, the attention output can be calculated as $c_t = \sum_i^n \alpha_{ti} h_i$.

Multihead Attention uses multiple attention heads in parallel to allow the model to simultaneously attend to multiple timestamps or input segments. This is especially intuitive in multi-label settings where accurate prediction of different target labels may rely on different input signals. In multi-head attention, each attention head learns a set of separate attention weights for the input sequence. In the end, the attention outputs are concatenated together to create the final representation. This allows the model to capture essential features from various timestamps that especially can be useful in multi-task problems such as the diagnoses classification problem in which the target task may rely on significantly different input signals.

3.6.2.2 Formulating Discharge Diagnoses as a Sequence Classification Problem

As discussed earlier, a major shortcoming of obtaining the feature representation of the entire patient's EHR data is that it fell short of capturing the temporal aspects and changes in the patient's clinical data during the visit. This motivates formulating the problem as a sequence classification problem. Sequence classification is a predictive modeling problem in which a sequence of observations or inputs is assigned to one or a set of categories or labels.

Following the notation in the beginning of this chapter (Section 3.5.1), we segment patient's length of stay into equal length time segments. Each time segment will represent a fixed time window and will be defined as $V_i = \{X_i, Y_i\}$ in which $X_i = \{x_i^1, x_i^2, ..., x_i^T\}$ and T refers patient's length of stay. However, the target task labels in Y_i will continue to be defined for the entire hospitalization and can be written as $Y_i = \{0, 1\}^M$ in which M represents the number of target diagnostic tasks. Our goal is to learn either a function f or a set of functions $\{f_1, f_2, ..., f_M\}$ that map $X_i \mapsto \{0, 1\}^M$.

We define each time segment's input features by learning the dense feature representation of the patient's EHR data during the corresponding time window using the following steps. First, we create the binary clinical events recorded during at each time window. Next, we create a BoW summarization from the events and finally learn a lower-dimensional representation of the patient's EHR data during each time segment by adopting the supervised method proposed earlier in Section 3.5.3.2. Figure 17 shows the general architecture of a deep sequence classification model. The feature representation learning layer is followed by a temporal neural network block that learns a new set of features that capture the temporal patterns in a patient's clinical data. The summarization block is designed to summarize the temporal features across all timestamps into one vector, and the final layer is a binary classification layer that uses the temporal feature representations of the patients to assign each patient visit to one or multiple target diagnostic codes.

As illustrated in Figure 17, we adopted a bi-directional LSTM neural network layer to model the temporal layer. We implemented the summarization layer using multiple approaches, including global max-pooling (MP), global mean-pooling(AP), attention(Attn), and multi-head attention blocks (MHA). The global max-pooling and mean-pooling layers are standard methods of creating an overall summarization of the hidden states. A 1D global max-pooling block takes a tensor of size (number of channels) x (timestamps) and computes the max value of all channels across all timestamps. A self-attention layer can be viewed as a generalization of the global average-pooling block that computes a weighted average of hidden states in which weights are learned based on the hidden state values. Finally, the multi-head attention layer further extends the functionality of the attention layer. It offers the capability to focus on various timestamps during a patient's hospitalization that might require important information for the classification of different patient diagnoses.

Throughout this section, we formulate the problem of assigning patients' diagnoses as a multi-label classification problem. Hence, the final classification layer can be implemented as a multi-label linear layer with a sigmoid activation function. Finally, we use a multi-label binary cross-entropy loss function to train the proposed neural network. Therefore, the loss

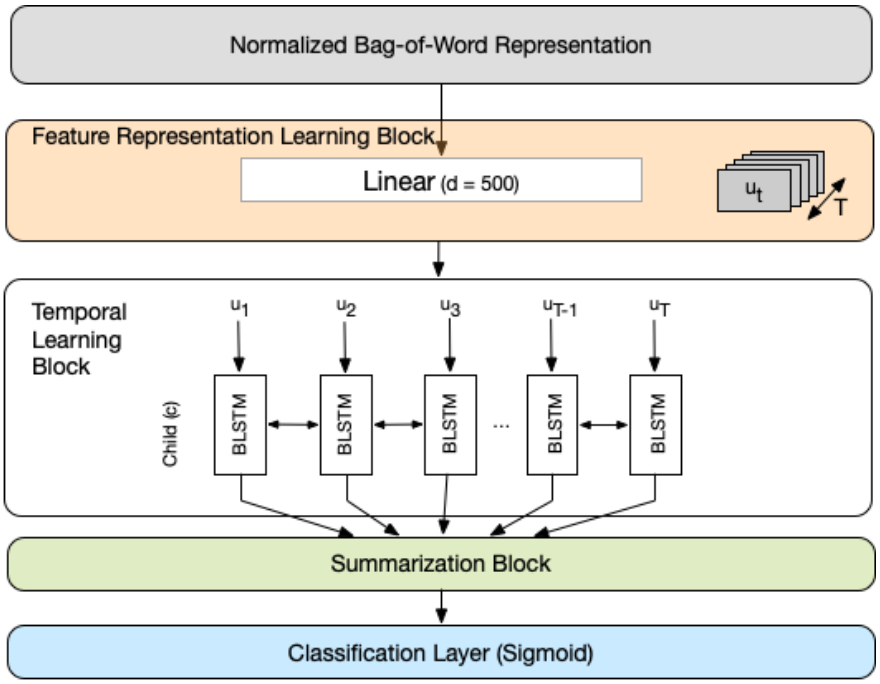


Figure 17: An overview of the deep neural network architecture for sequence classification

for sample i will be defined as:

$$Loss(y_i, f(i), c) = \frac{1}{L} \sum_{c}^{C} \mathcal{L}(y_i^l, f(i)^l) \tag{18}$$

in which C is the number of multi-label targets, L indicates the total number of target labels, and $\mathcal{L}$ refers to any binary classification loss. A common choice is the binary cross-entropy loss where $\mathcal{L}$ will be defined as:

$$\mathcal{L}(y_i^{\cdot} f(i)^l) = -y^l log(f(i)^l)) + (1 - y^l) log(1 - f(i)^l) \tag{19}$$

.

Implementation Details: The average of binary cross-entropy loss was adopted as a standard multi-label loss function. The size of the hidden state of the LSTM was set to 512, attention layers to 512, and the number of heads in the multi-head attention was set to 8

[143]. Dropout was used after all LSTM and attention layers with a probability of 0.5 to avoid overfitting. Additionally, the neural network architectures were optimized using Adam and a learning rate of 0.01. Finally, the patient segmentation windows were set to 24 hours based on the standard clinical staff's medical shift lengths.

Evaluation: All experiments were done based on a 70/30 percent train and test obtained using a multi-label stratification method [182]. Finally, the performance of the models was evaluated using the area under the receiver operating characteristics curve (AUROC) and the area under the precision-recall curve (AUPRC) and a statistical significance test was done using the pair-wise bootstrap-based method detailed in Appendix

Experiment Results: Table 4 depicts the overall model performance of the target diagnostic models across the model architectures in comparison with independently trained SVM models. The results demonstrate that the multilabel approach using Bidirectional LSTM and Max-Pooling is outperforming all of the other architectures and the independent SVMS. We believe the higher performance of the deep neural network solution can be attributed to two important factors: (1) the deep neural model retains temporal characteristics of patient clinical data, and (2) the deep learning solutions are jointly learning all target diagnostic tasks, thus facilitating the transfer of features through a shared feature learning layer (see 2.2.4.5). Finally, the overall model performances show that while some kind of summarization block was necessary, the simple approaches such as global max-pooling and mean-pooling performed as well or better than the attention-based models. Therefore, in future chapters, we will use global max-pooling as the primary method.

Table 4: Average AUROC and AUPRC of diagnostic models for NOMA and MIMIC-III datasets. The summarization layer was implemented using multiple approaches, including global max-pooling (MP), global mean-pooling(AP), attention(Attn), and multi-head attention blocks (MHA). Methods with a $\star$ sign were found to be statistically better than their baselines as outlined in Appendix .

		All Nodes		**Category Nodes**		**Leaf Nodes**	
Dataset	**Method**	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
MIMIC-III	SVD + SVM	0.759	0.162	0.756	0.215	0.761	0.118
	BiLSTM*	0.796	0.154	0.789	0.209	0.802	0.109
	BiLSTM + MP*	0.801	0.160	0.794	0.216	0.807	0.114
	BiLSTM + AP*	0.786	0.156	0.780	0.211	0.791	0.111
	BiLSTM + Attn*	0.775	0.135	0.768	0.190	0.780	0.091
	BiLSTM + MHA*	0.789	0.148	0.782	0.202	0.795	0.104
NOMA	SVD + SVM	0.778	0.133	0.782	0.188	0.776	0.095
	BiLSTM*	0.819	0.131	0.813	0.188	0.824	0.093
	BiLSTM + MP*	0.828	0.141	0.821	0.201	0.832	0.102
	BiLSTM + AP*	0.825	0.140	0.818	0.198	0.829	0.101
	BiLSTM + Attn*	0.820	0.134	0.813	0.191	0.824	0.096
	BiLSTM + MHA*	0.827	0.138	0.821	0.197	0.831	0.098

Table 5 compares the SVM models' performance with the best deep neural network models for the same subsets of the ICD-9 hierarchy reported in Section 3.6.1. The results across specific tasks also collaborate similar findings, which is that the joint training of the target task models was able to improve the model performances in some cases significantly. The improvements seem more consistent among tasks in the lower hierarchy levels for which independently trained SVM models had considerably lower performance.

Table 5: The model performance comparison between individually trained SVM models and the Bi-LSTM methods for a select subsets of the diagnoses evaluated using the NOMA dataset.

	SVD + SVM		BiLSTM + MP			SVD + SVM		BiLSTM + MP	
Task Name	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**	**Task Name**	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**
Heart failure	0.849	0.616	0.872	0.639	Chronic ulcer of skin	0.818	0.232	0.831	0.255
Systolic heart failure	0.871	0.225	0.894	0.296	Pressure ulcer	0.827	0.213	0.847	0.211
Systolic hrt failure NOS	0.752	0.028	0.814	0.028	Pressure ulcer, site NOS	0.643	0.015	0.847	0.025
Ac systolic hrt failure	0.850	0.050	0.888	0.061	Pressure ulcer, low back	0.832	0.178	0.874	0.197
Diastolic heart failure	0.835	0.223	0.868	0.207	Pressure ulcer, hip	0.808	0.066	0.884	0.106
Diastolc hrt failure NOS	0.728	0.043	0.813	0.051	Pressure ulcer, buttock	0.797	0.045	0.803	0.037
Ac diastolic hrt failure	0.784	0.026	0.872	0.034	Pressure ulcer, heel	0.725	0.014	0.856	0.025
Ac on chr diast hrt fail	0.862	0.093	0.911	0.118	Pressure ulcer, site NEC	0.717	0.024	0.777	0.016
Cmbd sys & dias hrt failure	0.835	0.082	0.880	0.094	Ulcer of lower limbs, except.	0.801	0.064	0.830	0.121
Syst-diast hrt fail NOS	0.710	0.005	0.811	0.010	Ulcer of lower limb NOS	0.745	0.015	0.830	0.121
Ac-chr syst-dia hrt fail	0.832	0.058	0.917	0.068	Ulcer of heel & midfoot	0.719	0.023	0.813	0.021

3.7 Qualitative Evaluation of the Model Performance in Dynamic Environment

In this section, we explore the behavior of the previously trained models using discharge labels by comparing the model prediction when applied to dynamic settings with clinical physician notes. We argue if the proposed methodology for learning patient representations is useful, we should be able to use the trained machine learning models to summarize patients' active diagnoses during each time segment. Ideally, we would expect to compare the model performance with dynamic diagnoses labels. However, this can be challenging or even impossible due to the lack of consistent real-time diagnostic labels. Hence, in this section, we rely on a qualitative evaluation of the model's dynamic behavior in comparison with their physician notes. Physicians' clinical notes often include their opinions and suspicion about potential conditions and diagnoses throughout the hospitalization. Therefore, we would expect diagnosis scores assigned dynamically by the models to demonstrate trends that are comparable with physician's notes. Since such performance comparison is not scalable, we limit the evaluations in this section to two example patients.

Applying these models to dynamic settings is not a straightforward problem. The di-

agnostic machine learning models based on the discharge labels are trained to use lower-dimension representations of patients' entire data during their hospitalization. However, to test the application of the diagnostic models in dynamic settings, patients' diagnoses should be classified according to their clinical data and information at or around a particular time during their visit. In other words, BoW summarization of patient information should be redesigned to capture underlying conditions currently visible at or close to time t. To do this, we introduce a new lookback mechanism with parameter λ, which defines the number of past time segments used at each timestamp t as input of the diagnostic models (Figure 18), and $u_i^{t'} = lookback(U^i, \lambda)$ in which u_i^t is the BoW summarization of patient's data within time segments $[t-\lambda, t]$. When $\lambda = 0$, $u_i^{t'}$ will simply become u_i^t and when $\lambda = \infty$, $u_i^{t'}$ will be the cumulative bag-of-word representation of patient data since the beginning of the patient hospitalization.

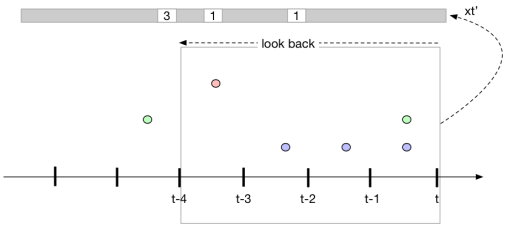


Figure 18: Lookback mechanism

Figures 19 and 20 shows predicted probabilities for dynamic assignment of patient diagnoses for two patients. In addition, Table 6 demonstrates relevant samples of clinical notes for the same patients as recorded in patients' EHR data. For the sake of reproducibility, we use the original de-identified patient identifiers (HAMID) provided in MIMIC-III dataset to refer to each patient. Patient visit 100182 in Figures 19a, 19b and 19c corresponds to a patient with admission diagnosis of chest pain. Her clinical notes (Table 6) indicate that she was initially admitted due to heart failure and had a prior history of acute kidney failure. She was also diagnosed with hyposmolality during hospitalization. Eventually, her conditions improved, and she was discharged to a skilled nursing unit which confirms the

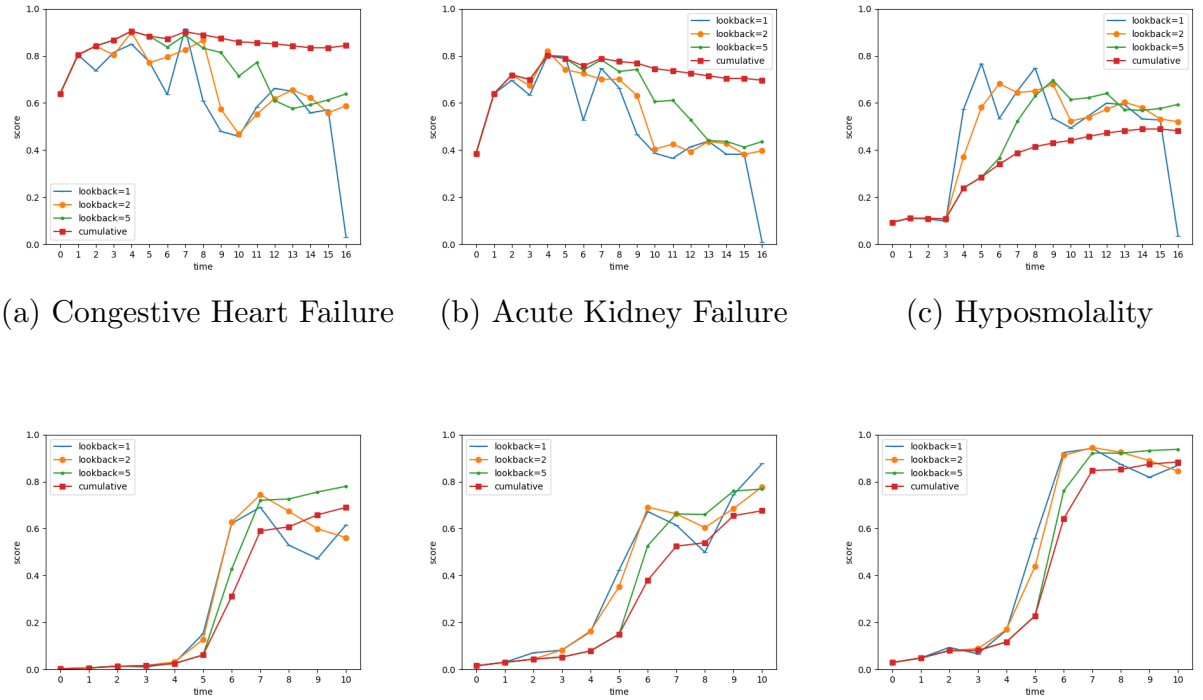
decreasing probabilities of diagnoses. This is captured by the dynamic predictions provided the corresponding models in Figure2 19.

Figures 19d, 19e and 19f belong to the visit id 120073 and show the development of sepsis, acute kidney failure, and finally, septic shock toward the end of the visit. The patient was admitted on 4/24 and passed away on 5/04. As notes indicate, the presence of renal failure is observed on 5/03, which validates the increasing predicted probability of aforementioned diseases by our models near the same time clinical notes show suspicion of infection, our models show a higher probability for sepsis. Eventually, the patient is diagnosed with severe sepsis and later enters sepsis shock.

Figure 20 shows the advantage of learning diagnosis category models. Figures 20a and 20b show the probability of specific diagnosis codes for the type of kidney failure assigned. At the same time, Figures 20c and 20d show the probability estimates for general acute kidney failure. Neither of the specific models has been able to assign patients with their actual assigned diagnosis, while the general model is confidently predicting acute kidney failure. This demonstrates not only the usefulness of more general category models in providing more accurate recommendations but also their capability of better capturing important signals in patients' clinical data that are predictive of the target diagnoses.

3.8 Conclusion and Discussion

In this chapter, we provided a simple yet flexible method for learning from a wide range of patient's clinical data in electronic health records that could be used for both supervised and unsupervised feature learning. We next developed machine learning methodologies based on independently trained SVM models and jointly trained multi-label deep neural networks to demonstrate the usefulness of the proposed feature representations in modeling a wide range of machine learning problems. Next, we evaluated the methodologies for assigning patients' diagnoses and diagnostic categories. Our results showed that the proposed method could effectively capture patients' necessary underlying clinical conditions and information for learning accurate diagnostic models. Finally, we demonstrated that an expressive feature



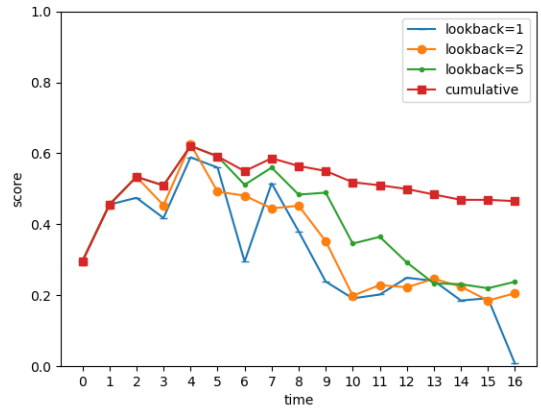
(a) Congestive Heart Failure (b) Acute Kidney Failure (c) Hyposmolality

(d) Severe Sepsis (e) Septic Shock (f) Acute Kidney Failure

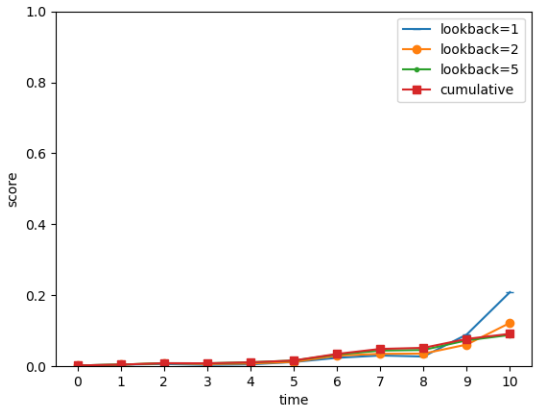
Figure 19: Dynamic predictions of diagnostic models for two sample patients (top: 100182, bottom: 100182)

learning method should not only combine various clinical information in patients' EHR to capture their important underlying conditions, but it should also be able to capture temporal patterns in patients' clinical data to achieve the best performance.

