University of California
Santa Barbara

AI and Big Data in Health: Boosting Reliability and Efficiency in Predictive Healthcare Models

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Yuqing Wang

Committee in charge:

Professor Linda Petzold, Chair
Professor John R. Gilbert
Professor Michael Beyeler

June 2023

The Dissertation of Yuqing Wang is approved.

Professor John R. Gilbert

Professor Michael Beyeler

Professor Linda Petzold, Committee Chair

June 2023

AI and Big Data in Health: Boosting Reliability and Efficiency in Predictive Healthcare Models

by

Yuqing Wang

To my dear mother,

who has guided me with wisdom and encouragement.

To my beloved grandparents,

who has instilled in me lessons of resilience and perseverance.

To my loving husband,

who has been my pillar of strength and support.

Acknowledgements

Embarking on this Ph.D. journey has been an enlightening adventure, and I'm profoundly appreciative of the exceptional individuals who have been pivotal in my academic progression. Foremost among them is my advisor, Prof. Linda Petzold. Her unparalleled expertise, understanding, and patience have formed the bedrock of my lab experience. Prof. Linda's unique style of mentorship, providing the perfect balance of autonomy and guidance, has empowered me to delve into areas of research that truly ignited my passion. Her unwavering belief in my abilities and the numerous opportunities she provided for me have profoundly shaped my academic and personal development. Her mentorship transcended the academic sphere, imparting crucial life lessons that I will treasure beyond the confines of this dissertation.

My heartfelt appreciation also extends to my committee members, Prof. John R. Gilbert and Prof. Michael Beyeler. Their expert guidance, insightful observations, and detailed feedback have been crucial in refining my research and enriching its academic depth. The opportunity to engage, interact, and collaborate with such esteemed scholars has been a privilege I deeply value.

Special thanks to Prof. Rachael A. Callcut for being really concerned and supportive during the collaboration. I am also grateful to my lab mates and friends. Their constant inspiration and unwavering support have truly enriched my PhD journey, making it not just an academic endeavor, but also a truly enjoyable and fulfilling experience.

Last, but certainly not least, I thank my cherished family, my enduring childhood friends, schoolmates, and former teachers. Their ceaseless encouragement and support during my tenure at UCSB have been my bedrock. Their faith in me has been unwavering, their love, boundless. Their contribution to this journey is immeasurable, as is my love and gratitude for them.

Curriculum Vitæ

Yuqing Wang

Education

2023	Ph.D. in Computer Science, University of California, Santa Barbara.
2020	B.S. in Mathematics, University of Minnesota, Twin Cities.

Research Interests

- Predictive modeling with the use of state-of-the-art deep learning models for accurate and efficient predictions of healthcare outcomes.
- Multimodal representation learning and fusion with biomedical data from different data modalities, such as clinical texts, longitudinal data, and medical images.
- Exploration of reinforcement learning-based approaches for the development of effective intervention strategies.
- Explainable clinical prediction models to make the decision-making process of AI systems transparent, which may mitigate the bias of AI algorithms.

Publications

- **Yuqing Wang**, Yun Zhao, and Linda Petzold. "Predicting the need for blood transfusion in intensive care units with reinforcement learning", in ACM-BCB 2022, Chicago, USA, Aug. 2022. (Recipient of the **Best Student Paper Award**)
- **Yuqing Wang**, Yun Zhao, and Linda Petzold. "Enhancing Transformer Efficiency for Multivariate Time Series Classification", in ICDM 2022, New York, USA, Jul. 2022.
- **Yuqing Wang***, Yun Zhao*, and Linda Petzold. "Integrating Physiological Time Series and Clinical Notes with Transformer for Early Prediction of Sepsis", in ICDM 2022, New York, USA, Jul. 2022.
- **Yuqing Wang***, Yun Zhao*, Junfeng Liu, Haotian Xia, Zhenni Xu, Qinghang Hong, Zhiyang Zhou, and Linda Petzold. "Empirical Quantitative Analysis of COVID-19 Forecasting Models", in DMBIH 2021, Auckland, New Zealand, Dec. 2021. (Recipient of the **Best Paper Award**)
- **Yuqing Wang***, Yun Zhao*, Rachael Callcut, and Linda Petzold. "Empirical Analysis of Machine Learning Configurations for Prediction of Multiple Organ Failure in Trauma Patients", in ICDM 2021, New York, USA, Jul. 2021.
- Yun Zhao, Qinghang Hong, Xinlu Zhang, Yu Deng, **Yuqing Wang**, and Linda Petzold. "BERTSurv: BERT-Based Survival Models for Predicting Outcomes of Trauma Patients", in ICDM 2021, New York, USA, Jul. 2021.

In Submission

- **Yuqing Wang**, Prashanth Vijayaraghavan, Ehsan Degan. "PM-Net: Prototype-based Multi-view Multi-branch Network for Interpretable Email Response Prediction".
- **Yuqing Wang**, Yun Zhao, and Linda Petzold. "An Empirical Study on the Robustness of the Segment Anything Model (SAM)".
- **Yuqing Wang**, Yun Zhao, and Linda Petzold. "Are Large Language Models Ready for Healthcare? A Comparative Study on Clinical Language Understanding".

Professional Experience

Summer 2022 Research Intern, IBM Research, Almaden, CA

- Developed a prototype-based multi-view multi-branch network for interpretable email response prediction. The proposed model provided explanations from multi-views: semantic perspective (BERT) and structural perspective (graph-based dependency parsing with GNN). Furthermore, it offered prototypes for explanations at the document/sentence/phrase levels.
- Performed experiments on two real-world email datasets and performance of the proposed model improved over the strongest baselines w.r.t. weighted average F1 score by 3.50% and 3.62% on the Enron corpus and IBM-SalesLoft corpus, respectively.
- Edited email contents based on prototypes over keywords / key phrases improved the overall email response ratio on the testing set by up to 1.9% and 3.8% on the Enron corpus and IBM-SalesLoft corpus, respectively.