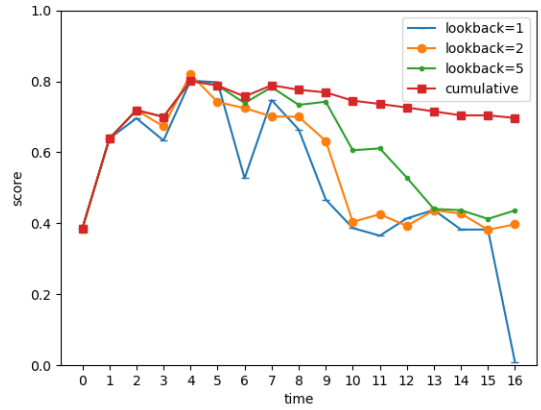
As discussed earlier in this chapter, the problem of modeling patients' diagnoses and diagnostic categories using their clinical information has been studied. However, most of the existing work relies on physicians' clinical notes [159, 144, 168]. One reason that patients' clinical notes might have been preferred in modeling patients' diseases is that they may consist of features directly indicating the presence of a particular diagnosis, thus, simplifying learning machine learning models that may not be as straightforward in patients' clinical EHR data. However, using patient structure clinical data in the EHR can also offer multiple



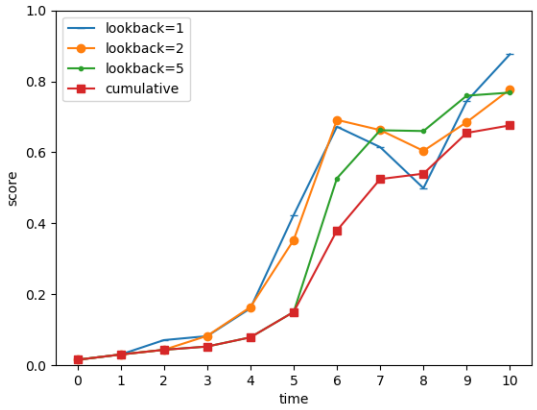
(a) Acute kidney failure NOS (584.9) for 100182



(b) Ac kidny fail, tubr necr (584.5) for 120073



(c) Acute Kidney Failure(584) for 100182



(d) Acute kidney failure(584) for 120073

Figure 20: Changes in probabilities of medical diagnoses for two sample patients that shows the categorical model detecting a diagnoses that was not picked up by the model for the child diagnoses assigned to the patient

advantages. First, if equally accurate machine learning models could be trained without relying on physicians' notes, it can facilitate the development of an automated clinical decision support system that can monitor patients' conditions in addition to physicians. Therefore,

Table 6: Parts of clinical notes related to diagnosis in Figures 19 and 20

Patient 100182 [4/29 - 5/15]	
Date	Note
5/05	"...is a very pleasant...woman with ischemic CMPY ... one functioning kidney, renal artery stenosis initially presented with shortness of breath thought to be secondary to volume overload and heart failure..."
Patient 120073 [4/24 - 5/4]	
Date	Note
4/30	"...Diff use bilateral ground glass opacities (e.g 3/46, RML) concerning for, superimposed multifocal infectious process..."
5/02	"...A 55 year old man with Sepsis and ..."
5/03	"... man with septic/cardiogenic shock... now admitted for subdural hematoma and developing renal failure..."

problems unnoticed by clinicians can potentially be captured by the machine learning models. This is while relying on physician notes can bound the performance of the machine learning models to physician understanding of patient's conditions. Second, depending on physician notes may delay the assignment of diagnoses if deployed at the bedside since they rely on updates in clinician notes which are not regularly updated. In contrary to notes, patients' clinical data are often regularly captured and recorded by medical devices or clinical staff throughout their hospital stay, providing access to a more dynamic and real-time source of data. Therefore, machine learning models trained using patient EHR data may provide a preferred solution concerning clinical usability if integrated into clinical workflows.

In this chapter, we demonstrated that patients' clinical data in EHR can be used to train machine learning models that automatically assign patients' diagnoses. While our results are not directly comparable due to many factors, including the difference in the datasets and randomization in creating train and test splits, assuming that generally, diseases have similar priors, the performance of models in this chapter that use EHR data are comparable with other approaches using clinical notes. For instance, DeepPatient proposed an autoencoder network to learn a lower-dimensional representation of patients' clinical notes and achieved

an average AUROC of 0.773 across leaf diagnostic tasks using an 81,214 patient dataset from the Mount Sinai hospital.

In addition to the above conclusions, our results demonstrated that when target task models are trained jointly, it can result in learning improved machine learning models. This was visible across the overall performance of all diagnostic tasks and the individual select branches of the hierarchy when we compared the performance of the multi-label deep neural network architecture and independently trained SVM models. Furthermore, the deep neural network architecture facilitated knowledge transfer among tasks by co-learning a joint feature representation of patients' EHR data used by individual tasks. Combined with the capability of capturing temporal information and patterns in patient conditions, the deep neural network approach could significantly outperform the independently trained SVM models.

Last but not least, our results demonstrated that, at least from clinical usability, the parent diagnostic tasks could significantly outperform the individual child tasks. This motivates studying the second research question in this dissertation, "How can one leverage the hierarchical relationships between the tasks to allow transfer of parameters in both top-down and bottom-up fashion?" We will explore ideas to answer this question in the following two chapters.

4.0 Hierarchical Multitask Learning Methods based on Parameter Transfer

In Chapter 3, we proposed a simple yet flexible framework for learning lower-dimensional representations of patient data from a wide range of clinical variables captured in EHRs that could be used in multi-task settings such as classification of patient diseases. Furthermore, we proposed and evaluated these representations to learn diagnostic models for the patient diagnosis problem. First, we adopted an independent learning algorithm that separately learn classification models for each diagnoses. Second, we adopted recurrent neural network based architectures to jointly learn these models by formulating the problem as a multi-label classification problem. However, the aforementioned approaches do not attempt to incorporate the hierarchical relations among tasks to guide the transfer of knowledge.

This chapter describes new hierarchical multi-task learning methods based on transfer of model parameters. First, in Section 2.2.3.2, we propose methods that leverage parameter transfer techniques to facilitate information sharing across parent-child relationships. This sharing takes place through an adaptive mechanism and an iterative algorithm to share parameters in both the top-down and the bottom-up fashion. In Section 4.2, we refine the earlier approach by proposing a new class-dependent version of the adaptation algorithm that dissects the transfer among the tasks based on positive and negative instances. We show this refinement is able to improve the transfer and leads to better classification results.

4.1 HA-MTL: Hierarchical Adaptive Multi-task learning

Earlier in Chapter 1, we argued that task categories represent a more generalized task compared to their children acting as a level of abstraction representing the commonalities across all group members. These abstractions can facilitate the learning of models that are easier to train and often more accurate and can facilitate co-learning of shared model parameters across all child target tasks. On the other hand, such abstractions may lose information related to particular signals that are critical for one or a number of its children.

Therefore, in this section, we focus on answering the following questions:

- Can we adopt parameter transfer techniques that use the stronger machine learning models at the higher levels of the hierarchy to learn improved models for the lower level tasks?
- Can we use lower level task models to learn improved models for the higher level categories by model parameter weights in a bottom-up fashion?

To study these questions, we propose a new hierarchical multi-task learning method called Hierarchical Adaptive Multi-task Learning (HA-MTL) which assumes that target tasks can exist across both leaf nodes and internal categorical nodes. Hence, it tries to improve machine learning models for all target tasks by transferring model parameters from both their parents and children. The idea behind HA-MTL is closely related to the top-down hierarchical classification models discussed in Section 2.2.1. This includes Sun et al.'s method that trained binary classifiers for each topic and topic category across a hierarchy in which each internal node's classifier would determine if an instance belongs to a sub-tree of the hierarchy [190]. It can also be found similar to Zhou et al. method that enforced orthogonality between parent and child task models [235]. However, one main difference between HA-MTL and the aforementioned approaches is that the end result for HA-MTL is a set of machine learning models for each task that can be used independently. This is because hierarchical classification methods usually adopt a recursive top-down classification algorithm to classify a new example.

In order to incorporate the benefits of the hierarchies into the learning process, we propose a new hierarchical adaptive learning framework that explicitly connects individual diagnostic tasks and attempts to use these connections to jointly learn a better collection of models. Our approach takes advantage of ideas implemented in the adaptive support vector machine approach and extends them to hierarchical task structures [215]. We test our new framework on MIMIC-III data where diagnoses are defined in terms of Ninth International Classification of Diseases (ICD-9) codes, and their hierarchy [188]. We show that our new framework improves upon diagnostic models built independently for each diagnostic task. We observe the effect of the hierarchy to be stronger for smaller training dataset sizes, demonstrating

that our framework can leverage the presence of a hierarchical structure to compensate for the lack of data and low priors when training the models.

The technical contributions of our work are two fold: (1) The design of Regularized Adaptive Support Vector Machine (RA-SVM) algorithm that can learn model parameters for a target classification task and its relation to auxiliary classification tasks simultaneously; (2) The development of a new multi-task learning framework that can leverage a predefined hierarchy of tasks to improve individual classification models by adapting parameters among parent and child tasks.

4.1.1 Proposed Methodology

Our goal is to learn predictive models for T tasks corresponding to diagnoses and diagnostic categories organized in a hierarchy. Each individual diagnostic task maps a dense representation of information in a patient's EHR (X) to one of the $\{0, 1\}$ labels. Each label reflects whether a specific diagnosis or a diagnostic category should be assigned to the patient defined by the information in X. The specifics of the X representation used in this section are not the main focus of this section and were covered in Chaper 3. We assume T classification tasks or diagnostic models here, are defined with the help of discriminant projections $f_1, f_2, ..., f_T$, where $f_t : X \rightarrow R$ reflects the specific class assigned to X for the task t depends on a threshold α_t defined on possible values of f_t.

A conventional approach is to learn each projection f_t independently. However, multi-task learning literature has shown that simultaneous learning of tasks can improve model performances [160, 231]. Unfortunately, in scenarios in which a large number of heterogeneous tasks exist, many multi-task learning algorithms that do not incorporate task relationships face negative transfer[160]. Hence, other multi-task learning methods have been proposed to learn relationships of target tasks to ultimately prevent negative transfer [15, 89].

Our objective in this work is to use a diagnostic hierarchy to guide the transfer of model parameters. Intuitively, when learning a diagnostic model, one can benefit from utilizing the models both from its immediate parent and children. This idea leads to the following

diagnostic model for task t:

$$f_t(x) = \sum_{j \in parent(t)} \tau_j f_j(x) + \sum_{i \in child(t)} \tau_i f_i(x) + \Delta f_t(x) \tag{20}$$

where parameters τ_k reflect the amount of transfer from task k and $\Delta f_t(x)$ is the task-specific component formed by a linear combination of features in x. Learning the best set of parameters $\Delta f_t(x)$ and transfer parameters τ_k is tricky because of circular dependencies in the definition of the functions. In this work, we solve the problem by defining a two-step (pass) algorithm to transfer parameters from one task to another. First, our algorithm learns a set of models by following the hierarchy in a top-down fashion where the transfer proceeds from higher-level diagnostic categories to lower-level diagnoses. Second, it uses the hierarchy to transfer the info in the bottom-up pass by adapting the model parameters from lower-level diagnoses to their immediate parents.

More formally, in the first top-down pass we learn models:

$$f_t^{td}(x) = \sum_{j \in parent(t)} \tau_j f_j^{td}(x) + \Delta f_t(x) \tag{21}$$

that ignore the influences from children. In the bottom-up pass, we consider the influences from children's models:

$$f_t^{bu}(x) = \tau_{td} f_i^{td}(x) + \sum_{i \in child(t)} \tau_i f_i^{bu}(x) + \Delta f_t(x) \tag{22}$$

Please note that both sets of parameters τ_i and $\Delta f_t(x)$ are re-optimized in every pass. The term $\tau_{td} f_i^{td}(x)$ in (22) represents a self adaption mechanism that enables transfer of parameters from the previous version of f_t trained to allow the model to keep any positive improvement during the top-down pass.

The above process involves learning a set of models $f_1, f_2, ..., f_T$ by adapting model parameters from hierarchically related or auxiliary models. Let $aux(t)$ define a set of auxiliary

models for model t used to train a specific version of f_t. We can rewrite the models trained in each pass as:

$$f_t(x) = \sum_{i \in aux(t)} \tau_i f_i(x) + \Delta f_t(x) \tag{23}$$

by simply varying the models included in the $aux(t)$ set. Recall the Adaptive Support Vector Machine (A-SVM) method from Section 2.2.3. A-SVM allows adaptive learning of $f_t(x)$ from the auxiliary tasks. However, A-SVM assumes that the weight of auxiliary tasks are known in advance. Hence, we propose Regularized Adaptive Support Vector Machine (RA-SVM) as a variation of A-SVM that simultaneously learns τ_a values and model parameters.

4.1.2 Regularized Adaptive Support Vector Machines

One shortcoming of A-SVM is that it requires the impact weight of each auxiliary task a as τ_a to be determined beforehand. This, however, is not sufficient for learning large hierarchies of tasks. Instead, it is favorable to use an algorithm that can simultaneously learn the importance of each auxiliary task. Therefore, we propose Regularized Adaptive SVM (RA-SVM), a new version of A-SVM that simultaneously learns the usefulness of auxiliary tasks while learning model parameters v_t (Δf_t). We achieve this by relaxing the assumption $\sum_{a \in aux(t)} \tau_a = 1$. Thus, as shown in (24), we introduce a new regularization term to regularize $\tau = [\tau_1, ..., \tau_a, ..., \tau_{|aux(t)|}]$ in which τ_a is the influence of auxiliary task a.

$$\begin{aligned} \min_{v_t, \varepsilon, \tau} \quad & \sum_{i}^{N_t} \varepsilon_i + C_1 ||v_t||^2 + C_2 ||\tau||^2 \\ s.t. \quad & y_i \sum_{a} \tau_a f_a(x_i) + y_i v_t^T x_i \geq 1 - \varepsilon_i, \quad \forall i \in \{1, ..., N_t\} \\ & \varepsilon_i \geq 0 \qquad\qquad\qquad\qquad\qquad\qquad \forall i \in \{1, ..., N_t\} \end{aligned} \tag{24}$$

Values of C_1 and C_2 determine the trade-off between regularizing model parameters and auxiliary task weights. We defined λ as $\lambda = \frac{C_2}{C_1}$. Higher values of λ will push further regularization of τ and therefore increase the impact of v_t in determining f. This translates to our tendency to independently learn the model parameters for task t. On the other hand, smaller values of λ imply that we prefer the model for task t to be more similar to auxiliary

task models. While the value of λ still needs to be determined using cross-validation or prior knowledge, it decreases the search space significantly while having an intuitive interpretation.

The optimization problem in (24) can be converted to a standard SVM optimization problem. Therefore, one can perform the optimization task using common existing libraries [192, 181, 25]. To do so, we define $F(x_i) = [f_1(x_i), ..., f_{|aux(t)|}(x_i)]$ for all auxiliary tasks. Next, we define a new weight vector v' and feature map ϕ over input feature x as shown in (25). Additionally, we define the cost parameter (C) in the standard SVM $C = \frac{1}{2C_1}$ and μ as $\mu = \sqrt{\lambda}$ for L2 regularization and $\mu = \lambda$ if L1 regularization is used. Therefore, the optimization problem in (24) can be re-written using new parameters and input shown in (25) and hence solved using any standard SVM library.

$$v' = [v_t, \mu\tau], \quad \phi(x_i) = [x_i, \frac{1}{\mu}F(x_i)] \tag{25}$$

4.1.3 Experiments

In this section, we first describe the used dataset, the adopted method for obtaining a dense representation of patients' EHR data, and evaluation metrics. Finally, we provide quantitative results and qualitative analysis of our method.

We conducted our experiments in this section using the MIMIC-III and NOMA datasets with identical configurations discussed in Chapter 3. Similarly, methods were evaluated using AUROC and AUPRC metrics (see Section 3.6.1.1), and a statistical significance as detailed in Appendix . Finally, we used the SVD + SVM methods proposed in Chapter 3 as the most appropriate baseline since it relied on an identical EHR lower-dimensional representation approach but learned machine learning models independently.

4.1.4 Quantitative Results

In order to compare the performance of our method with baselines, we used Area Under Receiver Operating Curve (AUROC) and Area Under Precision-Recall Curve (AUPRC). Finally, in order to test the significance of the improvements by our method, we used a pair-wise statistical test method that directly evaluates whether a method is statistically

significantly better than a baseline within the 95% confidence interval. We provide the details of the statistical test in Appendix . Additionally, we used random sub-sampling to generate 10 different 75%/25% train/test splits to evaluate the performance of the four methods described above. Moreover, we use 5 rounds of internal random sub-sampling for hyper-parameters optimization using the training set.

Overall Model Performance: The results in Table 7 depict the average AUROC and AUPRC of all tasks for HA-MTL compared to baselines. Our method is outperforming independently trained tasks across all nodes. Overall, the results show that the improvements are also consistent across both categorical and leaf nodes. The average model performance was improved by around 2% with respect to both AUROC and AUPRC, while more than 87% of the target tasks performed either better or as good as the baseline models. This overall improvement consisted of 35% and 10% of the target tasks experiencing respectively 2% and 5% performance improvement while only 1.6% (34 out of 2006) suffered from a considerable negative transfer (-2%). However, the greatest positive and negative improvements were significantly different from the average. When comparing the performance improvements across individual tasks, target task models were improved by up to 15% AUROC and 11% AUPRC, while the maximum negative transfer was -8% AUROC and -10% AUPRC.

The results align with our expectations, as we don't necessarily anticipate the adaptive models to improve all diagnostic tasks. On the contrary, we think the improvements would be gained when knowledge is transferred from more robust models to weaker ones. Thus, parent or child diagnostics that outperform their group will likely perform similarly. To better understand the impact of the knowledge transfer, it's helpful to study the model improvements across individual branches of the hierarchy.

The Challenge of Low Priors: One finding that was in contrast with our expectation was the existence of considerable negative transfers since we anticipated the HA-MTL algorithm to prevent dis-improvements by learning to minimize τ weights and thus avoid the impact of the auxiliary tasks. However, a careful review of the individual tasks relieved that most of these negative transfers happen in very low prior target tasks (average 0.002). This is while the HA-MTL method emphasizes regularizing the auxiliary task weights (reducing) using a model hyper-parameter that needs to be fine-tuned using internal cross-validation

for each target task. When the target task prior is extremely low, meaning that very few positive samples are available, the hyper-parameter tuning is prone to be biased and results in an imperfect result. Hence, this explains the presence of a few but considerable negative transfers.

Table 7: Average AUROC and AUPRC of diagnostic models for NOMA and MIMIC-III datasets. Methods with a $\star$ sign were found to be statistically better than their baselines as outlined in Appendix .

		All Nodes		**Category Nodes**		**Leaf Nodes**	
Dataset	**Method**	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
MIMIC-III	SVD + SVM	0.759	0.162	0.756	0.215	0.761	0.118
	SVD + HA-MTL*	0.779	0.173	0.778	0.230	0.781	0.126
NOMA	SVD + SVM	0.778	0.134	0.78	0.191	0.775	0.096
	SVD + HA-MTL*	0.794	0.149	0.793	0.20	0.795	0.105

Analysis Of Individual Task Models: To better understand the impact of the knowledge transfer, it's helpful to study the model improvements across individual branches of the hierarchy. Table 8 compares the performance of HA-MTL with individually trained SVM models on the same ICD-9 hierarchy subsets for "Heart Failure" and "Chronic Ulcer of sking" presented in earlier chapters. First, HA-MTL improved performance of individual diagnoses and diagnostic categories up to 8% in AUROC and 12% in AUPRC for some tasks. The improvements are usually greater in tasks with a lower prior. We believe this is because these tasks could benefit more by learning from the knowledge captured by the general group compared to their more commonly used siblings. One interesting observation specifically seen in "Chronic Ulcer of sking" sub-hierarchy is that all child tasks under the "Pressure Ulcer" category are equally improved. This is an excellent example of the general promise of MTL methods, which claims that simultaneous knowledge sharing through joint training of sufficiently similar target tasks can lead to improved performance of a set of equally weak target task models. Here, the incorporation of the task hierarchies allows the

model to identify similar groups at various levels of the hierarchy. On the other hand, it also means that the performance of the HMTL methods would depend on the quality of the provided hierarchy. Earlier in Section 1.3, we studied in detailed various forms of hierarchy imperfections and how they could impact the performance of hierarchical multi-task learning methods.