Honors & Awards

- Best Student Paper Award, 13th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics 2022
- Best Paper Award, 9th Workshop on Data Mining in Biomedical Informatics and Healthcare 2021
- Academic Excellence Fellowship, UCSB *2020*
- Undergraduate Research Scholarship, UMN *2018*
- Maroon Global Excellence Scholarship, UMN *2016 - 2020*

Abstract

AI and Big Data in Health: Boosting Reliability and Efficiency in Predictive Healthcare Models

by

Yuqing Wang

In the era of data-driven decision-making, healthcare stands as a critical domain where machine learning (ML) techniques can bring transformative changes. However, the application of ML in healthcare faces unique challenges due to clinicians' limited understanding of intricate ML processes, the diverse and unstructured nature of healthcare data, high computational costs, and the "black box" problem associated with ML algorithms. The recent advent of large language models (LLMs) further introduces the challenge of developing appropriate prompts to guide these models to provide meaningful and contextually relevant responses.

This dissertation grapples with these challenges across a series of studies. First, we analyze multiple ML configurations for the prediction of multiple organ failure in trauma patients, highlighting the impact of classifier choice on performance. Next, we propose a multimodal Transformer model for early sepsis prediction, demonstrating its efficacy over competitive baselines. To address the computational costs, we propose an efficient model for multivariate time series classification. Reinforcement learning is then applied to predict the need for blood transfusion in intensive care units, offering a decision support tool for effective treatment recommendations. Lastly, we conduct a comparative study on the readiness of LLMs for healthcare, introducing a novel prompting strategy to maximize their effectiveness.

The primary objective of this dissertation is to facilitate the advancement, compre-

hensive evaluation, and systematic optimization of machine learning applications specifically in the healthcare domain. Our work aims to connect complex ML methodologies with practical healthcare applications. As our work progresses, we remain committed to the continuous refinement and enhancement of these models. Our approach aims to balance technical sophistication with ease of use, minimizing the trade-off between the two. We believe that our ML advancements, tailored to the unique needs of healthcare applications, can improve patient outcomes and streamline healthcare delivery.

Contents

Chapter 1

Introduction

Machine learning (ML) techniques have become increasingly prevalent across numerous domains over the past decade, with healthcare standing out as one of the most promising areas for their application. This prominence is largely due to the data-intensive nature of healthcare, which continuously amasses vast amounts of multifaceted data, including electronic health records (EHRs), clinical notes, and real-time physiological monitoring data [1]. However, the application of ML in healthcare is fraught with unique challenges.

A major issue is the lack of understanding among clinicians regarding the intricate ML processes, often leading to skepticism and hesitation in adopting these technologies. Consequently, there is a growing need for more interpretable and transparent ML applications in healthcare to bridge this gap. Another obstacle is the diverse and unstructured nature of healthcare data. Data originating from various sources often come in a multitude of formats, necessitating advanced ML techniques that can effectively manage such heterogeneity. Further complexities arise from the high computational cost associated with model training and the infamous "black box" problem of ML algorithms. This problem refers to the lack of clear, understandable explanations for the decisions made by ML models, which is a significant concern in healthcare, a field where interpretability

is crucial.

With the recent rise of large language models (LLMs), there is potential for enhanced clinical language understanding. These models, with their ability to process and generate human-like text, can significantly advance the use of unstructured textual data in healthcare. However, their deployment is not without challenges. One of the key issues lies in the development of appropriate prompts that can guide LLMs to produce meaningful and contextually relevant responses. Addressing these challenges is integral to harnessing the full potential of ML in the healthcare landscape.

In the following chapters of this dissertation, we delve into these pressing challenges, offering innovative solutions that harness the power of machine learning. From enhancing the interpretability of ML models, grappling with the complexities of diverse and unstructured healthcare data, to mitigating the computational demands of ML, developing effective treatment recommendations, and addressing the unique challenges associated with large language models in healthcare, each chapter encapsulates a distinct study with a common objective - to transform the application of ML in healthcare. We explore how the intricate interplay of machine learning techniques can be effectively employed to address these unique issues within the healthcare domain.

In Chapter 2, we investigate the critical issue of multiple organ failure (MOF), a life-threatening condition where early detection is pivotal [2]. To tackle this, we perform a comprehensive analysis of machine learning configurations, including data preprocessing, feature selection, classifier choice, and hyperparameter tuning. We find that classifier choice has the most impact on performance improvement and variation, underscoring the need for a careful balance between complexity and performance stability.

Chapter 3 focuses on sepsis, a leading cause of death in Intensive Care Units [3]. We propose a multimodal Transformer model for early sepsis prediction, using physiological time series data and clinical notes within the first 36 hours of ICU admission. By eval-

uating our model on two large critical care datasets, we demonstrate its effectiveness in early sepsis detection, outperforming several established baselines.

In Chapter 4, we address the balance between efficiency and accuracy in multivariate time series (MTS) classification algorithms [4]. We propose a methodology based on module-wise pruning and Pareto analysis to explore this relationship, especially in the context of large-scale time series datasets. Our experiments illustrate the effectiveness of this approach in managing the trade-off between model efficiency and accuracy.

Chapter 5 centers around blood transfusion decisions in ICUs, a common but complex intervention [5]. We develop a decision support tool using an off-policy batch reinforcement learning algorithm to guide transfusion decisions. By conducting experiments on two real-world critical care datasets, we demonstrate the potential of reinforcement learning in treatment recommendations, showing that it can optimize real-time treatment strategies by improving patient outcomes.

In Chapter 6, we investigate the role of large language models (LLMs) in clinical language understanding tasks [6]. We conduct a comprehensive evaluation of state-of-the-art LLMs and introduce a novel prompting strategy, self-questioning prompting (SQP), tailored to enhance their performance. This chapter emphasizes the need for cautious implementation of LLMs in healthcare settings, ensuring a collaborative approach with domain experts and continuous verification by human experts to achieve responsible and effective use.

Chapter 2

Empirical Analysis of Machine Learning Configurations for Prediction of Multiple Organ Failure in Trauma Patients

2.1 Introduction

Multiple organ failure (MOF) is a clinical syndrome with variable causes including pathogens [7], complicated pathogenesis [8], and a major cause of mortality and morbidity for trauma patients who are admitted to Intensive Care Units (ICU) [9]. Based on recent studies on ICU trauma patients, up to 47% have developed MOF, and MOF increased the overall risk of death 6 times compared to patients without MOF [10]. To prevent the development of MOF for trauma patients from progression to an irreversible stage, it is essential to diagnose MOF early and effectively. Many scoring systems have been proposed to predict MOF [11, 12, 13, 14] and researchers have attempted to predict MOF

on trauma patients using predictive models in an early phase [15, 16].

The rapid growth of data availability in clinical medicine requires doctors to handle extensive amounts of data. As medical technologies become more complicated, technological advances like machine learning (ML) are increasingly needed to improve real-time analysis and interpretation of the results [17]. In recent years, practical uses of ML in healthcare have grown tremendously, including cancer diagnosis and prediction [18, 19, 20], tumor detection [21, 22], medical image analysis [23, 24], and health monitoring [25, 26].

Compared to traditional medical care, ML-assisted clinical decision support enables a more standardized process for interpreting complex multi-modality data. In the long term, ML can provide an objective viewpoint for clinical practitioners to improve performance and efficiency [27]. ML is often referred to as a black box: explicit input data and output decisions, but opaque at intermediate learning process. Additionally, in medical domains, there is no universal rule for selecting the best configuration to achieve the optimal outcome. Moreover, medical data has its own challenges such as numerous missing values [28] and colinear variables [29]. Thus it is difficult to process the data and choose the proper model and corresponding parameters, even for a ML expert. Furthermore, detailed quantitative analysis of the potential impacts of different settings of ML systems on MOF has been missing.

In this chapter, we experiment with comprehensive ML settings for prediction of MOF, considering 6 different dimensions from data preprocessing (missing value treatment, label balancing, feature scaling), feature selection, classifier choice, to hyperparameter tuning. To predict MOF for trauma patients at an early stage, we use only initial time measurements (hour 0) as inputs. We mainly use area under the receiver operating characteristic curve (AUC) to evaluate MOF prediction outcomes. We focus on analyzing the relationships among configuration complexity, predicted performance, and

performance variation. Additionally, we quantify the relative impacts of each dimension.

The main contributions of this chapter include:

(1) To the best of our knowledge, this is the first work to conduct a thorough empirical analysis quantifying the predictive performance with exhaustive ML configurations for MOF prediction.

(2) We provide general guidance for ML practitioners in healthcare and medical fields through quantitative analysis of different dimensions commonly used in ML tasks.

(3) Experimental results indicate that classifier choice contributes most to both performance improvement and variation. Complex classifiers including ensemble methods bring higher default/optimized performance, along with a higher risk of inferior performance compared to simple ones on average.

The remainder of this chapter is organized as follows. Section 2.2 describes the dataset and features we use. All of the ML configurations are available in Section 2.3. Experimental results are discussed in Section 2.4. Finally, our conclusions are presented in Section 2.5.

2.2 Dataset

Our dataset, collected from the San Francisco General Hospital and Trauma Center, contains 2190 highest level trauma activation patients evaluated at the level I trauma center. Due to the urgency of medical treatment, there are numerous missing values for time-dependent measurements. Thus we have chosen to consider only those features with a maximum missing value percentage of 30% over all patients. To obtain a timely prediction, early lab measurements (hour 0) as well as patients' demographic and illness

information were extracted as the set of features. Detailed feature statistics are available in Table 2.1.

Table 2.1: MOF dataset statistics. Italicized features are categorical.

Feature type	# of extracted features	Features
Demographic	5	*gender*, age, weight, *race, blood type*
Illness	2	*comorbidities, drug usage*
Injury factors	4	*blunt/penetrating trauma*, # of rib fractures, *orthopedic injury*, *traumatic brain injury*
Injury scores	8	injury severity score, 6 abbreviated injury scale (head, face, chest, abdomen, extremity, skin), Glasgow coma scale score
Vital sign measurements	4	heart rate, respiratory rate, systolic blood pressure, mean arterial pressure
Blood-related measurements	13	white blood cell count, hemoglobin, hematocrit, serum CO_2, prothrombin time, international normalized ratio, partial thromboplastin time, blood urine nitrogen, creatinine, blood pH, platelets, base deficit, *factor VII*

Our target variable consists of binary class labels (0 for no MOF and 1 for MOF). Then, the data with feature and target variables is randomly split into training and testing sets at the ratio of $7:3$.

2.3 Methods

Based on ML pipelines and special characteristics of our data such as large number of missing values and varying scales in feature values, we consider comprehensive ML configurations from the following 6 dimensions: data preprocessing (missing value treatment (MV), label balancing (LB), feature scaling (SCALE)), feature selection (FS), classifier choice (CC), and hyperparameter tuning (HT). In the remainder of the chapter, we will interchangeably use the full name and corresponding abbreviations shown in parentheses. Further details on each dimension are described below.

2.3.1 Data Preprocessing

Methods to handle the dataset with missing values, imbalanced labels, and unscaled variables are essential for the data preprocessing process. We use several different methods to deal with each of these problems.

Missing Value Treatment

In our dataset, numerous time-dependent features cannot be recorded on a timely basis, and missing data is a serious issue. We consider three different ways to deal with missing values, where the first method serves as the baseline setting for MV, and the latter two methods are common techniques of missing value imputation in ML.

1. Remove all patients with any missing values for the features listed in Section 2.2.

2. Replace missing values with mean for numerical features and mode for categorical features over all patients.

3. Impute missing values by finding the k-nearest neighbors with the Euclidean distance metric for each feature respectively.

Label Balancing

Our dataset is imbalanced as the sample class ratio between class 0 and class 1 is 11 : 1. Keeping imbalanced class labels serves as the baseline setting for LB. Three different ways are considered to resample the training set.