Table 8: The model performance comparison between individually trained SVM models and the HA-MTL methods for a select subsets of the diagnoses evaluated using the NOMA dataset

	SVD + SVM		SVD + HA-MTL			SVD + SVM		SVD + HA-MTL	
Task Name	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**	**Task Name**	**AUROC**	**AUPRC**	**AUROC**	**AUPRC**
Heart failure	0.849	0.616	0.858	0.634	Chronic ulcer of skin	0.818	0.232	0.831	0.253
Systolic heart failure	0.871	0.225	0.884	0.256	Pressure ulcer	0.827	0.213	0.834	0.232
Systolic hrt failure NOS	0.752	0.028	**0.781**	0.029	Pressure ulcer, site NOS	0.643	0.015	**0.689**	0.013
Ac systolic hrt failure	0.850	0.050	**0.884**	0.054	Pressure ulcer, low back	0.832	0.178	**0.850**	0.187
Diastolic heart failure	0.835	0.223	**0.864**	0.239	Pressure ulcer, hip	0.808	0.066	**0.850**	**0.125**
Diastolc hrt failure NOS	0.728	0.043	**0.791**	**0.078**	Pressure ulcer, buttock	0.797	0.045	**0.836**	0.052
Ac diastolic hrt failure	0.784	0.026	**0.816**	**0.030**	Pressure ulcer, heel	0.725	0.014	**0.755**	0.014
Ac on chr diast hrt fail	0.862	0.093	**0.905**	**0.110**	Pressure ulcer, site NEC	0.717	0.024	**0.767**	0.021
Cmbd sys & dias hrt failure	0.835	0.082	0.866	**0.096**	Ulcer of lower limbs, except.	0.801	0.064	**0.820**	0.078
Syst-diast hrt fail NOS	0.710	0.005	**0.793**	**0.007**	Ulcer of lower limb NOS	0.745	0.015	**0.786**	**0.024**
Ac-chr syst-dia hrt fail	0.832	0.058	**0.874**	0.068	Ulcer of heel & midfoot	0.719	0.023	**0.770**	**0.046**

Figure 22 shows the precision-recall plots for some of the improved precision-recall curves that resulted in improvements that could significantly enhance the clinical usability of the models. Earlier, in Section 3.6 we discussed that the clinical usability of models relies on a trade-off between high precision and high recall. That is, we would prefer models with predictions that offer a reasonable recall while guaranteeing a clinically meaningful precision. That is, when we analyze a model's precision-recall curve, we hope to see high score regions (low recall) that offer reasonably high precision. The examples of precision-recall in Figure 22 show how HA-MTL resulted in improvements that created such high precision regions that did not exist in the SVM baselines.

Impact of Small Datasets: Another motivation behind the adoption of multi-task learning methods is to mitigate the impact of a small data sample size. We will study this by artificially decreasing the size of the training dataset used for learning the target diagnostic

tasks. However, the test set used remained consistent across the various training sizes to allow one-to-one comparison. Additionally, we limited the study to 698 target diagnostic tasks that would have a sufficient number of positive samples when the training data was randomly reduced. Finally, we also evaluate the performance of two different versions of the algorithm with and without the bottom-up transfer of knowledge to study the importance of each step. Table 9, reports the results of our method across various artificially imposed limits on the available training data for each target task on the MIMIC-III dataset. The results show that our method outperforms the baselines on average and across different training sizes and that HA-MTL has been able to effectively find useful auxiliary tasks and adapt model parameters in even lower training data sizes. Another clear finding is that most improvements happen during the top-down step. This is more clearly visible in Section 4.1.5 where we study the transfer weights of auxiliary tasks.

Table 9: Average performance for all diagnostic tasks across different data sizes on the MIMIC-II dataset

Method Name	**AUROC**	**AUPRC**
Random (N=500)	0.5	0.065
SVD + SVM (N=500)	0.636	0.124
SVD + HA-MTL (N=500)	**0.656**	**0.13**
SVD + HA-MTL$_{td}$ (N=500)	0.655	0.129
Random (N=1000)	0.5	0.065
SVD + SVM (N=1000)	0.671	0.144
SVD + HA-MTL (N=1000)	**0.694**	**0.15**
SVD + HA-MTL$_{td}$ (N=1000)	0.692	0.148
Random (N=5000)	0.5	0.065
SVD + SVM (N=5000)	0.739	0.181
SVD + HA-MTL (N=5000)	**0.751**	**0.185**
SVD + HA-MTL$_{td}$ (N=5000)	0.746	0.184

4.1.5 Learned Auxiliary Task Weights

Figure 21 (a) and (b) show the learned transfer weights for the top-down and bottom-up steps[1]. We see that transfer of parameters occurs in both steps, but parent diagnoses have a stronger impact on improving child diagnoses. This agrees with our quantitative results showing the top-down step's impact is more significant. This can be explained by two intuitive reasons: First, diagnostic categories have a higher number of positive samples. Second, diagnostic categories, if defined properly, represent more general diagnostic tasks that are easier for training a model, as shown in past work [137]. Therefore, stronger models of parent diagnostic categories can translate into higher impacts in the top-down step.

Figure 21 (c) and (d) illustrate the weights of auxiliary tasks for top-down and bottom-up steps for the "Heart Failure" branch. We saw earlier in Table 8 that tasks under "Heart failure" are improved by the top-down step while "Heart failure" itself was not significantly improved. This can also be seen in Figure 21 (c) and (d) that parents generally have a higher impact on their children. This impact is as high as 0.96 for adaption of parameters from the diagnostic category "Combined systolic and diastolic heart failure" to "Ac/chr syst/diast heart failure", which means the parent model has equal importance as the learned model parameters for the target task (See (24)).

4.2 Class Dependent Hierarchical Adaptive Multi-task Learning

In this section, we first study how HA-MTL imposes similarities among the tasks in the top-down and bottom-up transfer of model parameters. Next, we propose a new hierarchical version of the HA-MTL method that allows asymmetric class-dependent adaptation of model behaviors by learning class-specific relatedness coefficients. Finally, we propose a new adaptive method to learn models with improved classification performance and analyze the difference between model adaptation from parent diagnostic categories for positive and negative classes.

[1] An interactive version is available at http://cs.pitt.edu/~salimm/hamtl/

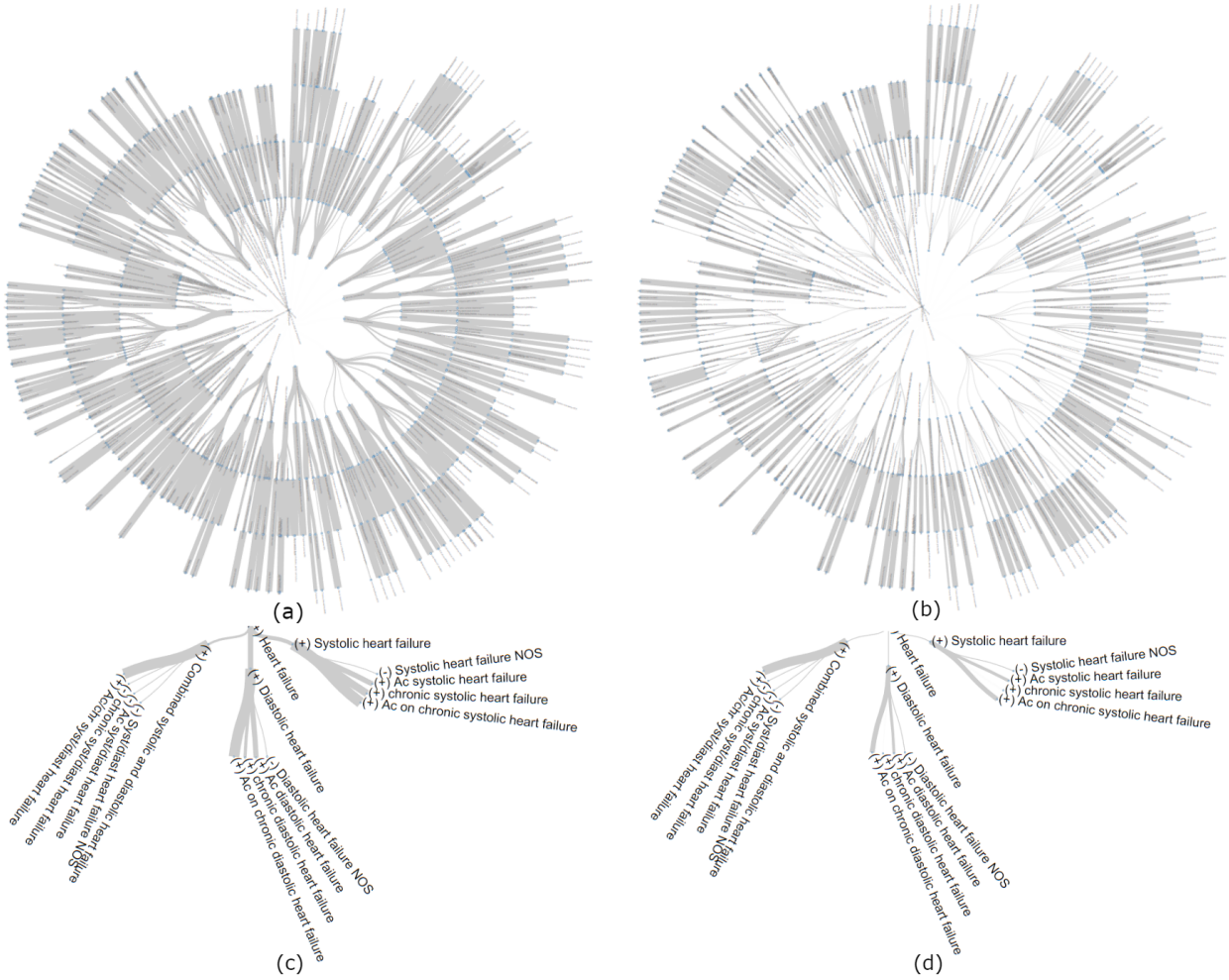


Figure 21: Figures (a)/(c) and (b)/(d) show weights of top-down and bottom-up steps for the entire hierarchy and the subset of it that belongs to "Heart Failure". Wider edges indicate higher weights and higher impact of auxiliary tasks. (+)/(-) signs show if a model was trained for the diagnosis code.

4.2.1 Proposed Methodology

Assume we have T diagnoses and diagnostic categories, each covered by a separate binary classification task. The tasks are organized in a hierarchical structure H. Additionally, we assume each patient's $X \in R^D$ consists of a D dimensional dense representation of the patient's EHR data. Our objective is to learn T discriminant functions $f_1, f_2, ..., f_T$ in which

$f_t : \mathbb{R}^D \rightarrow \mathbb{R}$. Hence, the predicted score of the discriminant function f_t can be mapped to one of the binary labels $0, 1$ using a task-specific threshold γ_t.

4.2.2 Understanding Hierarchical Adaptive Multi-task Learning

HA-MTL's goal is to adapt model parameters from parent and child diagnostic tasks while simultaneously learning the importance of the set of auxiliary models. They define the set of auxiliary tasks for the target task t as the set of its parent and child diagnostic tasks. HA-MTL improves each diagnostic task in an iterative fashion by proposing a two-phase (top-down and bottom-up) adaptation algorithm that transfers model parameters from either their parent or child diagnostic models. In order to perform the model parameter adaptation and simultaneously learn the importance of the auxiliary task, they propose Regularized Adaptive SVM (RA-SVM) as shown in Equation 24.

Lemma 1. *RA-SVM finds a trade-off between small impact of auxiliary tasks and similarity of target task model f_t to to the weighted average predictions of auxiliary models while minimizing loss*

This can be shown by re-writing the optimization problem in Equation 24 by replacing f_a with $w^{aT}x_i$ assuming all auxiliary tasks are trained using linear models and $w_t = \Sigma_a \tau^a w^{aT} x_i + v_t$ to refer to the final model parameters for target task t.

$$\begin{aligned} \min_{v_t, \varepsilon, \tau} \quad & \Sigma_i^{N_t} \varepsilon_i + C_1 ||w_t - \Sigma_a \tau^a w^{aT}||_2 + C_2 ||\tau||_2 \\ s.t. \quad & y_i w_t^T x_i \geq 1 - \varepsilon_i, \quad \forall i \in \{1, ..., N_t\} \\ & \varepsilon_i \geq 0 \qquad\qquad\quad \forall i \in \{1, ..., N_t\} \end{aligned} \tag{26}$$

The term $||w_t - \Sigma_a \tau^a w^{aT}||$ in 26 attempts to regularize the large difference between target task model outcomes from the auxiliary tasks. This is while $||\tau||_2$ promotes smaller influence of auxiliary tasks. This further clarifies the role of parameters $C1$ and $C2$. In fact high values of $\frac{C1}{C2}$ promotes further regularization of $||w_t - \Sigma_a \tau^a w^{aT}||_2$ and therefore promotes higher impact of auxiliary weights. In contrary lower ratios of $C1$ and $C2$ promote minimization of τ^a values and enable independent learning of f_t.

4.2.3 Not All Samples are Equal

RA-SVM method assumes the signal from auxiliary tasks is equally useful in improving the target task model's performance. However, intuitively one can imagine that auxiliary model scores in a hierarchical structure can have different meanings or impacts based on the type of dependency between related tasks. For instance, in the top-down adaptation phase, a negative score of the parent model (assuming the parent model has a higher performance) is more likely to translate to a negative label for the child task. In contrast, a positive class prediction of the parent may not necessarily mean that the child task will also be positive. As previously discussed in the hierarchical classification literature [185] negative samples in a hierarchy are passed top-down while positive class samples are promoted in a bottom-up fashion. Therefore, in this work, we propose an asymmetric adaptation mechanism based on ReLU operation to break down the scores of RASVM models to a pair of class-dependent signals $f_p^a = max(0, f_a)$ and $f_n^a = min(0, f_a)$.

The signals $f_p^a \in [0\ \infty]$ and $f_n^a \in [-\infty\ 0]$ allow target task models to learn two different relatedness coefficients τ_p^a and τ_n^a for positive and negative signals from auxiliary tasks. RA-SVM optimization problem in 24 can be written for AsymRA-SVM as shown in Equation 27.

$$
\begin{aligned}
\min_{v_t,\varepsilon,\tau} \quad & \sum_{i}^{N_t} \varepsilon_i + C_1||v_t||_2 + C_2||\tau||_2 \\
s.t. \quad & y_i \sum_{a} \tau_p^a f_p^a(x_i) + \tau_n^a f_n^a(x_i) + \\
& \qquad y_i v_t^T x_i \geq 1 - \varepsilon_i, \qquad \forall i \in \{1, ..., N_t\} \\
& \varepsilon_i \geq 0 \qquad\qquad\qquad\qquad \forall i \in \{1, ..., N_t\}
\end{aligned}
\tag{27}
$$

AsymRA-SVM enables the target task model to learn how important each signal from auxiliary tasks is by minimizing $||w_t - \Sigma_a \tau_p^a f_p^a + \tau_n^a f_n^a||_2$ and learning two separate weight for each auxiliary task based on the predicted score.

4.2.4 Optimizing Prediction Thresholds

AsymRA-SVM splits the auxiliary task signals by splitting the model outputs into two separate positive and negative signals by using zero as a default prediction threshold γ_t. However, due to highly imbalanced diagnostic tasks this can be an issue. To address this problem we attempt to optimize the decision threshold γ_t by maximizing the F1 score. Since F1 is neither differential nor a convex function [23] we used Particle Swarm Optimization method which has been shown to be suitable for ill-formatted, non-differentiable and non-convex optimization problems [184, 162]. To allow f_p^a and f_n^a to remain in $[0\ \infty]$ and $[-\infty\ 0]$ ranges we redefined f_t as $f_t = w_t + b_t + \gamma_t *$.

4.2.5 Experiments

We conducted our experiments in this chapter following the same configurations of MIMIC-III and NOMA datasets as discussed in Chapter 3.6.2. We evaluate our method by comparing the performance of AsymmHA-MTL to three baselines, including the independently trained SVD + SVM in Chapter 3 and the original HA-MTL method discused in Section 4.1.

Table 10: Comparison of AsymmHA-MTL with previous methods using NOMA and MIMIC-III datasets. Methods with a $\star$ sign were found to be statistically better than their baselines as outlined in Appendix .

		All Nodes		**Category Nodes**		**Leaf Nodes**	
Dataset	**Method**	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
MIMIC-III	SVD + SVM	0.759	0.162	0.756	0.215	0.761	0.118
	SVD + HA-MTL	0.779	0.173	0.778	0.230	0.781	0.126
	SVD + AsymmHA-MTL*	0.787	0.191	0.781	0.239	0.793	0.141
NOMA	SVD + SVM	0.778	0.134	0.78	0.191 M	0.775	0.096
	SVD + HA-MTL	0.794	0.149	0.793	0.20	0.795	0.105
	SVD + AsymmHA-MTL*	0.799	0.151	0.799	0.213	0.817	0.119

Table 10 shows the average AUROC and AUPRC for all target diagnostic tasks using

all methods. AsymmHA-MTL method outperforms the baselines and symmetric HA-MTL method. However, as discussed in [134] we expect the majority of improvements to happen on lower-level child diagnoses. Therefore, in table 11 we have provided results for parts of two sub-branches of ICD-9 hierarchy for Fracture of ribs and Diabetes. Our results show that considerable improvements have been gained by learning both HA-MTL and AsymmHA-MTL, and in instances, AsymmHA-MTL is outperforming the symmetric HA-MTL method. Similar trends are visible across different sub-branches of the hierarchy.

Table 11: Comparison of methods for example branches of ICD9

Diagnostic Task Name	**SVM AUROC**	**HA-MTL AUROC**	**AsymmHA-MTL AUROC**
Diabetes mellitus	0.866	0.863	0.864
_ Diabetes mellitus without mention of complication	0.714	0.718	0.773
__ Diabetes mellitus without complication, type II	0.689	0.68	0.757
_ Diabetes with hyperosmolarity	0.774	0.863	0.858
__ Diabetes with renal manifestations, type II	0.805	0.823	0.852
Fracture of rib(s) sternum larynx and trachea	0.914	0.912	0.919
_ Closed fracture of rib(s)	0.90	0.911	0.915
__ Closed fracture of multiple ribs, unspecified	0.690	0.764	0.833

However, one negative observation in the behavior of both HA-MTL and AsymmHA-MTL is that although many considerable improvements are made to lower-level child diagnostic, in many cases, RASVM methods have failed to prevent the negative transfer from parent diagnostic tasks. Since this is often happening in diagnostic tasks with a very high imbalance ratio (near 0.001), it seems both methods are sometimes failing to prevent negative transfer due to the failure to choose the right values of $C1$ and $C2$ hyperparameters using internal cross-validation. This can be explained by the significantly small number of positive samples in the validation and test set.

Figure 23 depicts the distribution of tau_p (impact of positive signals) and tau_n (impact of negative signals) in a top-down transfer of model parameters. The contrast between the

distribution of positive and negative signals shows that models are more likely to learn a higher impact for f_n^a when a is a parent of the target task. This agrees with our hypothesis discussed in Section 4.2.3 that negative signals from parent diagnostic tasks are more likely to translate to a negative score in the child diagnostic model.

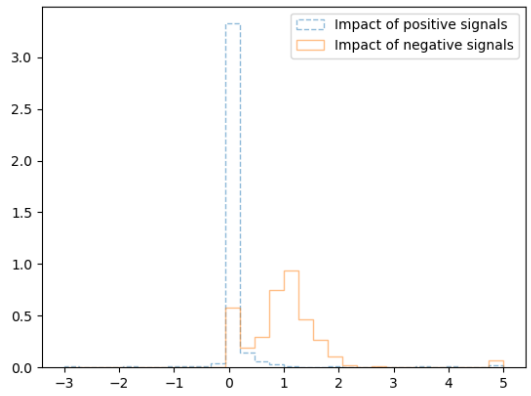


Figure 23: Distribution of τ_p and τ_a values in the top down transfer of model parameters

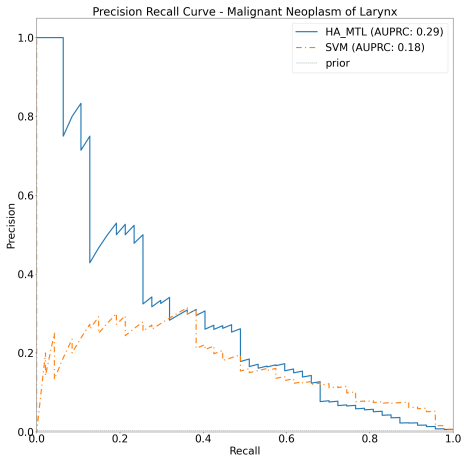
4.3 Summary and Shortcomings

In this chapter, we proposed a hierarchical adaptive multi-task learning framework for learning classification models for patient diagnoses and their diagnostic categories. Our method learns diagnostic models through a two-step process. First, it performs a top-down step that transfers model parameters from parents to children. Second, it performs a bottom-up pass that learns improved parent models by adapting from their children. By conducting experiments on MIMIC-III data and ICD-9 diagnosis hierarchy, we have demonstrated that our framework leads to improved performance when compared to independently learned models. This improvement is stronger for diagnoses with low prior and well-defined parent categories.

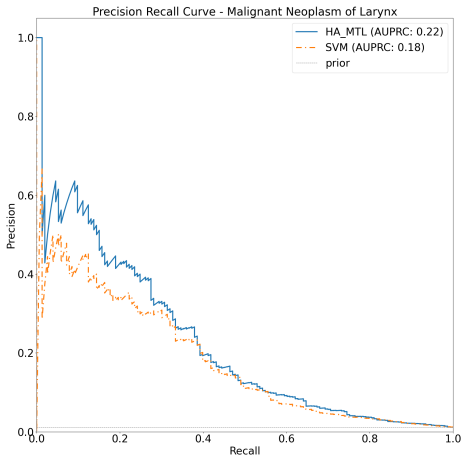
In Section 4.2 we argued that the usefulness and impact of related tasks in hierarchical multi-task learning problems could depend not only on the tasks but also on the classes of

samples. For example, in the top-down transfer of parameters, high negative scores of parent models are more likely to translate to negative scores of lower child models as negative labels are passed from a parent to children, while this is not necessarily true for positive labels. Therefore, we proposed an asymmetric hierarchical adaptive multi-task learning method that allows models to simultaneously learn model parameters and the importance of positive and negative scores of auxiliary tasks independently. Our results show that during the top-down model adaptation phase, our model is able to improve model performances compared to the symmetric version of the algorithm and baseline SVM models.

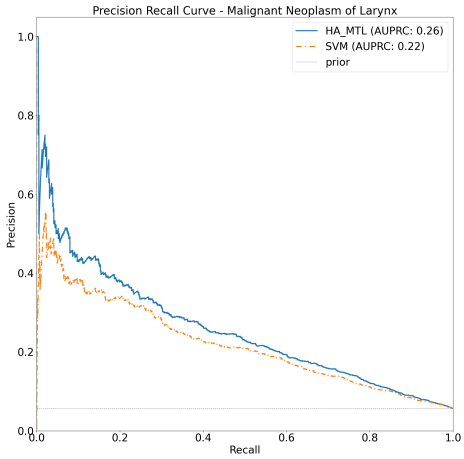
While our results show significant improvement across many of the branches of the hierarchy, a more detailed analysis of the results shows that the improvements are not consistent across all branches. In fact, we can observe negative transfer in some diagnostic tasks. Negative transfer happens in the top-down step when parent categories are not a suitable abstraction of the target child's task. This might be due to many reasons previously discussed as challenges in developing HMTL methods in Section 1.3. This includes the presence of outlier tasks, residual groups, and overly general categories. Imposing similarities in such unwanted situations will result in the reduced performance of models. Additionally, RA-SVM fails to prevent such negative transfers since learning model parameters happens individually for each target task and during an iterative process. This makes the algorithm sensitive to small sample sizes in leaf diagnostic codes during both model learning and hyper-parameter optimization.



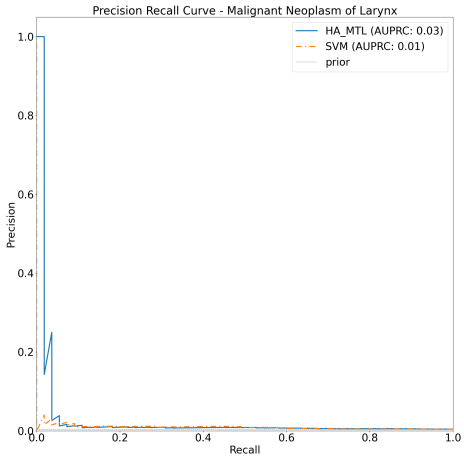
(a) Malignant neoplasm of larynx



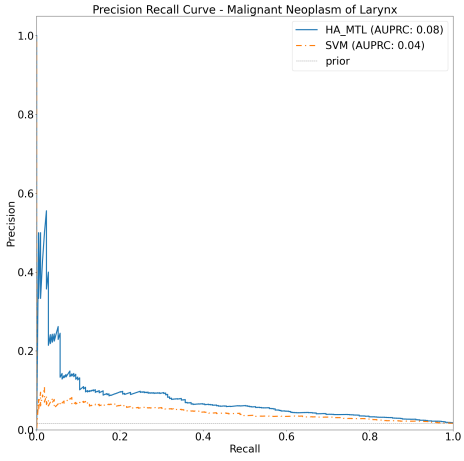
(b) Arterial embolism and thrombosis



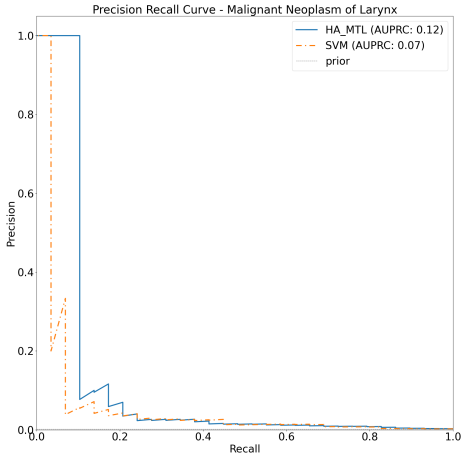
(c) Other venous embolism and thrombosis



(d) Stricture of artery



(e) Diastolc hrt failure NOS



(f) Pressure ulcer, hip

Figure 22: Changes in probabilities of medical diagnoses for two sample patients

5.0 Hierarchical Multitask Learning Methods based on Feature Transfer

In the previous chapter we explored ideas for iterative hierarchical multi-task learning methods that utilized parameter transfer techniques. However, we note that the design of the HA-MTL methods was somewhat limited resulting in multiple shortcomings:

- First, HA-MTL assumes the availability of common feature representations across all tasks. Thus limiting the ability of the target task machine learning models to learn task-specific features from the data.
- Second, as discussed in Section 4.3, the iterative one-by-one learning algorithm for HA-MTL results in unstable learning of models for lower-level tasks with very small priors for positive class due to models overfitting on training data.
- Third, it does not support the transfer of knowledge between siblings when it can be helpful.

Therefore, we aim to propose a novel deep HMTL framework that adopts feature transfer as the primary way of facilitating knowledge sharing between target task machine learning models. Our proposed method will guide the transfer of features in a top-down fashion allowing each target task model to either adopt or combine features learned by its parent task (shared feature representation among all of its siblings) with a new set of task-specific features learned separately from the group. Additionally, the proposed method aims to utilize a simultaneous learning algorithm as opposed to the iterative process adopted by HA-MTL and the Class-dependent HA-MTL described in Section 2.2.3.2 which could prevent overfitting, especially during hyperparameter optimization for tasks with a significantly small number of positive samples(rare diagnoses).

5.1 HD-MTFL: Hierarchical Deep Multi-task Feature Learning Method

In this section, we aim to propose a new hierarchical deep learning method that leverages the hierarchical structure of patient diagnoses to simultaneously learn the target task models and facilitate the transfer of information in a top-down fashion, from higher-level diagnostic codes with stronger classification models to lower-level ones. After that, we further refine the initial hierarchical model with a new disease interaction layer. Motivated by the field of differential diagnoses, the interaction layer learns to capture additional patterns from patients' EHR data to better discriminate among competing diagnoses and to fine-tune the predictions of the hierarchical layer. Finally, we will evaluate our proposed methods using the automated patient diagnoses classification tasks from electronic health records(see Section 3.5.1).

5.1.1 Methodology

Let D be the number of target diagnostic tasks of varying difficulty organized in a hierarchical structure H. Our goal is to learn classification models for each of these tasks by taking advantage of task relations reflected in H. The patients' EHRs are formed by complex sequences of observations, physiological events, treatments, and procedures. To facilitate the learning of classification models, the EHR sequences are often replaced with a compact vector-based representation that attempts to summarize the information in EHRs relevant to the specific prediction tasks. This transformed representation is also referred to as embedding. We follow the supervised method proposed in Section 3 to obtain lower dimensional representations of patient's EHR data that can be used for classification of patient's diagnoses. Finally, we add a new model layer that incorporates disease-disease interactions to learn additional task-specific features that aim to further refine the different diagnostic models. In contrast to HA-MTL, in this section, we aim to also capture the temporal patterns, changes and signals during patients' hospitalization. Hence, we follow the segmentation strategy proposed earlier in Chapter 3 and define each patient hospitalization as $V_i = \{U_i, Y_i\}$, while $U_i = \{u_i^1, u_i^2, ..., u_i^T\}$ are embedding at each timestamp and l_i refers

to the number of segments created during patient i's hospitalization. However, since we aim to solve a sequence classification problem, we continue to present $Y_i = \{0,1\}^M$ as a binary vector representing all diagnoses visible during the patient visit. For the sake of simplicity, we will commonly use T instead of l_i and ommit the visit index i throughout this section.

5.1.2 Hierarchical Multitask Learning Layer

Multi-task learning aims to train target tasks simultaneously and, hence, learn improved classification models by facilitating the transfer of knowledge between related tasks. In deep multi-task learning methods, this similarity is often achieved through either a set of common latent feature layers shared by all or groups of related tasks or through imposed similarities between a set of task-specific constrained feature layers. However, traditional methods may fail to efficiently leverage task relationships when facing a large number of heterogeneous tasks with various levels of similarities. There, hierarchical MTL methods aim to leverage underlying task hierarchies to efficiently direct sharing of information between target tasks.

Our proposed layer learns a separate set of task-specific neural network blocks for each target task in any arbitrary hierarchy while facilitating the inductive transfer of features in a top-down fashion by sharing hidden states of parent tasks with its children (see Figure 24). Additionally, following Sanh et al. [179] we use shortcuts (blue arrows) so that each target task can have access to the original EHR feature embeddings. This dual input mechanism enables each target task to either learn new features from the shared EHR embeddings, adopt features from more general categorical parent tasks p (black arrows), or combine these two sets of features in order to learn improved classification models. This is analogous to clinicians distinguishing specific diagnoses types by examining additional information that helps identify them from the other members of a group of diseases with similar symptoms. Task-specific blocks in this work are modeled using a bi-directional LSTM encoder architecture. The encoders take as input the concatenated vector of original EHR embeddings (v^t vectors) and the hidden states of their parent task p at each timestamp t (h_p^t). Next, a max-pooling layer($max([h_m^1, h_m^2, ..., h_m^T])$) for each target task was adopted to combine task-specific LSTM hidden states at all timestamps. Finally, a feed-forward layer with a sigmoid

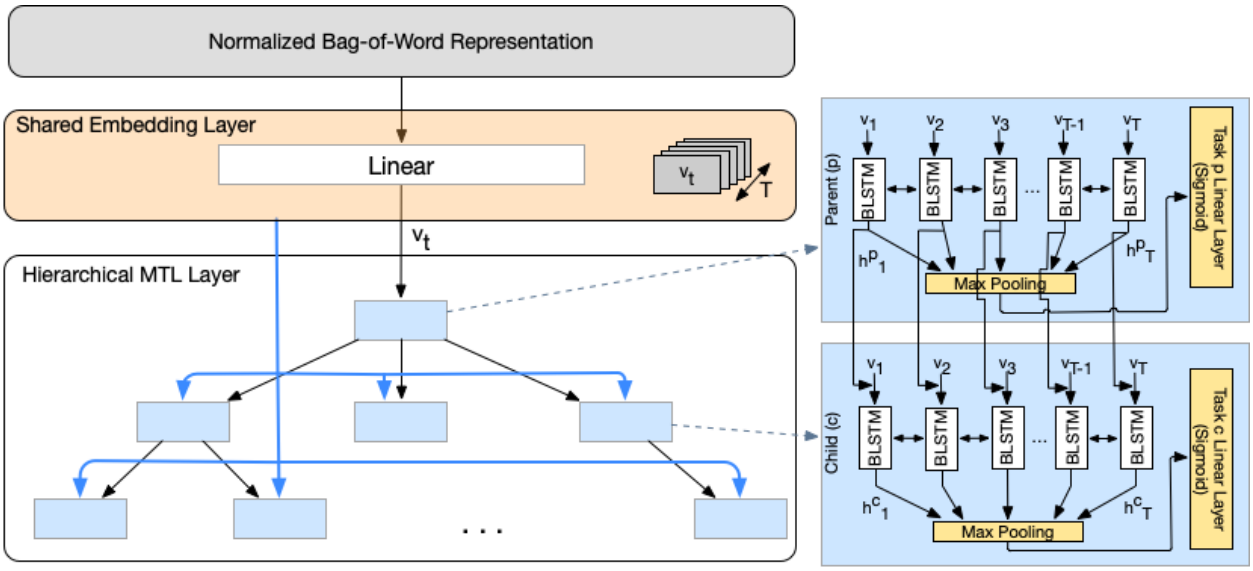


Figure 24: The proposed HD-MTFL network architecture

activation function was adopted to learn the final classification scores for each target task.

5.1.3 Disease-Disease Interaction Layer

Differential diagnoses in medicine refer to distinguishing a particular patient's disease from a set of competing diagnoses with similar features through systematic methods of acquiring and examining additional data. Similarly, a comprehensive machine learning solution should capture such disease-disease interactions to classify patients' diagnoses accurately. Therefore, we propose a fine-tuning step that is trained separately as a second step aiming to capture sibling interactions and improve the target task predictions by the hierarchical layer based on the prediction scores of other tasks. The proposed layer is designed to:

- Capture additional features and patterns from patient's EHR data that would allow the model to make a final prediction score for each target task m consider how it might interact with its siblings given the clinical context of the patient.
- Since patient's EHR data might represent a long hospital visit, such important new features could be presented at a or multiple specific times throughout the hospitalization.

Therefore, requiring the interaction layer to be able to attend to critical time segments that would contain relevant information.

Thus, the proposed interaction layer defines the final prediction probability for the target task m as $\hat{f_m} = sigmoid(f_m + \Delta f_m)$ where f_m is the initial score based on the hierarchical model and Δf_m determines the change to the scores based on the disease-disease interactions with its siblings. Motivated by the field of differential diagnoses, a task-specific feature attention-based learning block is adopted to learn additional features (Figure 25). First, a single linear layer is used to learn a low-dimensional task-specific feature vector v_m^t from the original EHR embeddings v^t for each target task m. This is followed by a scaled dot-product attention layer similar to the multi-head attention mechanism proposed in "Attention is All You Need"[205] that uses v_m^t vectors and the initial classification scores S_m from task m's siblings to learn a set of importance weights α_m^t for each timestamp t. Finally, a final feature vector is obtained as $v_m = \sum_t^T \alpha_m^t v_m^t$, where T refers to total number of timestamps. Please note that this task-specific architecture uses the initial predictions of siblings and the original EHR embeddings to capture new information from the most important window segments during a patient's hospitalization to fine-tune the initial predictions. This can be formulated as:

$$\begin{aligned} \hat{f_m} &= sigmoid(f_m^{hmtl} + \Delta f_m) \\ \Delta f_m &= W_s V_m + b_s \\ V_m &= attn(W_q S_m, W_k V_m^T, V_m^T) \end{aligned} \tag{28}$$

while:

$$attn(Q, K, V) = softmax(\frac{QK_T}{\sqrt{h}})V \tag{29}$$

where K, Q, V refer to key, query and value of the attention module. As shown in Figure 25, the key is set as the EHR embedding of each timestamp T while Q contains the original prediction scores for each target task. Therefore, the attention module is learning a attention weight α_t for each timestamp t by comparing the target task scores S as query Q which each

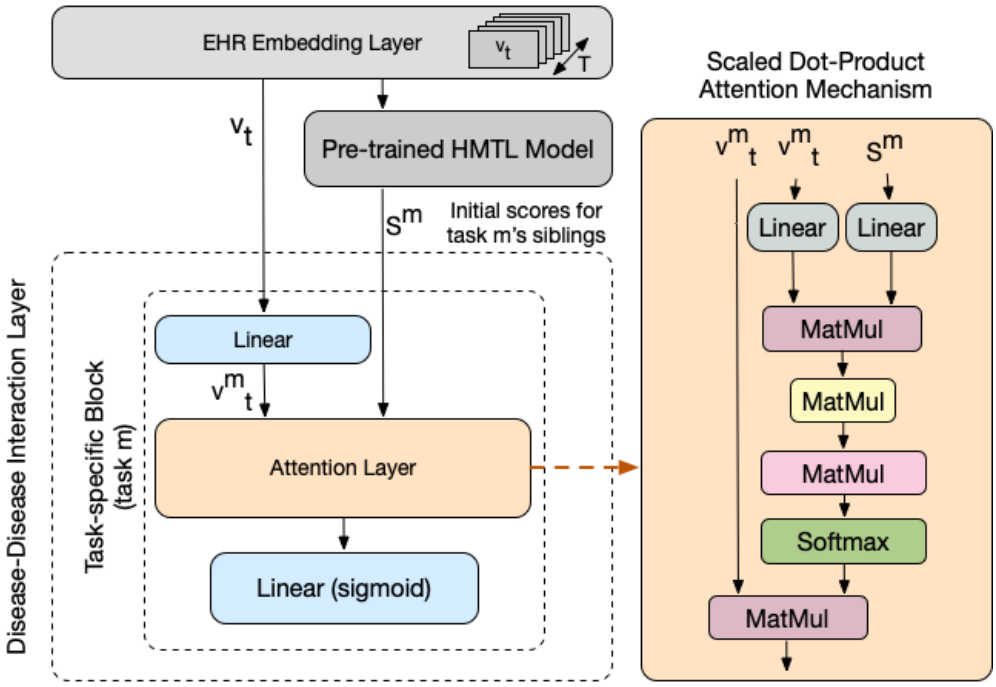


Figure 25: Task specific interaction layer

timestamp's EHR embedding u^t allowing the task-specific blocks to attend to critical time during patient's visit that could help determine the final target task prediction score $\hat{f^m}$.

5.1.4 Experiments

In this section we provide comprehensive evaluation of our proposed method and compare it with deep neural network baselines and methods previously proposed in chapters 2.2.3.2 and 2.3. The experiment setup in this chapter follows the same configurations of MIMIC-III and NOMA datasets and are explained in Section 3.6.2.

Implementation Details: The proposed HD-MTFL architecture was implemented with a linear embedding layer of dimension 512 and the task specific bi-LSTM used a hidden state of size 32. For evaluation, we adopted the weighted area under the receiver operating curve (AUROC) and the area under the precision-recall curve (AUPRC), which is suggested to be more suitable when using the average of the metrics across multiple imbalanced target tasks

with varying skewedness[168]. Finally, a random split of (70%/30%) the data was generated to create train and test sets.

Overall performance: We compared the overall performance of our proposed method with baselines including: (1) independently trained SVM models using studied in Chapter 3, (2) HA-MTL method proposed in Chapter 4, and (3) The multi-label LSTM architecture proposed in Section 3.6.2 which included a bidirectional LSTM layer, followed by a max-pooling layer to summarize patient's features across different timestamps. The latter allows knowledge transfer between all target tasks through a shared feature layer. However, the first two baselines rely on the SVD-based unsupervised lower-dimensional representations as explained in Chapter 3. Finally, we use average binary cross-entropy loss to train all deep neural network architectures.

Our empirical results show that HD-MTFL method results in strong improvements across both categorical and leaf target tasks, while the majority of this improvement can be attributed to the top-down hierarchical transfer of features. These improvements are consistent among both categorical and low-level leaves (low-prior and imbalanced), showing that the proposed method was able to transfer information top-down in an effective manner. Additionally, it shows that their model performance was more consistent across AUROC and AUPRC in the NOMA dataset, which we believe can be explained by the larger size of samples and a broader range of clinical variables resulting in more expressive representations. Therefore, the models could better model the commonalities and differences between related target tasks.

Task level analysis: While the overall results show strong improvements across all diagnoses and diagnostic categories ($M = 1228$), it's still valuable to evaluate the performance of the model across individual tasks. Figure 26 shows improvements in the individual target diagnostic tasks with respect to both weighted AUROC and weighted AUPRC metrics. For example in MIMIC-III, our proposed method resulted in considerable improvements ($\Delta > 0.05$) of nearly 50% of target tasks while preventing negative transfer with more 91% of classifiers performing at least as good as the baseline models($\Delta \geq 0$). In fact, only a handful of very rare diagnoses(2% of $0.004 \geq prior < 0.01$ group) demonstrated considerably lower performance than the baseline models ($\Delta < -0.05$). While a perfect MTL

Table 12: Comparison of overall performance of the HD-MTFL method method with baselines (average AUROC and AUPRC). Methods with a $\star$ sign were found to be statistically better than their baselines as outlined in Appendix .