1. Oversampling the minority class (label 1)
    1.1 Method: SMOTE (synthetic minority over-sampling technique) [30].
    1.2 Explanation: choose k-nearest neighbors for every minority sample and then create new samples halfway between the original sample and its neighbors.
2. Undersampling the majority class (label 0)
    2.1 Method: NearMiss [31].
    2.2 Explanation: when samples of both classes are close to each other, remove the samples of the majority class to provide more space for both classes.
3. Combination of oversampling and undersampling
    3.1 Method: SMOTE & Tomek link [32].
    3.2 Tomek link: two samples are k-nearest neighbors to each other but come from different classes.
    3.3 Explanation: first create new samples for the minority class and then remove the majority class sample in any Tomek link.

Feature Scaling

Since the range of feature values in our dataset varies widely, we perform feature scaling. No scaling on any feature serves as the baseline setting for SCALE. Two common scaling techniques are used for numerical features.

1. Normalization: rescale values to range between 0 and 1.

2. Standardization: rescale values with mean 0 and standard deviation 1.

2.3.2 Feature Selection

In medical datasets, there usually exist many highly correlated features, and some features that are weakly correlated to the target [29, 33]. Thus it is essential to identify the most relevant features that may help to improve the outcome of the analysis. Using all of the features described in Section 2 serves as the baseline setting for FS. We consider two main feature selection techniques: filter and wrapper methods.

1. Filter-based methods (independent of classifiers):

    1.1 Use correlation between features and the target to select features which are highly dependent on the target.

    1.2 Filter out numerical features using ANOVA F-test and categorical features using χ^2 test.

2. Wrapper-based methods (dependent on classifiers):

    2.1 Method: RFE (recursive feature elimination) in random forest.

    2.2 Explanation: perform RFE repeatedly such that features are ranked by importance, and the least important features are disregarded until a specific number of features remains.

2.3.3 Classifier Choice

We experimented with 15 classifiers on the dataset. In general, these classifiers can be divided into two main categories: single and ensemble. Lists of all classifiers are

available in Table 2.2. For ensemble classifiers (combination of individual classifiers), we tried bagging (BAG, RF, ET), boosting (GB, ABC, XGB, LGBM), voting (VOTE) and stacking (STACK). In bagging, DT is a homogeneous weak learner. Multiple DTs learn the dataset independently from each other in parallel and the final outcome is obtained by averaging the results of each DT. In boosting, DT also serves as a homogeneous weak learner. However, DTs learn the dataset sequentially in an adaptive way (new learner depends on previous learners' success), and the final outcome is determined by weighted sum of previous learners. In voting, heterogeneous base estimators (LR, RF, SVM, MLP, ET) are considered, where each estimator learns the original dataset and the final prediction is determined by majority voting. In stacking, several heterogeneous base learners (RF, KNN, SVM) learn the dataset in parallel, and there exists a meta learner (LR) that combines the predictions of the weak learners. Abbreviations of classifiers shown in parentheses for voting and stacking are the ones we use.

Table 2.2: List of 6 single classifiers and 9 ensemble classifiers. Corresponding abbreviations of each classifier are shown in parentheses.

Single classifiers	Ensemble classifiers
Logistic Regression (LR) Support Vector Machine (SVM) Naive Bayes (NB) K-nearest Neighbors (KNN) Decision Tree (DT) Multi-layer Perceptron (MLP)	Bagged Trees (BAG) Random Forest (RF) Extra Trees (ET) Gradient Boosting (GB) Adaptive Boosting (ABC) Extreme Gradient Boosting (XGB) Light Gradient Boosting Machine (LGBM) Voting (VOTE) Stacking (STACK)

2.3.4 Hyperparameter Tuning

Hyperparameters are crucial for controlling the overall behavior of classifiers. Default hyperparameters of classifiers serve as the baseline setting for HT. We apply grid search

to perform hyperparameter tuning for all classifiers. Detailed information about tuned hyperparameters is available in Table 2.3.

Table 2.3: Detailed configurations of tuned hyperparameters for all classifiers. All of the hyperparameter names come from *scikit-learn* [34].

Classifiers	♯ of tuned hyperparameters	Hyperparameter lists
LR	3	C, class_weight, penalty
SVM	4	C, gamma, kernel, class_weight
KNN	3	n_neighbors, weights, algorithm
NB	1	var_smoothing
DT	5	min_samples_split, max_depth, min_samples, leaf_max_features, class_weight
MLP	3	activation, solver, alpha
BAG	2	base_estimator, n_estimators
RF	2	n_estimators, max_features
ET	2	n_estimators, max_features
GB	2	n_estimators, max_depth
ABC	3	base_estimator, n_estimators, learning_rate
XGB	2	min_child_weight, max_depth
LGBM	4	num_leaves, colsample_bytree, subsample, max_depth
VOTE	2	C (SVM), n_estimators (ET)
STACK	2	C (SVM), n_neighbors (KNN)

2.4 Experiments and Results

We formulated MOF prediction as a binary classification task. All of the experiments in this chapter were implemented using *scikit-learn* [34]. As mentioned in Section 2.2, our training and testing dataset is randomly split with a ratio of 7 : 3. One-hot encoding is applied to all categorical features. For each classifier, we use the same training and testing dataset. We use AUC as our main performance metric, as it is commonly used for MOF prediction in the literature [12, 35, 36]. It provides a "summary" of classifier

performance compared to single metrics such as precision and recall. AUC represents the probability that a classifier ranks a randomly chosen positive sample (class 1) higher than a randomly chosen negative sample (class 0), and thus useful for imbalanced datasets.

In this section, we quantify the impacts (improvement and variation) of each dimension on the predicted performance over our testing dataset.

2.4.1 Influence of Individual Dimensions

First, we evaluate how much each dimension contributes to the AUC score improvement and variation respectively, and find the correlation between performance improvement and variation over all dimensions.