		All Nodes		Category Nodes		Leaf Nodes	
Dataset	**Method**	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
MIMIC-III	SVD + SVM	0.759	0.162	0.756	0.215	0.761	0.118
	SVD + HA-MTL	0.779	0.173	0.778	0.230	0.781	0.126
	Multilabel BiLSTM + MP	0.801	0.160	0.794	0.216	0.807	0.114
	HD-MTFL wo Interaction layer*	0.809	0.197	0.801	0.251	0.818	0.145
	HD-MTFL*	0.813	0.201	0.808	0.262	0.816	0.151
NOMA	SVD + SVM	0.778	0.134	0.78	0.191	0.775	0.096
	SVD + HA-MTL	0.794	0.149	0.793	0.20	0.795	0.105
	Multilabel BiLSTM + MP	0.828	0.141	0.821	0.201	0.832	0.102
	HD-MTFL wo Interaction layer*	0.842	0.165	0.835	0.229	0.848	0.122
	HD-MTFL*	0.849	0.171	0.84	0.233	0.854	0.128

method is expected to only result in positive improvements, this has proven difficult in practice, especially when facing a large number of target tasks [228]. We conjecture that the negative improvements are mainly due to the imperfect hierarchy designs caused by residual categories that include diagnoses not closely aligned with other diseases. This motivates research and development of future HMTL methods that simultaneously learn to improve the existing hierarchies for machine learning tasks.

Comparison with Parameter Transfer Methods Figure 27 compares the performance of the HD-MTFL (blue) learning method with the HA-MTL (grey) methods proposed in Chapter 4 with respect to the percentage of the positive transfer and negative transfer in different task prior groups. A cutoff threshold of 0.02 AUPRC was used to capture the relative significant changes. Finally, the HD-MTFL learning improvements were compared to both the SVM baseline and the BiLSTM baseline (with max-pooling) models to enable both one-to-one comparison with HA-MTL and comparison of enhancements with the closest baseline in methodology (we will refer to this as respective baseline). However, the hierarchi-

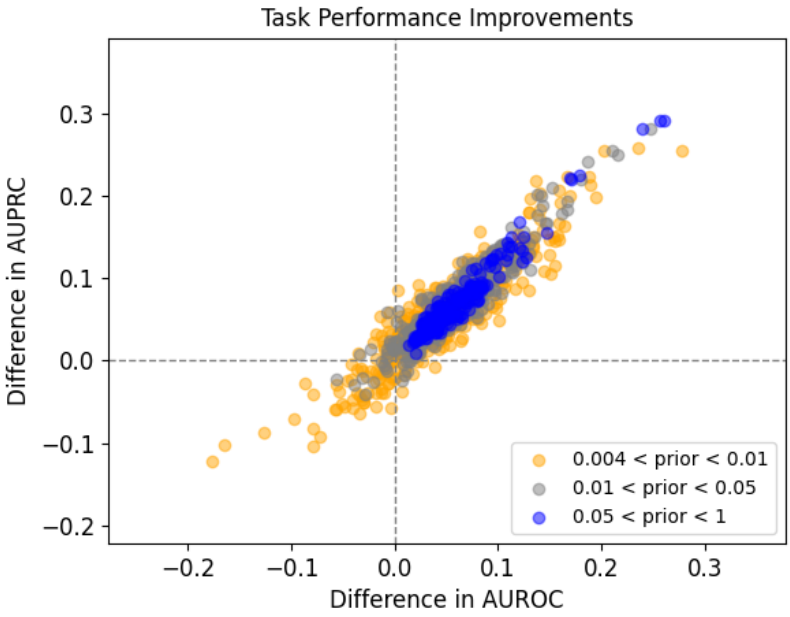
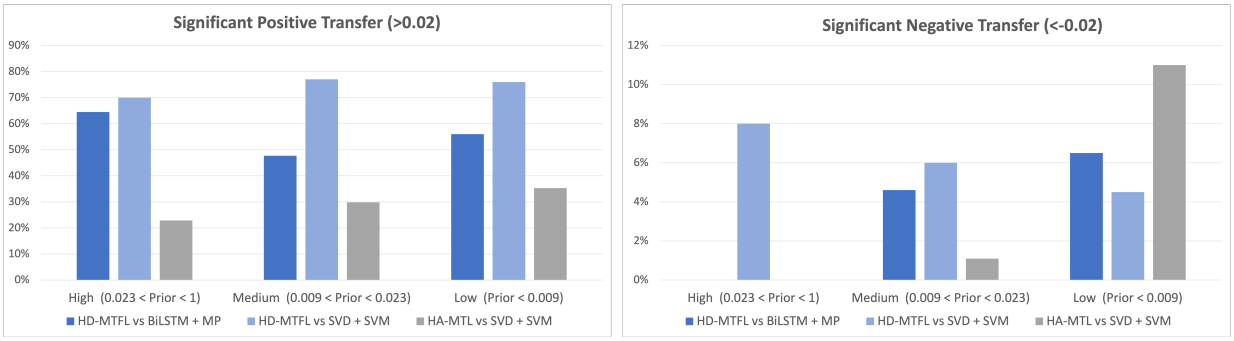


Figure 26: Performance improvements of individual tasks compared to the baseline multi-label LSTM models

cal adaptive multi-task learning method from Chapter 4 was only compared to the baseline methods since we believed comparison with the BiLSTM models would be inappropriate due to the RNN based models capability for capturing temporal patterns in patient's EHR data.

The results show that both models were robust in comparison to their respective baselines leading in no negative transfers. However, the HD-MTFL learning method was significantly more robust when comparing the performance of the target tasks with significantly low priors resulting in around 40% fewer negative transfers(Figure 27b). In contrast, the HA-MTL method resulted in fewer negative transfers in target tasks with medium-level priors. We argue that this is because of HA-MTL's direct capability of regularizing the impact of the parameter transfer when it did not result in better performance. However, this functionality did not perform as well in low prior tasks since it heavily relied on fine-tuning a hyper-parameter during the internal cross-validation, and low prior tasks could result

in biased hyper-parameter decisions. Finally, when comparing the performance of the two methods concerning positive transfer (Figure 27a), HD-MTFL consistently outperformed the HA-MTL method throughout any prior groups which were confirmed by the overall results reported earlier in Table 12.



(a) Positive Transfer (> 0.02 AUPRC) (b) Negative Transfer (< -0.02 AUPRC)

Figure 27: Comparison of the percentage of positive and negative transfers across target tasks in the MIMIC-III dataset. The groups were defined to lead to equal frequency bins with respect to task priors.

5.2 Summary and Shortcomings

In this chapter, we proposed a novel deep hierarchical multi-task learning method (HD-MTFL) framework that adopted feature transfer as the primary way of knowledge sharing between target task machine learning models. Our proposed method guided the transfer of features in a top-down fashion allowing each target task model to either adopt or combine features learned by its parent task (shared feature representation among all of its siblings) with a new set of task-specific features learned separately from the group.

Additionally, the proposed method improved upon the previous methods proposed in

Chapter 4 by mediating the impact of target tasks with a small sample size. This was achieved by utilizing a simultaneous learning algorithm as opposed to the iterative process adopted by HA-MTL and the Class-dependent HA-MTL, which showed promising results in preventing overfitting especially for target tasks with small number of positive samples.

However, there a number of shortcomings that could be addressed in future work. First, the proposed method uses a two step training algorithm to incorporate both parent-child and sibling relationships. This is while an end-to-end solutions could optimize the benefits of incorporating both types of relationships. Another shortcoming of the proposed method is its limited scalability since the number of model parameters linearly increase with the number of target machine learning task. This can render the proposed method useless when facing problems with much larger number of target tasks.

6.0 Modeling Patient Medication Orders using Electronic Health Records

In earlier chapters, we proposed and evaluated of performance of our proposed hierarchical multi-task learning methods for the classification of patient diagnoses and diagnostic categories. In this chapter, we aim to broaden the evaluation of our methods in the context of a prediction of future patient medication orders as a new interest of the application of HMTL methods [71, 29]. Learning accurate medication order prediction models can facilitate the development of life-saving clinical decision support solutions that can change a patient's healthcare experience. For instance, Hauskrecht et al. proposed an outlier detection framework that leveraged probabilistic predictive models for future medication orders to alert clinicians about potential missing medication orders in real-time. If accurate predictive models can be trained, the proposed solution can help reduce medication errors by providing a continuous monitoring and alerting solution.

However, learning accurate machine learning models for the prediction of patient medication orders face a number of important challenges similar to the diagnoses classification problem, including low sample size and presence of rare medication. These challenges motivate the adoption of hierarchical multi-task learning methods that aim to learn improved predictive models by facilitating knowledge transfer between hierarchically organized target tasks.

In the rest of this chapter, we first review the existing standard medication hierarchies that could be used in HMTL methods. Next, we provide a detailed description of the problem and explore new unique challenges introduced in the prediction of medication orders. In Section 6.4, we will explain the methodology used to adapt the approaches proposed in earlier chapters to model patients' medication orders and provide an extensive evaluation of our methods using the NOMA dataset in Section 6.6. Finally, we conclude this chapter by presenting a detailed analysis and discussions regarding the performance of our methods and shortcomings that will need addressing to address unique challenges in medication hierarchies better.

6.1 Standard Medication Hierarchies

The early standardized classifications of medications were originally developed as a tool for the pharmaceutical industry to classify and register their products. However, these classifications are today used to facilitate the development of healthcare applications and to allow research and comparison of consumption patterns of various medications across different populations. In addition, drug classifications are also developed and adopted to capture information related to drug-drug interactions, side effects, and relationships that physicians should consider when prescribing new treatments [150].

One of the most widely used medication hierarchies is the RxNorm hierarchy. the RxNorm hierarchy is a US-specific terminology system that contains all medications available in the US. It was originally proposed as a part of Unified Medical Language System (UMLS) terminology [116]. RxNorm categorizes medications based on their active ingredients and dosage. Therefore, the RxNorm classification uses a lattice structure in which each medication can be associated with multiple parent categories that represent an individual active ingredient.

Other drug classifications group medications in different ways. For example, the Therapeutic Chemical (ATC) classification system was developed by the World Health Organization (WHO). ATC is a classification system that categorizes medications according to their target organs, their therapeutic nature, and chemical ingredients. Figure 28 shows a small subset of the ATC hierarchy.

Figure 28: A subset of the anatomical therapeutic chemical (ATC) hierarchy

The U.S. Department of Veteran Affairs has also developed its own medical classification hierarchy called MED-RT. MED-RT contains a lattice-like classification that creates drug hierarchies according to multiple types of drug-drug relationships such as mechanisms of action (MoA), physiologic effects (PE), therapeutic categories (TC), or associated diseases [22]. Another popular drug hierarchy is SNOMED CT which also groups medications into hierarchical structures based on their chemical ingredients [47]. However, SNOMED CT also contains information about drug interactions structured as a graph.

While modeling patient medication orders has been studied in the past, to the extent

of our knowledge, the benefits of incorporating the hierarchical structure of patient medications in the learning process have not yet been studied. This is perhaps due to multiple existing challenges: First, large datasets with accurate medication order timestamps are not readily available. Second, incorporating medication hierarchies in the learning process requires access to an accurate mapping of EHR-specific codes to existing standard medication ontologies. However, such mappings are usually unavailable in public datasets such as MIMIC-III [87] and creating such mappings is not straightforward. In this work, we leverage an existing large retrospective electronic health record data extraction from the University of Pittsburgh's Medical Center, for which code mappings to standard medication hierarchies were previously developed.

6.2 Problem Definition

Medication orders in a patient's clinical care plans represent the physician's intention of administering the ordered medication to a patient immediately or in the future. This is different from a patient's medication administration records which will outline the specifics of administration for a particular medication order throughout a patient's hospital stay. In electronic health records, a medication order is usually represented with a start and end time and contains detailed information related to the what and hows of administration of a drug. This information can include administration frequency, dosage, route, etc. The start and end of the order represent the intended duration of administration. For instance, an order that requires a continuous serum infusion for 12 hours will have a start time as requested by the physician and end within 12 hours of the start time. In open-ended medication orders, the end time will remain open until the clinician has decided to discontinue or cancel the order.

Our goal is to learn machine learning models that can predict physicians' intention to order certain medication during a patient's hospitalization using available data prior to the prediction time. In other words, we aim to model physician decision-making criteria according to the patient's current clinical conditions. To this end, we follow the notation

in Section 3.5.1, to segment a patient's EHR data into equal-length segments that capture the clinical data recorded within that time segment. Next, each segment is linked to a set of future patient medication orders according to a predefined prediction window. Thus, we define target task labels $Y_i = y_i^1, y_i^2, ..., y_i^T$ in which $y_i^t = 0, 1^M$ determines whether any of the target M tasks were present during the prediction window w_p after timestamp t of patient i's hospitalization. Our goal is to learn machine learning models that can accurately predict whether a target medication task will be ordered within a predefined prediction window w_p from a particular timestamp t. Thus, we only associate future medication orders that are started with the prediction window and exclude the orders that are the continuation of orders that previously existed.

Figure 29 illustrates this process. The patient's clinical data before timestamp t is divided into multiple EHR segments and is linked with future medication orders. The segmentation of patients' EHR data before the prediction time creates four equal-length time segments, which will be featurized separately and used as the input to the machine learning models.

The figure also shows the associated medication orders that are observed within the predefined prediction window from timestamp t. However, here we only define a positive label when a future medication order is started after the current time, thus excluding the top medication order marked differently. From a clinical perspective, this design allows the systems to learn to alert physicians only when new medications are required while avoiding alerts that remind the clinical team about the treatment needs they might already be aware of. Additionally, this enables us to avoid training machine learning models that are learning to predict future orders by heavily depending on already existing orders for the same medication.

6.3 Unique Challenges in Modeling Medication Orders

Learning accurate machine learning models for patient medication hierarchies faces several critical challenges. First, extracting accurate labels for medication orders can be a complex problem since it relies on determining an optimum prediction window. However,

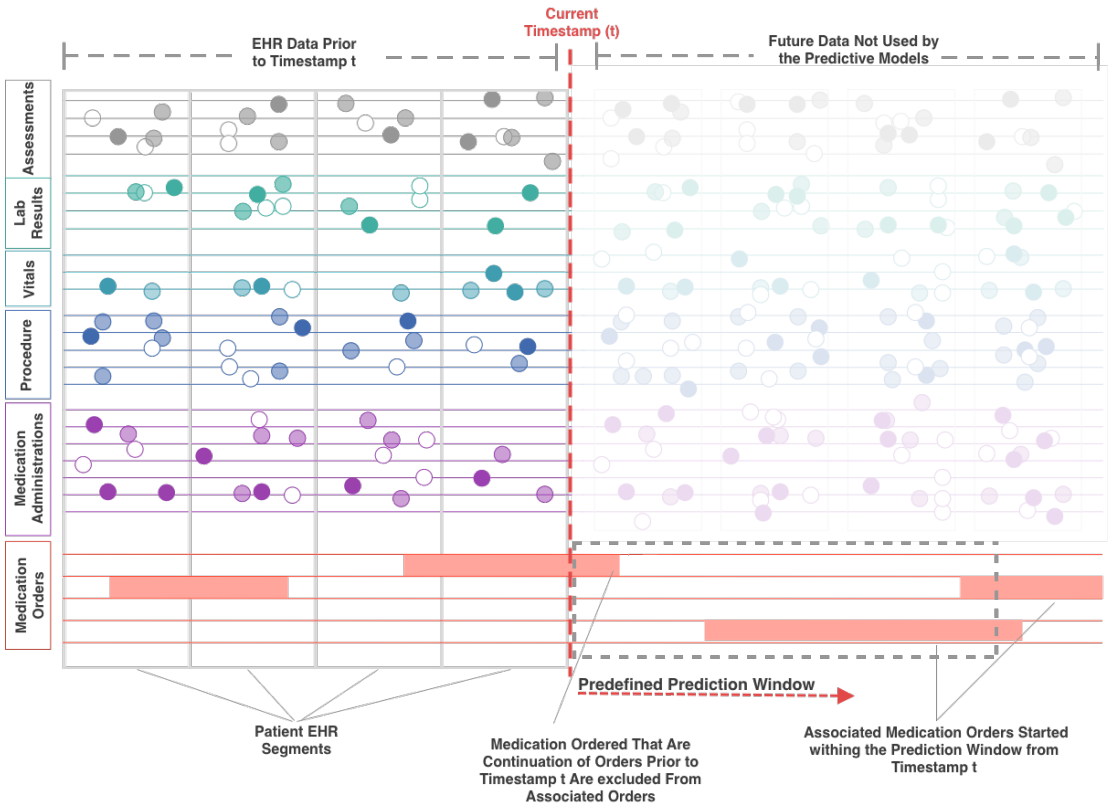


Figure 29: Visualization of the segmentation of patient's past EHR data prior to timestamp t and the associated future medication orders.

finding an optimum global prediction window may not be feasible. Therefore, the search for the ideal prediction window will remain a completely drug-specific process. Second, accurately modeling patient medication orders will require the incorporation of complex drug-drug indications. Drug-drug interactions (DDI) refer to unwanted chemical reactions between drugs that may decrease the impact of other medications or increase the chances of unexpected severe side effects. For example, simultaneous administration of pain medication such as Vicodin and a sedating antihistamine can result in significant feelings of drowsiness. Therefore, an accurate solution will need to capture such interactions. However, this can prove to be a complex problem as various drugs can interact with many other medications with different levels of importance. For instance, the Drug Bank dataset includes more than

12,000 DDIs between different medications[208].

Modeling patient medication orders also faces several new challenges that complicate the adoption of hierarchical multi-task learning methods. First, many common medications may have multiple intended uses. This can be because the FDA might have approved the medication for treating several different conditions. For example, Tadalafil marketed under the name Adcirca is generally used for treating pulmonary hypertension disorder, while the brand name Cialis is approved for other clinical reasons. However, both medications have Tadalafil as their active ingredient. In other cases, the route of administration may result in different reasons for administration. Nystatin, for example, when administered orally, is intended to be used to treat stomach yeast infections. However, nystatin is also prescribed as a topical medication that is typically used to treat fungal diseases. Such drugs can result in negative transfer when imposing similarities that may not necessarily exist. Therefore, to prevent such negative transfers, multi-task learning methods must be able to identify which use case was intended for a specific sample.

Another critical challenge related to medications with multiple intended uses is that some medications can be used alternatively. For instance, several statin medications such as Atorvastatin and Simvastatin are commonly used for lowering a patient's blood cholesterol levels. However, the choice of a specific statin order may not depend on the patient's conditions but be based entirely on pharmacy availability. In these cases, the target machine learning tasks representing alternative medication will not represent mutually exclusive related tasks. This can result in ill-defined labels since a negative label may not necessarily represent the lack of clinical need or the clinician's intention to order. Therefore, it can create challenges in adopting hierarchical multi-task learning methods since it will be hard to identify similarities and differences between sibling target medication tasks that are closely related.

6.4 Methodology

In this section, we aim to describe the adopted methodologies from the earlier chapters to model patients' future medication orders. We will particularly use two approaches: (1)

the multi-label LSTM based architecture initially introduced in Section 3.6.2 and the HD-MTFL framework proposed in Chapter 5 which demonstrated the most promising results compared to the other hierarchical multi-task learning methods proposed in this dissertation. In the earlier chapters, we formulated the problem of classifying patient diagnoses as a sequence classification problem where the general model architecture included (1) a shared lower-dimensional f learning layer, (2) a recurrent learning layer, and (3) a summarization block that combined the features at different timestamps to learn a final featurization from the entire patient visit. In contrast to the diagnoses classification problem, modeling a patient's future medication orders will be a prediction task that happens every timestamp. Therefore, we change the proposed architectures for the diagnoses problem by replacing the summarization block with the most recent patient context at a particular timestamp and using a standard LSTM layer instead of the bidirectional LSTM to respect the temporal causality of the time segments and outcomes. We follow a simple and intuitive approach that uses the last hidden states of the LSTM layer as the input to the predictive models at each timestamp. While other more complex methods proposed to creatively combine recent and past patient information could also be used, we believe a simple approach would be sufficient for evaluating hierarchical multi-task learning methods.

6.5 Implementation Details

The proposed HD-MTFL architecture used similar configuration described in Chapter 5. Briefly, the linear embedding layer dimension was set to 512, and the task-specific LSTM used a hidden state of size 32. Finally, since the medication prediction problem at each timestamp t is formulated as a binary prediction problem, we use a multi-task binary cross-entropy as the training loss function similar to the diagnoses classification problem in Section 3.6. In order to address the highly imbalanced prediction tasks, we used a cost-sensitive formulation of the cross-entropy loss in which the weight of positive samples was set to $\frac{Numberofnegativesamples}{Numberofpositivesamples}$. Finally, we optimized the neural network model using an AdamW optimization method with a learning rate 0.1.

6.6 Experiments and Discussions

The experiments in this section are performed using only the NOMA dataset since the medication mappings to standard ATC codes were not available in MIMIC-III. Similar to the diagnoses experiments in Section 3.6, we limited the target medications tasks to those that were ordered to at least 100 unique admissions. Additionally, we used a 1 hour EHR segmentation window 24-hour lookback window for feature generation, and a 3-hour prediction window size to determine associated medication orders with timestamp t. Finally, we report the results based on a random split of (70%/30%) of the data generated to create train and test splits. Table 13 shows the statistics regarding the size and score of the experiments. To evaluate the models, we adopted the area under the receiver operating curve (AUROC) and the area under the precision-recall curve (AUPRC).