Performance Improvement across Dimensions

For HT, MV, LB, SCALE, and FS, we define the ***baseline*** as default hyperparameter choices, using no missing value imputation, no label balancing, no feature scaling, and no feature selection, respectively. For CC, we choose SVM, which achieves the median score among all classifiers, as the ***baseline***. Then we quantify the performance improvement of each dimension. Fig. 2.1 shows the percentage that each dimension contributes to the improvement in the AUC score over baseline by tuning only one dimension at a time while leaving others at baseline settings. We observe that CC contributes most to the performance improvement (15.00%) for MOF prediction. After CC, LB (10.81%), FS (10.09%), MV (7.90%), HT (6.94%), and FS (2.45%) bring decreasing degrees of performance improvement in the AUC score.

Table 2.4 shows the improvement of every single dimension on each classifier over the baseline. In general, MV and LB tend to provide the greatest performance improvement for most classifiers. For RF, ET, and LGBM, FS contributes the most to improvement in

performance since these classifiers require feature importance ranking intrinsically, and external FS improves their prediction outcomes to a large extent. Note that the classifier for which SCALE has the largest impact is KNN, as it is a distance-based classifier which is sensitive to the range of feature values. Also, due to instability and tendency to overfit, HT is the most critical for DT improvement.

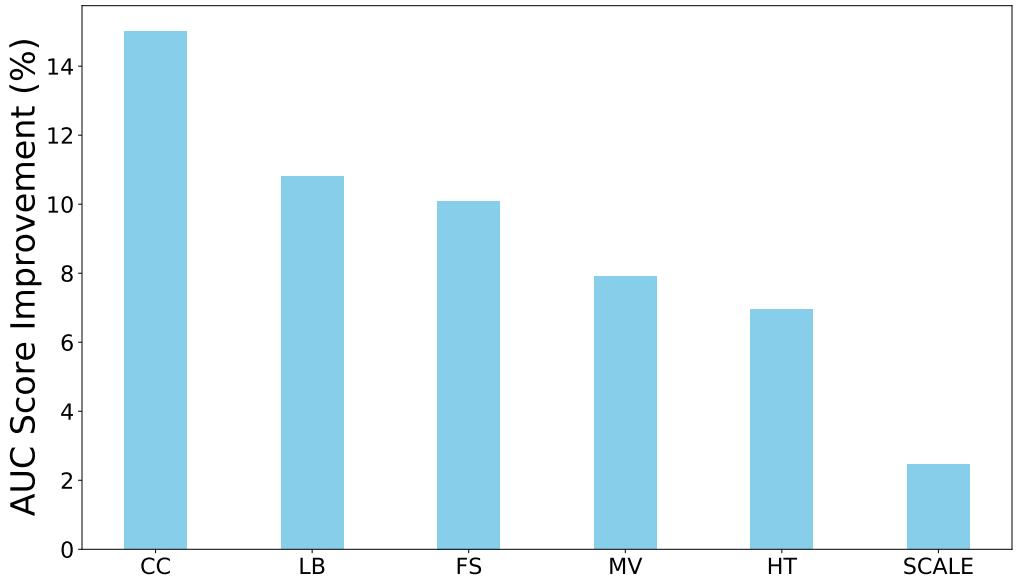


Figure 2.1: Performance improvement in the AUC score of each dimension over the baseline when tuning only one dimension at a time while leaving others at baseline settings. CC brings the greatest performance improvement, followed by LB, FS, MV, HT, and SCALE in decreasing order of improvement.

In addition to AUC, 6 other performance metrics are used to measure the performance improvement degree of each dimension. The results in Table 2.5 reveal that CC brings the greatest improvement regardless of the metrics we use. Contributions from HT and SCALE are relatively small compared to other dimensions.

Table 2.4: Column 1 shows a total of 15 classifiers. Columns 2 to 6 represent the percentage (two decimal places accuracy) of AUC score improvement when tuning each individual dimension while leaving other dimensions at baseline settings for each classifier. Bold entries represent the dimension that contributes to the largest improvement for the specific classifier. MV and LB tend to dominate in performance improvement for most classifiers.

Classifier	MV (%)	LB (%)	SCALE (%)	FS (%)	HT (%)
LR	2.78	**11.48**	0.30	5.50	3.03
SVM	3.37	**26.83**	2.38	20.95	3.21
KNN	13.60	11.85	**17.68**	13.12	15.60
NB	0.60	**38.90**	0.17	4.12	2.84
DT	12.87	16.22	0.42	15.34	**38.85**
BAG	2.94	**8.91**	0.28	7.05	5.43
RF	4.13	5.34	0.28	**5.85**	1.04
ET	3.82	7.87	0.00	**18.96**	1.33
ABC	**19.33**	7.02	0.00	16.99	12.99
GB	**12.44**	3.81	0.02	6.63	4.08
LGBM	7.03	1.85	2.75	**10.39**	3.13
XGB	**11.46**	3.97	0.02	7.47	4.27
MLP	**10.78**	5.08	6.05	7.53	5.69
STACK	6.94	**8.94**	4.32	5.48	1.82
VOTE	**6.38**	4.00	2.11	6.04	0.85

Table 2.5: Performance improvement in different metrics of each dimension. The performance improvement of each dimension on other metrics displays an order consistent with that of the AUC score.

	AUC	F-score	G-mean	Precision	Sensitivity/ Recall	Specificity	Accuracy
CC (%)	15.00	15.58	10.50	16.41	10.50	11.86	10.50
LB (%)	10.81	11.34	9.33	13.27	9.33	10.72	9.34
FS (%)	10.09	7.33	6.30	10.61	6.30	6.94	6.30
MV (%)	7.90	5.30	4.60	5.83	4.59	4.95	4.59
HT (%)	7.46	2.11	3.21	3.41	3.20	4.64	3.20
SCALE (%)	2.45	1.04	0.65	3.03	0.65	0.48	0.65

Performance Variation across Dimensions

For all of the ML configurations, we further investigate how much each dimension contributes to the performance variation in the AUC score. By tuning only one dimension at a time while leaving other dimensions at baseline settings, we obtain a range of AUC scores. Performance variation is the difference between the maximum and the minimum score of each dimension. Fig. 2.2 shows the proportion of each dimension that brings the performance variation in the AUC score. Based on Fig. 2.2, we notice that CC, which brings the largest performance improvement, also brings the largest performance variation (10.98 %). After CC, LB (7.00 %), FS (6.93 %), MV (5.64 %), HT (4.97 %), and SCALE (1.66 %) bring decreasing degrees of performance variation in the AUC score.

Table 2.6 shows the variation of every single dimension on each classifier over the baseline. We observe that for each classifier, if one dimension brings a larger performance improvement, it also results in a larger performance variation. For our assessment of performance variation, the same metrics as above are used for evaluation on each dimension. Using the same metrics as above, Table 2.7 shows that the proportion of performance variation in different metrics from each dimension follows an order that is consistent with the performance improvement in Table 2.5. Thus, for different metrics, greater improvement brings greater variation of each dimension. For every step that researchers take when predicting MOF using ML, they should always be aware of the trade-off between benefits (improvement in performance) and risks (variation in performance) when adjusting each dimension.

2.4.2 Performance Comparison across Classifiers

We have shown that classifier choice is the largest contributor to both performance improvement and variation in the AUC score. Hence, we further investigate the perfor-

Table 2.6: Columns 2 to 6 represent the proportion (two decimal places accuracy) of each dimension that contributes to the performance variation in the AUC score. Bold entries represent the dimension that contributes to the largest variation for the specific classifier. MV and LB tend to result in larger performance variation for most classifiers.

Classifier	MV (%)	LB (%)	SCALE (%)	FS (%)	HT (%)
LR	2.28	**8.44**	0.25	4.27	2.49
SVM	2.45	**16.87**	1.79	13.04	2.42
KNN	7.97	6.95	**10.36**	7.69	9.14
NB	0.48	**22.22**	0.13	3.25	2.25
DT	7.13	8.99	0.23	8.50	**21.53**
BAG	2.22	**6.17**	0.21	5.26	4.10
RF	3.35	4.14	0.23	**4.49**	0.84
ET	3.22	6.14	0.00	**13.41**	1.12
ABC	**13.09**	4.61	0.00	11.51	9.40
GB	**9.59**	2.83	0.02	5.11	3.14
LGBM	5.54	1.43	2.16	**7.76**	2.47
XGB	**8.70**	3.01	0.02	5.59	3.24
MLP	**7.89**	3.55	4.43	5.30	4.16
STACK	5.47	**6.47**	3.41	4.20	1.43
VOTE	**5.19**	3.13	1.71	4.63	0.69

Table 2.7: Performance variations in different metrics of each dimension. The performance variation of each dimension on other metrics displays an order that is consistent with that of the AUC score.

	AUC	F-score	G-mean	Precision	Sensitivity/ Recall	Specificity	Accuracy
CC (%)	10.98	11.87	8.57	12.86	8.57	10.60	8.57
LB (%)	7.00	7.29	6.83	9.02	6.83	10.14	6.82
FS (%)	6.93	5.27	4.38	7.62	4.37	4.70	4.38
MV (%)	5.64	4.36	3.69	4.77	3.69	2.98	3.68
HT (%)	4.87	2.55	3.52	1.54	3.52	2.72	3.53
SCALE (%)	1.66	0.88	0.57	1.47	0.57	0.46	0.57

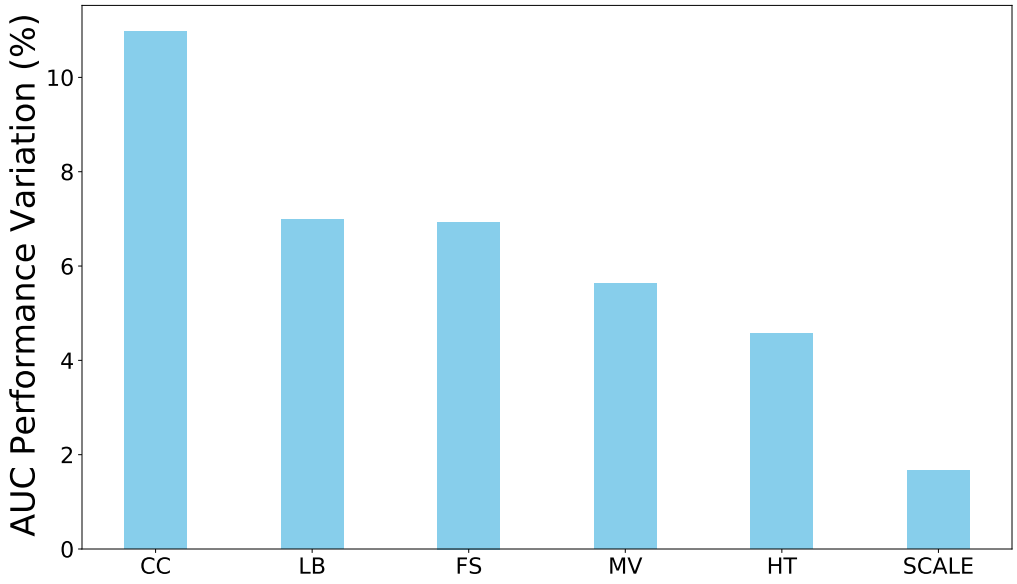


Figure 2.2: Performance variation in the AUC score when tuning only one dimension at a time while leaving others at baseline settings. CC brings the greatest performance variation, followed by LB, FS, MV, HT, and SCALE in decreasing order of variation. Larger improvement also brings the risk of larger variation for each dimension.

mance differences among classifiers. Specifically, we investigate the relationships among classifier complexity, performance, and performance variation.

Default versus Optimized Performance

Default classifiers are defined as classifiers with default parameters, while ***optimized*** classifiers are those for which hyperparameter tuning with 10-fold cross validation is applied using grid search. We compare the performance of default and optimized classifiers in consideration of all other dimensions, i.e., MV, LB, SCALE, and FS. The average AUC scores of all classifiers with default and optimized settings are shown in Fig. 2.3. In general, ensemble classifiers perform better than single classifiers regardless of default

or optimized performance.