Table 13: Basic information about each EHR dataset used in this study

Dataset	Admissions	Samples	Medication Tasks	Clinical Events
NOMA	43,788	3,583,200	276	8507

The results in Table 14 compare the performance of the deep hierarchical multitask feature learning method with the multi-label LSTM models. From an initial review of the overall performance comparison, it appears that the HD-MFTL method is only slightly better than the baseline approach. However, a deeper look into the performance improvements at the level of the individual tasks demonstrates that the models for some tasks gained significant improvements (Figure 30).

However, the lack of significant overall improvements can be explained by a large number of small negative transfers. We believe that one critical reason that resulted in the negative transfer was the extremely low priors of the target medication tasks. Figure 30 illustrates the model improvements concerning AUROC and AUPRC metrics across three groups of medi-

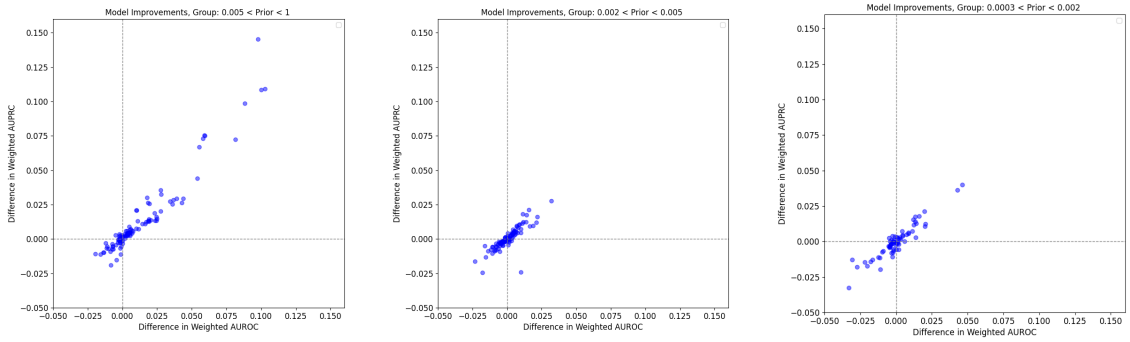
Table 14: Comparison of multi-label LSTM model with the HD-MFTL method proposed in Chapter 5. The overall results for the hierarchical model appears be similar to the baseline approach.

	All Nodes		**Category Nodes**		**Leaf Nodes**	
Method Name	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Multi-label LSTM	0.834	0.076	0.832	0.085	0.838	0.063
HD-MTFL	0.842	0.0859	0.846	0.099	0.837	0.068

cation tasks with different ranges of imbalance ratio. The results show that the HD-MTFL method resulted in considerable model performance improvements as high as 0.15 AUROC and 0.1 AUPRC across tasks in the higher prior group. In contrast, the target tasks in the low prior groups witnessed smaller improvements and significantly larger number of minor negative transfers. This suggests that while the HD-MTFL learning method proved to be more robust compared to the earlier iterative methods proposed in Chapter 4, it can still suffer from negative transfer when modeling extremely low prior target tasks.

6.7 Summary and Shortcomings

In this chapter, we studied the problem of modeling patients' medication orders and evaluated our proposed deep hierarchical multi-task feature learning method. Our results demonstrated that while our proposed HD-MTFL algorithm could significantly outperform the baseline model in certain medications, this improved performance was not consistent. We argue that these results can be explained by multiple challenges unique to the problem of predicting patient medication orders, such as the presence of alternative medication and



(a) Prior > 0.005 (b) $0.005 <$ Prior < 0.002 (c) Prior < 0.002

Figure 30: Model performance improvements for target medication tasks in three different prior groups

multiple drugs' intended uses. These challenges can result in falsely imposed similarities that may result in a negative transfer. Furthermore, the medication problem also includes target tasks with extremely low priors. Therefore, while the HD-MTFL proved to be more robust compared to the earlier iterative algorithms in Chapter 4 it appears to continue to be sensitive to extremely low priors.

7.0 Conclusion

In this dissertation, we explored the subject of hierarchical multi-task learning and its application in a number of novel problems in healthcare. The primary goal of this work was to introduce new methods that leverage the hierarchical task structures, various types of task relationships, and different forms of transfer of knowledge to facilitate the learning of improved machine learning models. However, the effective application of these methods to large-scale healthcare problems also depended on our capability to learn expressive features from patients' electronic health records. Therefore, we also studied and proposed techniques for learning lower-dimension representations from patients' data that could underlying patient conditions important for learning target task machine learning models. In the rest of this chapter, we will first review the main contributions in this dissertation. Next, we discuss the shortcomings and limitations of the methods proposed throughout this work. Finally, we conclude this dissertation by discussing open questions and promising future research directions in the field of hierarchical multi-task learning.

7.1 Contributions

In Chapter 3, we explored the ideas related to the first research goal of this dissertation. We hypothesized that high-dimensional multivariate time-series data stored in a patient's electronic health record can be represented with a smaller set of underlying components that explain the patient's conditions and information and that such representations could be used to learn target task machine learning models (published in [137]). To investigate this hypothesis, we proposed both unsupervised and supervised techniques that learn a lower-dimensional representation of patients' EHR from binary summarization of patients' clinical events. The unsupervised method used singular value decomposition to learn a dense, orthogonal, and lower-dimensional representation of patient's EHR summarizations. On the other hand, the supervised approach used a deep neural network architecture based on the

Recurrent Neural Network (RNN) with lower-dimension representation sufficient to support task predictions. Finally, we concluded this chapter by providing an extensive evaluation of our methods in the context of modeling the classification of patients' diagnoses and showed that the learned representation could be used for such learning tasks offering a comparable performance in comparison with state-of-the-art techniques that use patient's clinical notes
.

The second research goal in this work aimed to study and develop new hierarchical multi-task learning methods based on the parameter-based transfer of knowledge. We hypothesized that: (1) facilitating top-down and bottom-up parameter transfer could lead into the learning of improved classification models, and (2) related tasks' (models) prediction scores organized in expert-defined hierarchies do not have the same level of similarity among different classes of samples. To evaluate these hypotheses, we first proposed a novel hierarchical adaptive multi-task learning method (HA-MTL) that guided the transfer of model parameters in both top-down and bottom-up fashion using an iterative adaptive algorithm. The iterative algorithm trained each target task machine learning model in a step-wise fashion while guiding the transfer of model parameters by imposing similarities between parent and child task models. Additionally, the proposed method identified and excluded outlier tasks from their categories by simultaneously learning an asymmetric relatedness weight between parents and their children (published in [134]). Next, we further improved our proposed HA-MTL method by presenting a class-dependent version of the adaptation algorithm that dissects the transfer among the tasks based on positive and negative instances (published in [135]).

In Chapter 5, we investigated two main hypotheses related to the third and final research goal of in this dissertation. First, we hypothesized that the hierarchical structure of tasks can be used to guide transfer of knowledge in the form of feature-transfer. Second, accurate learning of target task models may also depend on capturing the sibling-sibling interactions. In order to probe this assumptions, we proposed a novel deep hierarchical multi-task feature learning (HD-MTFL) method that guided the transfer of features in a top-down fashion allowing each target task model to either adopt or combine features learned by its parent task (shared feature representation among all of its siblings) with a new set of task-specific features learned separately from the group. Second, we developed an interaction learning layer

following ideas in the field of differential diagnoses to capture the sibling-sibling interactions. The proposed method is trained separately and uses the initial predictions of siblings to find additional helpful information in patients' clinical data to further improve the target task predictions (published in [136]).

Finally, in Chapter 6, we evaluated our proposed hierarchical multi-task learning methods on the problem of modeling patients' future medication orders, our second application of methods in healthcare. Our experiments demonstrated that the proposed methods could help improve the target task machine learning models. However, we also identified new challenges faced when applying HMTL methods to the medication hierarchy that had not been addressed by the methods proposed in this dissertation. We will discuss these challenges in the following sections when we review the limitations of our solutions and open problems.

7.2 Limitations of the Methods

The methods and solutions proposed in this dissertation also faced a number of limitations:

Abstraction of Real-Values: Our proposed approach for learning feature representations of patients' electronic health records relied on obtaining a binary bag-of-word summarization of patients' clinical data. Although this approach offers multiple advantages, such as computational efficiency and the flexibility that allows it to be easily extended to new data types, it can result in the loss of critical information by discarding the detailed numerical values of patient's clinical data. For example, numerical values reflecting the dosage of vasopressors can determine the intended usage of the medication. When vasopressors are administered in high volumes, also known as a bolus, it can indicate treatment of hypotensive or septic shock. On the other hand, a low dose might suggest treatment for cardiovascular problems such as heart attack. One may argue that our binary event summaries could be easily expanded to cover specific circumstances. However, defining such comprehensive events would require extensive clinical knowledge and domain expertise. However, even if such expertise is readily available, extending these definitions for all types of clinical data in patients'

electronic health records may prove to be practically impossible task. Therefore, an ideal solution should be able to directly use the numerical values associated with patient's clinical data and capture such information automatically.

Discrete Time Segmentation and Bag-of-Word Summarization: In this thesis, we modeled the temporal information in patient's clinical data using time segmentation. That is, we segmented patient's hospitalization into equal width time windows in which the choice of the window size depended on the application. This approach allowed us to capture the temporal patterns in patient's clinical data using recurrent neural networks and lower-dimensional representation of patients' EHR data within each time segment as the input to the RNN network. However, this approach has a number of drawbacks. First, modeling patient's clinical data using discrete time windows results in loss of information related to the accurate timing of the events. For example in the context of predicting patient's future medication orders, if the segmentation window size is set to 12 hours, knowing that a particular medication was administered in the prior time window does not clearly specify whether this administration had happened at the beginning of the time segment or at the end. Knowing the exact time since the last medication administration might be critical for determining whether a new dosage might be necessary. One solution to this problem is to keep the time of last administration and compile the distributions of administration frequencies for all medication as proposed recently by Lee and Hauskrecht [106, 108]. In addition to losing information related to the accurate timing of events, the bag-of-words summarization of the clinical data within a time segment discards critical information related to the temporal context of information and focuses on the frequency of events. However, the order in which clinical events happen can be important in determining patient's underlying conditions. For example, its important to consider a medication order in the clinical context in which it was given to understand the treatment motivation and intended use of the drug. These shortcomings motivates study and research into feature learning techniques that can preserve such information.

Scalability of the Hierarchical Multi-task Learning Methods: Hierarchical multi-task learning methods studied in this thesis were designed to solve problems with many target tasks (ranging between 1000 - 3000). However, if the number of target tasks increases

significantly, it can introduce a new set of computational challenges. For example, the algorithms proposed in Chapter 2.2.3.2 rely on an iterative algorithm that trains target task models in a step-wise fashion. Therefore, if the number of target tasks is significantly larger, it can result in a very long training time. One way to solve this problem would be to adopt parallelization techniques that could scale the training processing to high-performance and highly scalable cloud infrastructures.

Similarly, the HD-MTFL learning method proposed in Chapter 5 can also face computation problems that might prove to be more challenging. Since the neural network architecture is designed according to the complexity and size of the target task hierarchy, when the number of tasks or the hierarchy complexity increases significantly, training the neural network model will face GPU memory limitations. Therefore, adopting the HD-MTFL methods for problems with an extremely large number of target tasks will require the design and development of strategies to distribute the computation to multiple GPU units.

7.3 Open Problems and Future Directions

7.3.1 Remaining Problems in Hierarchical Multi-task Learning

In Chapter 1, we reviewed several unique challenges that must be addressed before we can witness a wide adoption of hierarchical multi-task learning methods. In this dissertation, we attempted to address a number of these challenges, including imbalanced target tasks, small sample sizes, and outlier tasks in the context of both parameter-transfer and feature-transfer methods. However, others continue to require further research. Here we will briefly discuss the remaining challenges:

Alternative Tasks: One interesting challenge is how similar tasks interact with each other. One such interaction that can complicate the effective transfer of knowledge is alternative target tasks. Two similar tasks are considered alternatives when one can substitute another. While only one of these tasks can happen, the information required for determining the appropriate tasks may not be provided in the input data, nor consideration of such infor-

mation might be of interest. This is closely aligned with some of the challenges we faced in the medication hierarchy. For example, two medications can be considered alternative medications, meaning they could be administered for the exact same reasons, while the reason for choosing one might entirely be based on availability. Another way alternative tasks may create challenges is when one target task could be considered similar to multiple different groups of tasks that are unrelated to each other. Once again, in the context of the medication hierarchy, a drug might have multiple intended uses either by design or based on the route of administration. In such cases, the medication can belong to two multiple parent categories. This can result in falsely imposed similarities within each group in particular circumstances, thus resulting in negative transfer. This motivates the study and research of new hierarchical multi-task learning methods that can identify such circumstances and prevent negative transfer.

Heterogeneous Hierarchies: Another interesting remaining challenge is the presence of heterogeneous task hierarchies. In many domains, one can define various types of task relationships by considering different characteristics of tasks. As a result, multiple task hierarchies can be created. For instance, in the medication problem, task hierarchies can be created based on the medication's mechanism of action, intended use, or chemical compounds, each based on a different definition of similarity between target medication tasks. While this can represent a challenge in the effective application of HMTL methods, it also offers the opportunity to leverage additional sources of information in these heterogeneous hierarchies to facilitate the learning of more accurate machine learning models. Thus, such heterogeneous relationships between tasks prompt the research of hierarchical multi-task learning methods that can simultaneously use and combine multiple hierarchies to enhance knowledge transfer between target tasks. For example, one way multiple hierarchies could be incorporated into the proposed methods in this dissertation is by merging them. The results will be a lattice or graph in which each target task will be associated with multiple parent categories, each presenting either a completely disjoint or multiple overlapping sets of similar siblings. In feature-transfer based approaches such as the methods proposed in Chapter 5, these multiple groups can be combined by first learning separate feature representations for each parent group and then using an attention mechanism to allow the target task to focus

on the parent group that is most useful given the current context of the patient. Next, the target task can learn to combine the features from that group with new task-specific features that facilitate accurate prediction of each target task.

Optimization of Hierarchies: One possible source of negative transfer in hierarchical multi-task learning methods is the presence of imperfections in task hierarchies. Often, expert-defined hierarchies are designed to assist in visualizing or organizing medical concepts. Therefore, many such hierarchies may prove unsuitable or imperfect for use in machine learning methods as imposing similarities or guiding any form of knowledge transfer can negatively impact model performance due to a lack of sufficient similarities. One source of hierarchy imperfections is the presence of outlier tasks which are target tasks that are not equally similar to the other members of their parent groups. We tackled this challenge in Chapter 4 by learning similarity weights between parent and child target tasks. Later, in Chapter 5 we devised a deep neural network method that allowed each target task to decide to combine features from their parent with task-specific features learned separately with respect to the patient's clinical context.

In more complex circumstances, hierarchy imperfections can result from groups that are too general, requiring further categorization. Residual groups are another place where imperfections can appear. This is when we create categories that are not devoted to subclasses of related tasks but instead they include all those tasks that could not fit into other more specific groups. Moreover, hierarchies can have broader imperfections that may have been caused either due to error or because tasks were not organized for data analytics or machine learning. Therefore, an ideal hierarchical multitask learning algorithm should be able to train the target machine learning models while simultaneously attempting to optimize the task hierarchy.

Hierarchy imperfections can be addressed either using a two-step algorithm that first optimizes the hierarchy and then uses the improved hierarchy to learn the target task models in the second step. Alternatively, model learning and hierarchy optimization can simultaneously occur in a supervised fashion. Each of these methods can offer advantages while facing unique shortcomings. The former approach is advantageous since it can be more computationally efficient. However, improving the hierarchy will need to rely on unsupervised

optimization techniques that may lead to sub-optimal outcomes for the model learning step. Alternatively, the second approach is optimizing the hierarchy while learning the target task models, thus optimizing the machine learning objective function. While this approach theoretically can result in optimal solutions, the objective function will represent a non-convex problem, which might be hard to optimize and will not be guaranteed to find a global optimum answer. Therefore, studying and exploring new ideas to address these challenges can represent interesting future research topics.

7.3.2 Remaining Problems Related to Applications

Clinical Deployment: In this thesis, we demonstrated that hierarchical multi-task learning methods could be adopted to facilitate the training of improved machine learning models for critical healthcare problems: (1) automated classification of patient diagnoses and (2) prediction of future medication orders. However, additional research is needed to achieve the desired performance and expected clinical usability levels to encourage clinical deployment. Some of the important and potentially promising direction includes:

- **Learning Better EHR Featurization:** Earlier in this chapter, we reviewed a number of critical limitations of our proposed method for learning dense feature representations from patients' EHR data. Addressing such limitations can lead to learning more expressive and comprehensive feature representations from patients' clinical information that can thus lead to learning more accurate machine learning models. A number of promising research directions that can lead to considerable improvements include: (1) directly modeling the continuous and irregular timing of EHR data instead of relying on discrete time segmentation approach, (2) capturing the relationships between various EHR data elements, (3) combining EHR data with other modalities of patient's clinical data such as physiological waveforms measurements. Addressing such remaining challenges can result in the learning of improved machine learning models and thus facilitate clinical deployment of AI and machine learning in real-world settings.
- **Clinical Usability:** Learning accurate machine learning models across diagnostic and medication order tasks is not the only requirement for clinical adoption. In fact, effective

clinical workflow integration will also rely on the development of intuitive and meaningful software and clinical decision support solutions. [35, 5]. An important critique of early software solutions developed for healthcare is that major clinical workflow integration, clinician user experience, and alignment with clinician requirements at the point of care were not sufficiently studied in the user interface design. Therefore, it has been discussed and demonstrated that existing user experience design failures have contributed to critical problems such as alarm fatigue and clinician burnout [93]. This has resulted in a major shift of focus and expectation requiring the design and development of new clinical decision support tools to properly study and evaluate the clinical usability of such solutions in real-world settings [79]. Another closely related topic that has gained significant importance is explainability, which would ensure that physicians can easily interpret model predictions and recommendations and independently derive appropriate conclusions [198, 35, 62].

- **Fairness:** Another critical issue for adoption is model fairness. With the widespread use of AI and machine learning solutions in solving real-world problems, accounting for fairness has gained significant importance [133, 26]. However, in many applications, AI solutions have shown to be biased or under-perform for certain sub-populations, resulting in unfair bias [141] which can be due to either model development or bias in the training data[6]. However, the cost of unfair bias models can be relatively high in healthcare since it can not only result in biased decision-making and undesired health outcomes but also result in loss of life in some cases. Therefore, adopting machine learning solutions in healthcare would require careful evaluation of such solutions with respect to sensitive populations and the development of methods that can mitigate potential bias in the context of multi-task and hierarchical multi-task problems.

Multimodal Clinical Data: This work modeled target machine learning tasks using the lower-dimension representation of patients' structured electronic health records. However, accurate modeling of many prediction and classification problems in medicine would depend on learning feature representations that can capture critical information from other sources of patients' clinical data. The various sources of clinical data are commonly referred to as different modalities and may include text (clinical notes taken during patient's hospital-

ization), medical imaging (MRI, X-Ray, etc.), and high-frequency biomedical waveforms (electrocardiogram, Pulse Oximeter Pleth, etc.) which provide complementary information about patient's conditions. For example, a patient's high-frequency cardiovascular waveforms, such as an electrocardiogram (ECG), can provide a detailed view of the patient's cardiovascular stability. On the other hand, physician's notes present additional detail related to potential physician concerns, intentions, and care plans that may not be present in structure data.

Therefore, to learn comprehensive representations from patients' clinical contexts, one must develop feature learning methods that combine various modalities of patient clinical data. However, this can prove challenging as learning expressive features from these data sources has been the subject of extensive research in the biomedical informatics community. Therefore, in the future, we plan to adopt and develop new feature learning techniques that can: (1) learn expressive feature representations from multiple data modalities and (2) combine multiple modalities using fusion techniques that can incorporate the temporal relationships between them.

One exciting data modality in patients' clinical data is cardiovascular waveforms such as ECG and PPG data. From a clinical perspective, accurate and reliable modeling of many machine learning problems, including prediction of cardiovascular diagnoses and hemodynamic instabilities, would not be possible without accurate modeling of these waveforms. This is because clinical data related to the heart and cardiovascular system are often infrequently recorded, while modeling prediction of patient diagnoses such as hypotension usually requires capturing rapid changes in patient conditions.

From a technical perspective, while past work has proposed preliminary solutions in this area that either use temporal and statistical feature extraction techniques (See Section 2.3) [69, 219], or a combination of 1D convolutional and recurrent neural networks [27, 80, 147, 165], many critical challenges in this area remain unanswered which can be the subject of future research. Therefore, we plan to explore ideas that not only can learn expressive feature representations from these waveforms but also can combine these features with irregularly sampled electronic health records while incorporating the temporal relationships between the two modalities.