In addition to AUC, 6 other performance metrics are used to evaluate the performance of all classifiers. We use the median score to rank classifiers with both default and optimized settings. Then, NDCG (normalized discounted cumulative gain), one of the most prevalent measures of ranking quality [37], is used to compare classifier rankings between each of these metrics and the AUC score. Detailed relevance scores are shown in Table 2.8. The result indicates that the median performance of each classifier is similar no matter which metric is used. This also suggests that the AUC score can represent classifiers' overall performance well.

Based on the above experiments, ensemble classifiers should be prioritized in MOF prediction since they usually bring better predictive performance than single classifiers.

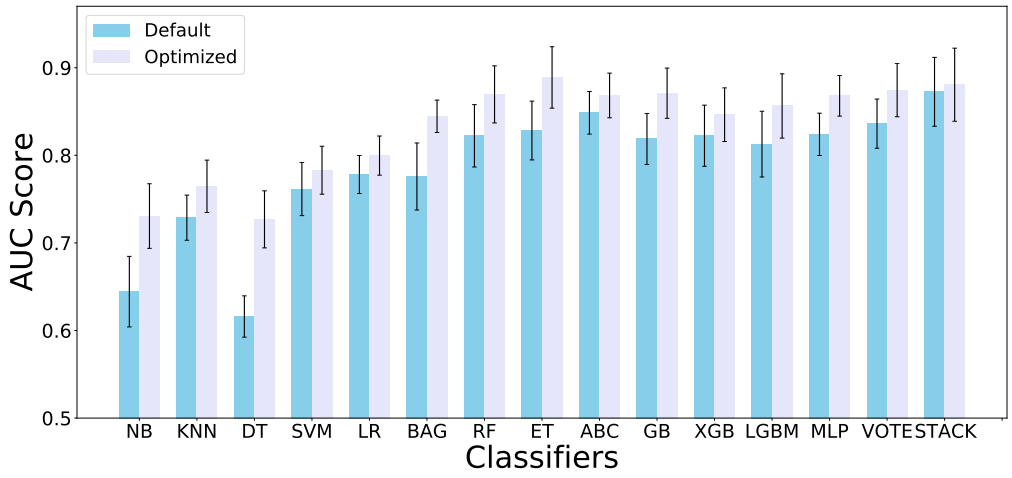


Figure 2.3: Comparison of default and optimized performance over all classifiers. Classifiers listed on the left-hand side of BAG are single while the ones on the right-hand side are ensemble and MLP. Overall, ensemble methods have better default and optimized performance compared with single classifiers.

Table 2.8: Column 1 represents 6 other performance metrics. Columns 2 and 3 show the NDCG score between each of these metrics and the AUC score when ranking 15 classifiers by their median performance in default and optimized settings, respectively. Median performance of classifiers is similar regardless of which metric to use.

	Default (%)	Optimized (%)
F-score	96.92	97.92
G-mean	96.46	97.63
Precision	95.49	90.01
Sensitivity/Recall	98.42	97.59
Specificity	95.35	97.46
Accuracy	96.46	97.59

Performance Variation across Classifiers

We measure the performance variation for each classifier in consideration of all other dimensions, i.e., MV, LB, SCALE, FS, and HT. For each classifier, we get a range of AUC scores. The size of the range determines the extent of performance variation. Fig. 2.4 shows the performance variation in the AUC score of all classifiers. The order of listed classifiers on the x-axis is based on increasing model complexity, which is measured by classifier training time with default settings. The complexity of classifiers and performance variation demonstrates an evident 'U-shaped' relationship. When the classifier is 'too simple', its performance variation is relatively large. When the complexity of the classifier is 'appropriate', the performance variation is relatively small. If the classifier becomes 'too complex', it is also at the risk of larger performance variation. Therefore, classifiers with 'appropriate' complexity are more stable, with smaller changes in performance, while 'too simple' or 'too complex' classifiers are relatively unstable with larger changes in performance in general.

In addition to AUC, the same metrics as above were used to validate the performance variation of all of the classifiers. We use the range (difference between maximum and minimum scores) to rank classifiers in consideration of MV, LB, SCALE, FS, and HT.

Then, NDCG is used to compare classifier rankings between each of these metrics and the AUC score. Table 2.9 displays detailed relevance scores. The result suggests that other metrics show a similar 'U-shaped' relationship between classifier complexity and performance variation as the AUC score. When predicting MOF, it is inappropriate for clinical practitioners to choose 'too simple' and 'too complex' classifiers since they may run the risk of underfitting and overfitting, respectively.

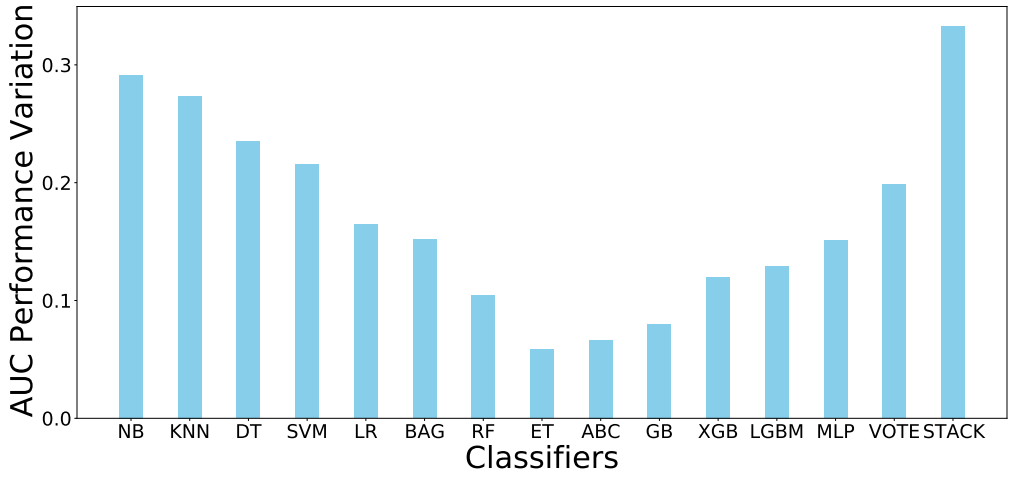


Figure 2.4: Performance variation comparison over all classifiers. The order of classifiers listed on the x-axis is based on increasing model complexity. 'Too simple' and 'too complex' classifiers result in larger performance variation. The performance variation of classifiers with 'appropriate' complexity is relatively small.