Addressing Limitation of Discrete Time Segmentation Approach: In Section 7.2 we discussed how the limitation of the proposal EHR representation learning approach proposed in this dissertation could lead to the loss of important information related to the accurate timing of events when using time segmentation. Therefore, an important future research direction is to study and investigate new methods and ideas that consider the accurate temporal information of patient clinical data and can incorporate such properties in more comprehensive EHR representations. One approach to address this problem is adopting temporal point process models to avoid time segmentation completely[38, 117]. Point processes can model event sequences by representing events as points in continuous time and defining the probabilistic distribution of points in space. Therefore, instead of using time segmentation to discretize the prediction task, point process methods model the rate and patterns of occurrence of events using the intensity function of the underlying conditional distribution. In general, point process methods either directly model the relationship between past events and future events (regressive point process) or indirectly, in which the dependencies are modeled through a latent space (latent space point processes) and learn a probabilistic distribution over the possible intensity functions. The direct approach in the regressive point process approach results in multiple advantages since it is easy to apply and facilitates more interpretable models. However, it relies on a-priori knowledge about the intensity function. On the contrary, the latent-space point process approach provides a flexible and generalizable framework for modeling any problem while it's harder to explain model predictions. Although the adoption of point processes methods for the prediction of clinical time-series events have been studied in the past [120, 119], many challenges remain unsolved that are critical for solving the problems studied in this application, including modeling the hierarchical relationships between the target tasks. Another key challenge in the adoption of point process models is their limitations in high-dimensional problems, as learning distributions over possible intensity functions across high-dimensional input spaces can become challenging. Therefore, adopting point process models for applications studied in this dissertation would require further research in the context of high-dimension large-scale multi-task problems.

Learning from Soft Class Labels: A typical training of classification machine learning

models assumes that each data instance is assigned to just one class label. However, in many practical (clinical and other) problems, the assignment of a class label may not be obvious, and it may come with a great deal of uncertainty. For example, diagnosis of a patient by a clinician is not a straightforward process given the data and individual diagnoses may be associated with the uncertainty of whether the patient has a specific disease or not. In such a case, the learning of classification models can be often improved by explicitly incorporating class uncertainty into the model training process. Training of classification models with *soft label information* [151, 152, 153] that permits class label uncertainty to be folded into the model training has demonstrated improved learning of classification models, especially in cases when prior to the concept occurrence is very low. The soft label learning process may help to alleviate the label annotation cost and can be combined with other efficient annotation solutions such as active learning [211, 212, 213, 214]. We note that soft-label learning ideas are also closely related to model calibration methods [222, 154, 149, 148] that aim to predict correct proportions of class instances by the model, as well as, various recent label smoothing solutions [145].

Choice of Prediction Window in Modeling Future Medication Orders : In Chapter 6 we proposed a general framework for modeling patients' future medication orders organized in a hierarchy. However, learning accurate machine learning models for patient medications depends on extracting accurate labels for medication orders which can be a complex problem since it relies on determining an optimum prediction window. However, finding an optimum global prediction window may not be feasible since various medications will inherently require a different window size. However, this information is not readily available. ,

One way this problem can be addressed is to develop new approaches that can derive an optimum drug-specific prediction window from existing EHR data and clinical patterns. Alternatively, one can attempt to address this challenge by adopting soft class labels. In this approach, instead of finding an ideal prediction window for each medication, we can use a set of prediction windows representing the time sensitivity of a particular medication order. Thus, the model will simultaneously predict whether a medication will be needed within multiple future prediction windows in which the smaller window sizes will represent a more immediate need for that medication. Another approach is using label smoothing [146]. Label

smoothing replaces a one-hot binary label with a probability within the prediction window that starts from zero and gradually increases the probability score to one. Thus samples closer to the actual time of a medication order receive a higher probability score. This gradual increase in probability scores can take many mathematical forms, including linear or exponential functions. Finally, one can attempt to remove the prediction windows from the problem definition altogether. One standard approach is to model time series prediction problems using point processes which discussed in more details earlier in this Section.

Appendix Multi-task Statistical Test

Evaluating the performance of proposed techniques by comparing the average performance of multi-task learning methods with baselines across the complete set of target tasks can provide a straightforward and intuitive solution to evaluate the overall performance of methods. However, a critical shortcoming of this approach is that it fails to determine whether the overal results represent statistically significant improvements. Therefore, we adopt a pair-wise statistical significance testing approach based on the bootstrap technique to address this issue. The proposed method leverages random subsamples with replacements from the original test set to evaluate the consistency of the model performance improvements in comparison with a baseline (Figure 31). We first generate a set of random bootstrap samples with replacements from the original test set. Next, each method obtains a set of predictions for the bootstrap samples. Next, each model is then evaluated according to a set of desired metrics. Next, we examine the difference in average model performance across all tasks with respect to each desired metric and for each bootstrap sample as:

$$\Delta avg\_metric_{tasks}(p^{\star}, y^{\star}) = avg\_metric_{tasks}(p^{\star}_{m}, y^{\star}) - avg\_metric_{tasks}(p^{\star}_{b}, y^{\star}) \tag{30}$$

In which $(p^{\star}_{m}$,$y^{\star}$) and $(p^{*}_{b},\ y^{\star})$ correspond to the model predictions and target labels for the proposed method m and baseline b. Thus, the bootstrap method allows us to measure the average model improvements ($\Delta avg\_metric_{tasks}(p^{\star}, y^{\star})$) across all bootstrap samples and the lower-band and upper-band improvement within a certain confidence interval (i.e. 95%). In order to calculate the confidence intervals we use the percentile method that has been proven to have near accurate empirical estimates of lower and upper bound confidence intervals if number of bootstrap samples are sufficiently large [53]. Finally, we conjecture that model m is statistically significantly better than baseline b if the model consistently outperforms the baseline within an acceptable confidence interval. In other words, within such a confidence interval $\Delta avg\_metric_{tasks}(p^{\star}, y^{\star})$ should always remain a positive value. Algorithm 19 provides the details related to the steps involved in this method.

Algorithm 1 Pair-wise statistical test to compare model with baseline

Require: M: number of target tasks

Require: *metric_name*: the desired metric name (i.e. AUROC or AUPRC)

Require: *num_bootsrap*: number of boostrap samples

Require: Y: Target labels for the test set and all target tasks

Require: $scores_m$: Model scores for the proposed method as $\mathbb{R}^M$

Require: $scores_b$: Model scores for the proposed method as $\mathbb{R}^M$

Ensure: $scores_m$ and $scores_b$ corresponds to identical test samples

Ensure: $scores_m$ and $scores_b$ corresponds to same set of target tasks

$N \leftarrow$ number of samples in test set

$delta\_metric\_list \leftarrow$ initialize as empty list

for $b \leftarrow 1$ to $num\_bootsrap$ **do**

Draw a random sample x^* of size N with replacement from $\{1, ..., N\}$

$scores_m^* \leftarrow$ a random sub-sample of $scores_m$according to x^*

$scores_b^* \leftarrow$ a random sub-sample of $scores_b$ according to x^*

$Y^* \leftarrow$ The sub-sample of Y according to x^*

$avg\_metric_m \leftarrow$: Calculate average target tasks performances using metric

$avg\_metric_b \leftarrow$: Calculate average target tasks performances using metric

▷ # Calculate the difference between the average metric scores on the bootstrap sample

$diff\_metric \leftarrow avg\_metric_m - avg\_metric_b$

Append($delta\_metric\_list$, $diff\_metric$)

end for

▷ # Calculate bootstrap statistics

$bootstrap\_mean \leftarrow mean(delta\_metric\_list)$

$lb, up \leftarrow$ Calculate lower and upper bound percentiles for 95% confidence interval

▷ # Perform the test to verify if method m is statistically better than baseline b

$test\_flag \leftarrow$ True if the range [lb, up] is always positive which entails that method m consistently outperforms the baseline withing the 95% confidence interval otherwise False

return $test\_flag$, $bootstrap\_mean$, lb, ub

As discussed in Chapter 3, we adopt AUROC and AUPRC metrics to evaluate the model performance and use a 95% confidence interval to verify the statistical improvements in the proposed methods in this dissertation in comparison to the appropriate baseline. Also, as shown in Figure 31, random bootstrap samples are consistent across all pair-wise comparisons in our implementation.

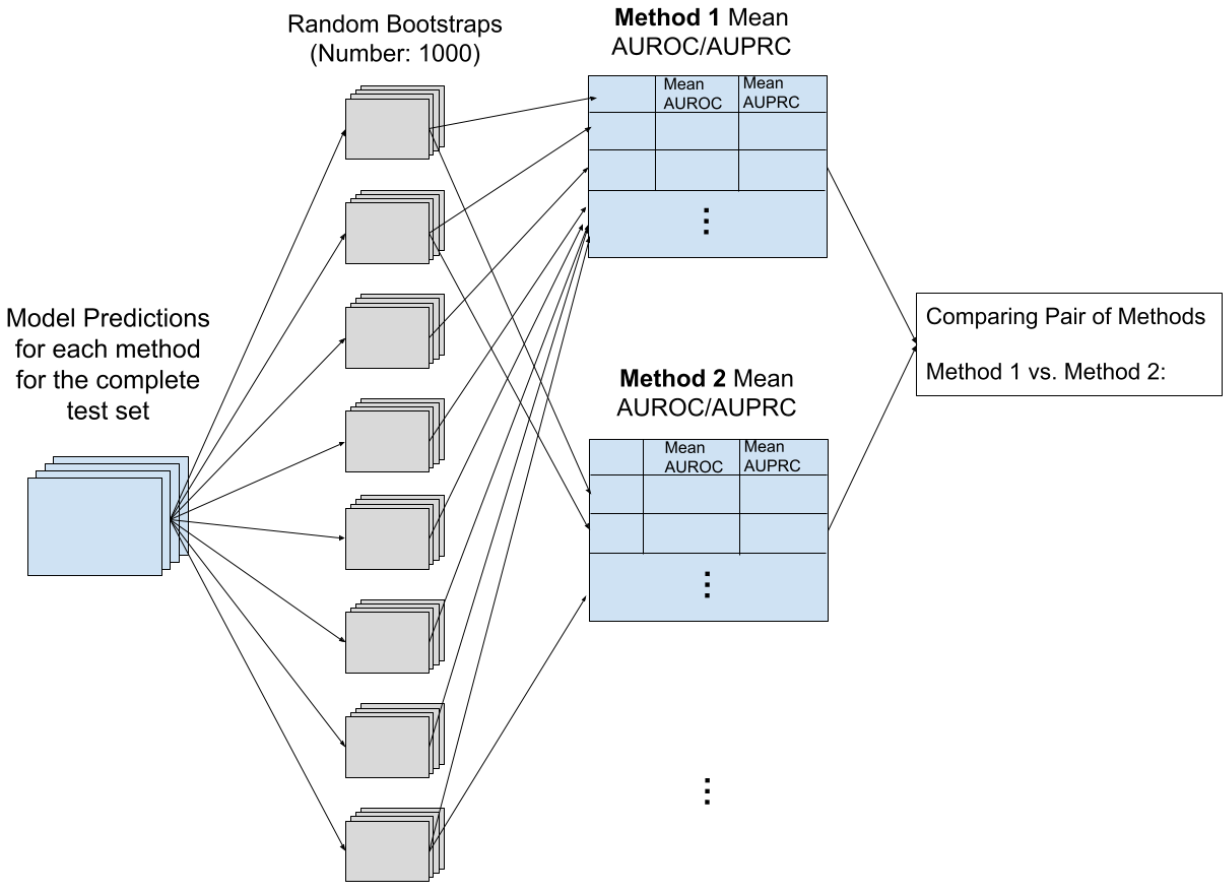


Figure 31: The pair-wise bootstrap based method to evaluate statistical significance of model improvements when comparing two methods

Table 15 demonstrates the results of the statistical test across all methods in this dissertation. The results show that, in general, each method outperforms in comparison with their most appropriate baseline with respect to both AUROC and AUPRC. However, the hierar-

chical deep multi-task learning method (HD-MTFL) proposed in Chapter 5 also outperforms the earlier methods discussed HA-MTL and Asymm HA-MTL discussed in Chapter 4.

Table 15: Bootstrap statistics calculated on MIMIC dataset for pairwise comparison of the models

			95% CI of $\Delta avg\_metric_{tasks}(p^\star, y^\star)$ (LB, Mean, UP)	
Dataset	**Method**	**Baseline**	AUROC	AUPRC
MIMIC-III	SVD + SVM	SVD + SVM	(0.014, 0.015, 0.016)	(0.0068, 0.007, 0.0087)
	SVD + AssymmHA-MTL	SVD + SVM	(0.013, 0.017, 0.020)	(0.008, 0.01, 0.012)
	SVD + AssymmHA-MTL	SVD + HA-MTL	(0.0033, 0.006, 0.0086)	(0.0013, 0.004, 0.007)
	HD-MTFL	SVD + SVM	(0.046, 0.049, 0.051)	(0.034, 0.036, 0.039)
	HD-MTFL	BiLSTM	(0.015, 0.017, 0.019)	(0.048, 0.051, 0.053)
	HD-MTFL	SVD + HA-MTL	(0.031, 0.034, 0.036)	(0.026, 0.029, 0.031)

Bibliography

[1] Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, 19:1–9, 2005.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pages 41–48, 2007.

[3] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] Mirza Mansoor Baig, Hamid GholamHosseini, Aasia A Moqeem, Farhaan Mirza, and Maria Lindén. A systematic review of wearable patient monitoring systems–current challenges and opportunities for clinical adoption. *Journal of medical systems*, 41(7):1–9, 2017.

[6] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. *Nips tutorial*, 1:2, 2017.

[7] Matthew Barren and Milos Hauskrecht. Improving prediction of low-prior clinical events with simultaneous general patient-state representation learning. In *International Conference on Artificial Intelligence in Medicine*, pages 479–490. Springer, 2021.

[8] Iyad Batal, Gregory Cooper, and Milos Hauskrecht. A bayesian scoring technique for mining predictive and non-spurious rules. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 260–276. Springer, 2012.

[9] Iyad Batal, Gregory F Cooper, Dmitriy Fradkin, James Harrison, Fabian Moerchen, and Milos Hauskrecht. An efficient pattern mining approach for event detection in multivariate temporal data. *Knowledge and information systems*, 46(1):115–150, 2016.

[10] Iyad Batal, Dmitriy Fradkin, James Harrison, Fabian Moerchen, and Milos Hauskrecht. Mining recent temporal patterns for event detection in multivariate time

series data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–288, 2012.

[11] Iyad Batal and Milos Hauskrecht. Constructing classification features using minimal predictive patterns. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 869–878, 2010.

[12] Iyad Batal, Charmgil Hong, and Milos Hauskrecht. An efficient probabilistic framework for multi-dimensional classification. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2417–2422. ACM, 2013.

[13] Iyad Batal, Hamed Valizadegan, Gregory F. Cooper, and Milos Hauskrecht. A pattern mining approach for classifying multivariate temporal data. In *Proceedings of the IEEE international conference on bioinformatics and biomedicine (BIBM)*, 2011.

[14] Iyad Batal, Hamed Valizadegan, Gregory F. Cooper, and Milos Hauskrecht. A Temporal Pattern Mining Approach for Classifying Electronic Health Record Data. *ACM Transaction on Intelligent Systems and Technology (ACM TIST), Special Issue on Health Informatics*, 2012.

[15] Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, pages 567–580. Springer, 2003.

[16] Steffen Bickel, Jasmina Bogojeska, Thomas Lengauer, and Tobias Scheffer. Multi-task learning for hiv therapy screening. In *Proceedings of the 25th international conference on Machine learning*, pages 56–63. ACM, 2008.

[17] Concha Bielza, Guangdi Li, and Pedro Larranaga. Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, 52(6):705–727, 2011.

[18] Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2008.

[19] Marc H Bornstein and Martha E Arterberry. The development of object categorization in young children: Hierarchical inclusiveness, age, perceptual attribute, and group versus individual analyses. *Developmental psychology*, 46(2):350, 2010.

[20] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[21] Geoffrey C Bowker and Susan Leigh Star. *Sorting things out: Classification and its consequences*. MIT press, 2000.

[22] Steven H Brown, Peter L Elkin, S Trent Rosenbloom, Casey S Husser, Brent A Bauer, Michael J Lincoln, John S Carter, Mark Erlbaum, and Mark S Tuttle. Va national drug file reference terminology: a cross-institutional content coverage study. *Medinfo*, 11(Pt 1):477–81, 2004.

[23] Róbert Busa-Fekete, Balázs Szörényi, Krzysztof Dembczynski, and Eyke Hüllermeier. Online f-measure optimization. In *Advances in Neural Information Processing Systems*, pages 595–603, 2015.

[24] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, 7(Jan):31–54, 2006.

[25] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

[26] Richard J Chen, Tiffany Y Chen, Jana Lipkova, Judy J Wang, Drew FK Williamson, Ming Y Lu, Sharifa Sahai, and Faisal Mahmood. Algorithm fairness in ai for medicine and healthcare. *arXiv preprint arXiv:2110.00603*, 2021.

[27] Yang Chen, Joo Heung Yoon, Michael R Pinsky, Ting Ma, and Gilles Clermont. Development of hemorrhage identification model using non-invasive vital signs. *Physiological measurement*, 41(5):055010, 2020.

[28] Yangchi Chen, Melba M Crawford, and Joydeep Ghosh. Integrating support vector machines in a hierarchical output space decomposition framework. In *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, volume 2, pages 949–952. IEEE, 2004.

[29] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318, 2016.

[30] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In

Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 787–795. ACM, 2017.

[31] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512, 2016.

[32] Edward Choi, Xiao, et al. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. *arXiv preprint arXiv:1810.09593*, 2018.

[33] Amanda Clare and Ross D King. Knowledge discovery in multi-label phenotype data. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–53. Springer, 2001.

[34] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[35] Christine M Cutillo, Karlie R Sharma, Luca Foschini, Shinjini Kundu, Maxine Mackintosh, and Kenneth D Mandl. Machine intelligence in healthcareperspectives on trustworthiness, explainability, usability, and transparency. *NPJ digital medicine*, 3(1):1–5, 2020.

[36] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive bayes classifiers for text classification. In *AAAI*, volume 7, pages 540–545, 2007.

[37] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007.

[38] Daryl J Daley, David Vere-Jones, et al. *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer, 2003.

[39] Hal Daumé III. Bayesian multitask learning with latent hierarchies. *arXiv preprint arXiv:0907.0783*, 2009.

[40] Jesse Davis and Pedro Domingos. Deep transfer via second-order markov logic. In *Proceedings of the 26th annual international conference on machine learning*, pages 217–224. ACM, 2009.

[41] C. DeCoro and Z. Barutcuoglu. Hierarchical shape classification using bayesian aggregation. In *IEEE International Conference on Shape Modeling and Applications 2006(SMI)*, volume 00, page 44, 06 2006.

[42] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[43] Ofer Dekel, Joseph Keshet, and Yoram Singer. Large margin hierarchical classification. In *Proceedings of the twenty-first international conference on Machine learning*, page 27. ACM, 2004.

[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[45] Thomas G Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, 2:263–286, 1994.

[46] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierarchical classification of diatom images using ensembles of predictive clustering trees. *Ecological Informatics*, 7(1):19–29, 2012.

[47] Kevin Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279, 2006.

[48] Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.

[49] Lixin Duan, Ivor W Tsang, Dong Xu, and Stephen J Maybank. Domain transfer svm for video concept detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1375–1381. IEEE, 2009.

[50] Lixin Duan, Dong Xu, Ivor Wai-Hung Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1667–1680, 2011.

[51] Susan Dumais and Hao Chen. Hierarchical classification of web content. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 256–263, New York, NY, USA, 2000. ACM.

[52] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)*, pages 845–850, 2015.

[53] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[54] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(Apr):615–637, 2005.

[55] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[56] Jianping Fan, Tianyi Zhao, Zhenzhong Kuang, Yu Zheng, Ji Zhang, Jun Yu, and Jinye Peng. Hd-mtl: Hierarchical deep multi-task learning for large-scale visual recognition. *IEEE transactions on image processing*, 26(4):1923–1938, 2017.

[57] Jianping Fan, Ning Zhou, Jinye Peng, and Ling Gao. Hierarchical learning of tree classifiers for large-scale plant species identification. *IEEE Transactions on Image Processing*, 24(11):4172–4184, 2015.

[58] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3205–3214, 2019.

[59] Liqiang Geng and Howard J Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9–es, 2006.

[60] Daniel George, Hongyu Shen, and EA Huerta. Deep transfer learning: A new deep learning glitch classification method for advanced ligo. *arXiv preprint arXiv:1706.07446*, 2017.

[61] Zoubin Ghahramani and Geoffrey E Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[62] Marzyeh Ghassemi, Luke Oakden-Rayner, and Andrew L Beam. The false hope of current approaches to explainable artificial intelligence in health care. *The Lancet Digital Health*, 3(11):e745–e750, 2021.

[63] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 22–30. Springer, 2004.

[64] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[65] Pinghua Gong, Jieping Ye, and Changshui Zhang. Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 895–903. ACM, 2012.

[66] Siddharth Gopal and Yiming Yang. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 257–265. ACM, 2013.

[67] James Douglas Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

[68] Lei Han and Yu Zhang. Learning tree structure in multi-task learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 397–406. ACM, 2015.

[69] Feras Hatib, Zhongping Jian, Sai Buddi, Christine Lee, Jos Settels, Karen Sibert, Joseph Rinehart, and Maxime Cannesson. Machine-learning algorithm to predict hypotension based on high-fidelity arterial pressure waveform analysis. *Anesthesiology*, 129(4):663–674, 2018.

[70] Milos Hauskrecht, Iyad Batal, Charmgil Hong, Quang Nguyen, Gregory F Cooper, Shyam Visweswaran, and Gilles Clermont. Outlier-based detection of unusual patient-management actions: an icu study. *Journal of biomedical informatics*, 64:211–221, 2016.

[71] Milos Hauskrecht, Iyad Batal, Michal Valko, Shyam Visweswaran, Gregory F Cooper, and Gilles Clermont. Outlier detection for patient monitoring and alerting. *Journal of biomedical informatics*, 46(1):47–55, 2013.

[72] Milos Hauskrecht, Michal Valko, Iyad Batal, Gilles Clermont, Shyam Visweswaram, and Gregory Cooper. Conditional Outlier Detection for Clinical Alerting. In *Proceedings of the American Medical Informatics Association (AMIA)*, November 2010.

[73] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[74] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[75] Joyce C Ho, Joydeep Ghosh, and Jimeng Sun. Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 115–124. ACM, 2014.

[76] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[77] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[78] Charmgil Hong. *Multivariate Data Modeling and Its Applications to Conditional Outlier Detection*. PhD thesis, University of Pittsburgh, 2018.

[79] Jan Horsky, Gordon D Schiff, Douglas Johnston, Lauren Mercincavage, Douglas Bell, and Blackford Middleton. Interface design principles for usable decision support: a targeted review of best practices for clinical prescribing interventions. *Journal of biomedical informatics*, 45(6):1202–1216, 2012.

[80] Borui Hou, Jianyong Yang, Pu Wang, and Ruqiang Yan. Lstm-based auto-encoder model for ecg arrhythmias classification. *IEEE Transactions on Instrumentation and Measurement*, 69(4):1232–1240, 2019.

[81] Daniel J Hsu, Sham M Kakade, John Langford, and Tong Zhang. Multi-label prediction via compressed sensing. In *Advances in neural information processing systems*, pages 772–780, 2009.

[82] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7304–7308. IEEE, 2013.

[83] Junzhou Huang. *Structured sparsity: theorems, algorithms and applications*. Citeseer, 2011.

[84] Oliver Ibe. *Markov processes for stochastic modeling*. Newnes, 2013.

[85] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. Clustered multi-task learning: A convex formulation. In *Advances in neural information processing systems*, pages 745–752, 2009.

[86] Xin Jin, Fuzhen Zhuang, Sinno Jialin Pan, Changying Du, Ping Luo, and Qing He. Heterogeneous multi-task semantic feature learning for classification. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1847–1850. ACM, 2015.

[87] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[88] Rudolf Emil Kalman. Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics*, 1(2):152–192, 1963.

[89] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *ICML*, volume 2, page 4, 2011.

[90] Alireza Karbalayghareh, Xiaoning Qian, and Edward R Dougherty. Optimal bayesian transfer learning. *IEEE Transactions on Signal Processing*, 66(14):3724–3739, 2018.

[91] Tsuyoshi Kato, Hisashi Kashima, Masashi Sugiyama, and Kiyoshi Asai. Multi-task learning via conic programming. In *Advances in Neural Information Processing Systems*, pages 737–744, 2008.

[92] Takayuki Katsuki, Masaki Ono, Akira Koseki, Michiharu Kudo, Kyoichi Haida, Jun Kuroda, Masaki Makino, Ryosuke Yanagiya, and Atsushi Suzuki. Risk prediction of diabetic nephropathy via interpretable feature extraction from ehr using convolutional autoencoder. In *MIE*, pages 106–110, 2018.

[93] Saif Khairat, Cameron Coleman, Paige Ottmar, Dipika Irene Jayachander, Thomas Bice, and Shannon S Carson. Association of electronic health record use with physician fatigue and efficiency. *JAMA network open*, 3(6):e207385–e207385, 2020.

[94] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, volume 2, page 1, 2010.

[95] Aniket Kittur, Ed H Chi, and Bongwon Suh. What's in wikipedia?: mapping topics and conflict using socially annotated category structure. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1509–1512. ACM, 2009.

[96] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 170–178, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[97] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[98] Shailesh Kumar, Joydeep Ghosh, and Melba M Crawford. Hierarchical fusion of multiple classifiers for hyperspectral data analysis. *Pattern Analysis & Applications*, 5(2):210–220, 2002.

[99] Lara K Kutschenko. In quest of goodmedical classification systems. *Medicine Studies*, 3(1):53–70, 2011.

[100] Günter Last and Mathew Penrose. *Lectures on the Poisson process*, volume 7. Cambridge University Press, 2017.

[101] Patrick J Laub, Thomas Taimre, and Philip K Pollett. Hawkes processes. *arXiv preprint arXiv:1507.02822*, 2015.

[102] Giwoong Lee, Eunho Yang, and Sung Hwang. Asymmetric multi-task learning based on task relatedness and loss. In *International conference on machine learning*, pages 230–238. PMLR, 2016.

[103] Hae Beom Lee, Eunho Yang, and Sung Ju Hwang. Deep asymmetric multi-task feature learning. In *International Conference on Machine Learning*, pages 2956–2964. PMLR, 2018.

[104] Jeong Min Lee and Milos Hauskrecht. Recent context-aware lstm for clinical event time-series prediction. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 13–23. Springer, 2019.

[105] Jeong Min Lee and Milos Hauskrecht. Clinical Event Time-series Modeling with Periodic Events. In *The Thirty-Third International Flairs Conference*. AAAI, 2020.

[106] Jeong Min Lee and Milos Hauskrecht. Clinical Event Time-series Modeling with Periodic Events. In *The Thirty-Third International Flairs Conference*. AAAI, 2020.

[107] Jeong Min Lee and Milos Hauskrecht. Multi-scale temporal memory for clinical event time-series prediction. In *Intl Conf on AI in Medicine (AIME)*, 2020.

[108] Jeong Min Lee and Milos Hauskrecht. Modeling multivariate clinical event time-series with recurrent temporal mechanisms. *Artificial Intelligence in Medicine*, February 2021.

[109] Jurica Levatić, Dragi Kocev, and Sašo Džeroski. The importance of the label hierarchy in hierarchical multi-label classification. *Journal of Intelligent Information Systems*, 45(2):247–271, 2015.

[110] Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):1–12, 2020.

[111] Carolyn E Lipscomb. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265, 2000.

[112] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint*, 2015.

[113] An-An Liu, Yu-Ting Su, Wei-Zhi Nie, and Mohan Kankanhalli. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):102–114, 2017.

[114] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 339–348. AUAI Press, 2009.

[115] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019.

[116] Simon Liu, Wei Ma, Robin Moore, Vikraman Ganesan, and Stuart Nelson. Rxnorm: prescription for electronic drug information exchange. *IT professional*, 7(5):17–23, 2005.

[117] Siqi Liu and Milos Hauskrecht. Nonparametric regressive point processes based on conditional gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1062–1072, 2019.

[118] Siqi Liu and Milos Hauskrecht. Event outlier detection in continuous time. In *Proceedings of the 38th International Conference on Machine Learning, PMLR 139*, 2021.

[119] Siqi Liu and Milos Hauskrecht. Event outlier detection in continuous time. In *International Conference on Machine Learning*, pages 6793–6803. PMLR, 2021.

[120] Siqi Liu, Adam Wright, and Milos Hauskrecht. Change-point detection method for clinical decision support system rule monitoring. *Artificial intelligence in medicine*, 91:49–56, 2018.

[121] Xiaobo Liu, Zhentao Liu, Guangjun Wang, Zhihua Cai, and Harry Zhang. Ensemble transfer learning algorithm. *IEEE Access*, 6:2389–2396, 2017.

[122] Zitao Liu and Milos Hauskrecht. Clinical time series prediction: Toward a hierarchical dynamical system framework. *Artificial Intelligence in Medicine*, 65(1):5–18, 2015.

[123] Zitao Liu and Milos Hauskrecht. A regularized linear dynamical system framework for multivariate time series analysis. In *The 29th AAAI Conference on Artificial Intelligence*, pages 1798–1804, 2015.

[124] Zitao Liu and Milos Hauskrecht. Learning adaptive forecasting models from irregularly sampled multivariate clinical data. In *The 30th AAAI Conference on Artificial Intelligence*, pages 1273–1279, 2016.

[125] Zitao Liu and Milos Hauskrecht. Learning linear dynamical systems from multivariate time series: A matrix factorization based framework. In *SIAM International Conference on Data Mining*, 2016.

[126] Zitao Liu and Milos Hauskrecht. A personalized predictive framework for multivariate clinical time series via adaptive model selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1169–1177. ACM, 2017.

[127] Zitao Liu, Lei Wu, and Milos Hauskrecht. Modeling clinical time series using gaussian process sequences. In *SIAM International Conference on Data Mining*, 2013.

[128] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[129] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. Learning multiple tasks with multilinear relationship networks. *Advances in neural information processing systems*, 30, 2017.

[130] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

[131] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017.

[132] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5334–5343, 2017.

[133] Samual MacDonald, Kaiah Steven, and Maciej Trzaskowski. Interpretable ai in healthcare: Enhancing fairness, safety, and trust. In *Artificial Intelligence in Medicine*, pages 241–258. Springer, 2022.

[134] Salim Malakouti and Milos Hauskrecht. Hierarchical adaptive multi-task learning framework for patient diagnoses and diagnostic category classification. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019.

[135] Salim Malakouti and Milos Hauskrecht. Not all samples are equal: Class dependent hierarchical multi-task learning for patient diagnosis classification. In *The Thirty-Third International Flairs Conference*, 2020.

[136] Salim Malakouti and Milos Hauskrecht. Hierarchical deep multi-task learning for classification of patient diagnoses. In *International Conference on Artificial Intelligence in Medicine*, pages 122–132. Springer, 2022.

[137] Seyedsalim Malakouti and Milos Hauskrecht. Predicting patients diagnoses and diagnostic categories from clinical-events in ehr data. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 125–130. Springer, 2019.

[138] Zvika Marx, Michael T Rosenstein, Leslie Pack Kaelbling, and Thomas G Dietterich. Transfer learning with an ensemble of background tasks. *Inductive Transfer*, 10, 2005.

[139] Andreas Maurer, Massi Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. In *International conference on machine learning*, pages 343–351, 2013.

[140] Craig McGarty. Social categorization. *Oxford Research Encyclopedia of Psychology*, 2018.

[141] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[142] Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, pages 6754–6764, 2017.

[143] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[144] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6:26094, 2016.

[145] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

[146] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[147] Fatma Murat, Ozal Yildirim, Muhammed Talo, Ulas Baran Baloglu, Yakup Demir, and U Rajendra Acharya. Application of deep learning techniques for heartbeats detection using ecg signals-analysis and review. *Computers in biology and medicine*, 120:103726, 2020.

[148] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Binary classifier calibration using a bayesian non-parametric approach. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 208–216, 2015.

[149] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, 2015.

[150] Stuart J Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. Normalized names for clinical drugs: Rxnorm at 6 years. *Journal of the American Medical Informatics Association*, 18(4):441–448, 2011.

[151] Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. Learning classification with auxiliary probabilistic information. In *2011 IEEE 11th International Conference on Data Mining*, pages 477–486. IEEE, 2011.

[152] Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. Sample-efficient learning with auxiliary class-label information. In *Proceedings of the Annual American Medical Informatics Association Symposium*, pages 1004–1012, 2011.

[153] Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. Learning classification models with soft-label information. *Journal of American Medical Informatics Association*, 2013.

[154] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632. Association for Computing Machinery (ACM), 2005.

[155] Guillaume Obozinski, Ben Taskar, and Michael Jordan. Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep*, 2(2.2), 2006.

[156] Guillaume Obozinski, Ben Taskar, and Michael I Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

[157] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[158] World Health Organization et al. International classification of diseases (icd). *http://www.who.int/classifications/icd/en/*, 2006.

[159] Serguei VS Pakhomov, James D Buntrock, and Christopher G Chute. Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. *Journal of the American Medical Informatics Association*, 13(5):516–525, 2006.

[160] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[161] David Pardoe and Peter Stone. Boosting for regression transfer. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 863–870. Omnipress, 2010.

[162] Jong-Bae Park, Yun-Won Jeong, Joong-Rin Shin, and Kwang Y Lee. An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Transactions on Power Systems*, 25(1):156–166, 2009.

[163] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[164] David N Perkins, Gavriel Salomon, et al. Transfer of learning. *International encyclopedia of education*, 2:6452–6457, 1992.

[165] Georgios Petmezas, Kostas Haris, Leandros Stefanopoulos, Vassilis Kilintzis, Andreas Tzavelis, John A Rogers, Aggelos K Katsaggelos, and Nicos Maglaveras. Automated atrial fibrillation detection using a hybrid cnn-lstm network on imbalanced ecg datasets. *Biomedical Signal Processing and Control*, 63:102194, 2021.

[166] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[167] Rajat Raina, Andrew Y Ng, and Daphne Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 713–720. ACM, 2006.

[168] Alvin Rajkomar and and others Oren. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):1–10, 2018.

[169] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[170] Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.

[171] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333, 2011.

[172] Lior Rokach and Oded Maimon. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):476–487, 2005.

[173] Eleanor Rosch. Cognitive representations of semantic categories. *Journal of experimental psychology: General*, 104(3):192, 1975.

[174] Eleanor Rosch, Carolyn B Mervis, Wayne Gray, David Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive psychology: Key readings*, 448, 2004.

[175] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005.

[176] Daniel M Roy and Leslie Pack Kaelbling. Efficient bayesian task-level transfer learning. In *IJCAI*, volume 7, pages 2599–2604, 2007.

[177] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4822–4829, 2019.

[178] Michael A Ruggiero, Dennis P Gordon, Thomas M Orrell, Nicolas Bailly, Thierry Bourgoin, Richard C Brusca, Thomas Cavalier-Smith, Michael D Guiry, and Paul M Kirk. A higher level classification of all living organisms. *PloS one*, 10(4):e0119248, 2015.

[179] Victor Sanh, Thomas Wolf, and Sebastian Ruder. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956, 2019.

[180] Dietmar Schabus, Marcin Skowron, and Martin Trapp. One million posts: A data set of german online discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1241–1244, Tokyo, Japan, August 2017.

[181] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[182] Konstantinos Sechidis et al. On the stratification of multi-label data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011.

[183] Carol A Seger and Earl K Miller. Category learning in the brain. *Annual review of neuroscience*, 33:203–219, 2010.

[184] Yuhui Shi and Russell C Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, pages 1945–1950. IEEE, 1999.

[185] Carlos Silla and Alex Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72, 01 2011.

[186] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72, 2011.

[187] Anima Singh, Girish Nadkarni, John Guttag, and Erwin Bottinger. Leveraging hierarchy in medical codes for predictive modeling. In *Proceedings of the 5th ACM conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 96–103. ACM, 2014.

[188] Vergil N Slee. The international classification of diseases: ninth revision (icd-9). *Annals of internal medicine*, 88(3):424–426, 1978.

[189] Edward E Smith and Douglas L Medin. *Categories and concepts*, volume 9. Harvard University Press Cambridge, MA, 1981.

[190] Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 521–528. IEEE, 2001.

[191] Charles Sutton and Andrew McCallum. Composition of conditional random fields for transfer learning. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 748–754. Association for Computational Linguistics, 2005.

[192] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[193] Farbound Tai and Hsuan-Tien Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

[194] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer, 2018.

[195] Martin Thoma. The reuters dataset, July 2017.

[196] Edward L Thorndike and Robert S Woodworth. The influence of improvement in one mental function upon the efficiency of other functions: Iii. functions involving attention, observation and discrimination. *Psychological Review*, 8(6):553, 1901.

[197] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[198] Sana Tonekaboni, Shalmali Joshi, Melissa D McCradden, and Anna Goldenberg. What clinicians want: contextualizing explainable machine learning for clinical end use. In *Machine learning for healthcare conference*, pages 359–380. PMLR, 2019.

[199] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global, 2010.

[200] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[201] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08)*, volume 21, pages 53–59. sn, 2008.

[202] G. Valentini. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(3):832–847, May 2011.

[203] Michal Valko and Milos Hauskrecht. Feature importance analysis for patient management decisions. *Studies in health technology and informatics*, 160(Pt 2):861, 2010.

[204] Linda C Van Der Gaag and Peter R De Waal. Multi-dimensional bayesian network classifiers, 2006.

[205] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[206] Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Machine learning*, 73(2):185, 2008.

[207] Mansfield Tracy Walsorth. *Twenty Questions: A Short Treatise on the Game to Which are Added a Code of Rules and Specimen Games for the Use of Beginners*. Holt, 1882.

[208] David S Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl_1):D901–D906, 2008.

[209] F Wu, J Zhang, and V Honavar. Learning classifiers using hierarchically structured class taxonomies. In *Lecture Notes in Computer Science*, volume 3607. Springer, Germany, 2005.

[210] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(Jan):35–63, 2007.

[211] Yanbing Xue and Milos Hauskrecht. Active learning of classification models with likert-scale feedback. In *SIAM International Conference on Data Mining (SDM)*, pages 28–35, 2017.

[212] Yanbing Xue and Milos Hauskrecht. Efficient learning of classification models from soft-label information by binning and ranking. In *Proceedings of the 30th International Florida AI Research Society Conference*, pages 164–169, 2017.

[213] Yanbing Xue and Milos Hauskrecht. Active learning of multi-class classifiers with auxiliary probabilistic information. In *Proceedings of the 31st International Florida AI Research Society Conference.*, 2018.

[214] Yanbing Xue and Milos Hauskrecht. Active learning of multi-class classification models from ordered class sets. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, 2019.

[215] Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 188–197. ACM, 2007.

[216] Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016.

[217] Yongxin Yang and Timothy M Hospedales. Trace norm regularised deep multi-task learning. *arXiv preprint arXiv:1606.04038*, 2016.

[218] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1855–1862. IEEE, 2010.

[219] JH Yoon, S Malakouti, M Hauskrecht, MR Pinsky, and G Clermont. Machine learning driven prediction of hypotension using real world multigranular data. In *B43. ICU MANAGEMENT*, pages A5615–A5615. American Thoracic Society, 2022.

[220] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[221] Ke Yu, Mingda Zhang, Tianyi Cui, and Milos Hauskrecht. Monitoring icu mortality risk with a long short-term memory recurrent neural network. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2020*, pages 103–114. World Scientific, 2019.

[222] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer, 2001.

[223] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[224] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[225] Lei Zhang. Transfer adaptation learning: A decade survey. *arXiv preprint arXiv:1903.04687*, 2019.

[226] Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 999–1008. ACM, 2010.

[227] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

[228] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. Overcoming negative transfer: A survey. *arXiv preprint arXiv:2009.00909*, 2020.

[229] Yu Zhang. Heterogeneous-neighborhood-based multi-task local learning algorithms. In *Advances in neural information processing systems*, pages 1896–1904, 2013.

[230] Yu Zhang and Qiang Yang. Learning sparse task relations in multi-task learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[231] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[232] Yu Zhang and Dit-Yan Yeung. Multi-task boosting by exploiting task relationships. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 697–710. Springer, 2012.

[233] Yu Zhang and Dit-Yan Yeung. Multilabel relationship learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(2):7, 2013.

[234] Xiangyun Zhao, Haoxiang Li, Xiaohui Shen, Xiaodan Liang, and Ying Wu. A modulation module for multi-task learning with applications in image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 401–416, 2018.

[235] Denny Zhou, Lin Xiao, and Mingrui Wu. Hierarchical classification via orthogonal transfer. In *International Conference on Machine Learning*, 2011.

[236] Qiang Zhou and Qi Zhao. Flexible clustered multi-task learning by learning representative tasks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):266–278, 2015.

[237] Alon Zweig and Daphna Weinshall. Hierarchical regularization cascade for joint learning. In *International Conference on Machine Learning*, pages 37–45, 2013.