Table 2.9: NDCG score between each of 6 other performance metrics and the AUC score in terms of classifier complexity and performance variation. Different metrics show a similar 'U-shaped' relationship.

	F-score	G-mean	Precision	Sensitivity/ Recall	Specificity	Accuracy
Relevance (%)	93.15	94.98	94.37	93.24	93.13	93.77

2.5 Discussion

We have provided a timely MOF prediction using early lab measurements (hour 0), patients' demographic and illness information. Our study quantitatively analyzes the performance via the AUC score in consideration of a wide range of ML configurations for MOF prediction, with a focus on the correlations among configuration complexity, predicted performance, and performance variation. Our results indicate that choosing the correct classifier is the most crucial step that has the largest impact (performance and variation) on the outcome. More complex classifiers including ensemble methods can provide better default/optimized performance, but may also lead to larger performance degradation, without careful selection. Clearly, more MOF data is needed to provide a more general conclusion. Our work can potentially serve as a practical guide for ML practitioners whenever they conduct data analysis in healthcare and medical fields.

Chapter 3

Integrating Physiological Time Series and Clinical Notes with Transformer for Early Prediction of Sepsis

3.1 Introduction

Sepsis is a life-threatening organ dysfunction caused by a dysregulated host response to infection [38], contributing to $30\% - 50\%$ of inpatient mortality in the U.S [39]. The capability of early detection of sepsis allows for earlier interventions and treatment, thus improving patient outcomes. Following the widespread adoption of electronic health record (EHR) systems, researchers are particularly interested in using the EHR data to predict sepsis [40, 41, 42].

One challenge of using EHR is that it stores both structured data (e.g., vital signs and laboratory measurements) and unstructured data (e.g., physician and nursing notes).

Nevertheless, the heterogeneities across modalities increase the difficulty of performing sepsis prediction tasks. Thus, previous research works have focused on analyzing single data modality in isolation [41, 43, 44]. Structured physiological data can represent patients' true physiological signals. However, in the case of sepsis, this data is incomplete and irregular due to urgency in the Intensive Care Units. Although unstructured medical notes can help understand patients' conditions more directly by capturing information regarding patients' symptom changes, it is insufficient to use notes alone to determine patients' status without support from physiological data.

To address the issues above, we propose a multimodal Transformer model that incorporates information from both physiological time series data and clinical notes for early prediction of sepsis. We use two large critical care datasets: the Medical Information Mart for Intensive Care III (MIMIC-III) [45] and the eICU Collaborative Research Database (eICU-CRD) [46]. Comprehensive experiments are conducted on the above two datasets to validate our approach, including performance comparison with baselines, ablation analysis, and case studies. Experimental results suggest that our proposed method outperforms six baselines on all metrics on both datasets. In addition, empirical ablation analysis and case studies indicate that each single modality contains unique information that is unavailable to the other modality. Hence, our model improves predictive performance by utilizing both physiological time series data and clinical notes.

The main contributions of this chapter are highlighted as follows:

(1) To the best of our knowledge, this is the first Transformer-based model that incorporates multivariate physiological time series data and clinical notes for early sepsis prediction.

(2) Our experimental results indicate that both modalities complement each other. Thus, our method with both physiological data and clinical notes results in the

best overall performance compared with unimodal methods. When using both modalities, our method outperforms competitive baselines on all metrics.

(3) We perform attention mechanism visualization on clinical notes to improve the interpretability regarding the patients' status, which is not available in physiological data. In addition, distinctive distributions of physiological features between sepsis and non-sepsis patients demonstrate the unique information contained in physiological data but not in clinical notes.

The remainder of this chapter is organized as follows. Section 3.2 describes related work. The formal problem description is in Section 3.3. The proposed model is outlined in Section 3.4. Section 3.5 describes the datasets we use for evaluation. Experiments and results are discussed in Section 3.6. Finally, our conclusions are presented in Section 3.7.

3.2 Related Work

In this section we review related work on clinical notes and physiological time series modeling, as well as multimodal methods in the clinical domain.

3.2.1 Clinical Notes Modeling

With the increasing availability of clinical notes over the past several years, there has been notable progress in understanding and using clinical text data to improve clinical prediction outcomes. Natural language processing (NLP) and information retrieval techniques have been widely applied on different types of clinical tasks, such as clinical relation extraction [47], de-identification of clinical notes [48], and clinical question answering [49]. One common method for text representation is word embedding. In recent years, the appearance of the Transformer-based BERT [50] has offered an advantage over

previous word embedding methods such as Word2Vec [51] and GloVe [52] since it produces word representations that are dynamically informed by the words around them, which can effectively capture information from both the left and right contexts. In the clinical domain, BioBERT was pre-trained on PubMed abstracts and articles and was able to better identify biomedical entities and boundaries than base BERT [53]. Base BERT and BioBERT have been further fine-tuned on the MIMIC-III dataset [45] and released as ClinicalBERT and Clinical BioBERT, respectively [54].

3.2.2 Physiological Time Series Modeling

Previous studies applied classical models such as Gaussian process (GP) and linear dynamical systems (LDS) to clinical time series modeling [55, 56]. Given the growing availability of clinical data, recent studies demonstrate that RNN-based deep learning (DL) methods have become sought-after alternatives in clinical sequence modeling [41, 57]. More recently, an attention-based DL model has been proposed for clinical time series modeling [58].

3.2.3 Multimodal Methods in the Clinical Domain

Multimodal representation learning is a fundamentally complex problem due to multiple sources of information [59]. Undeniably, multiple sources of data can provide complementary information, enabling more robust predictions [60]. In the clinical domain, predictive models have been developed by integrating continuous monitoring data and discrete clinical event sequences [61]. Combinations of multiple modalities such as clinical texts, procedures, medications, and laboratory measurements have shown improved performance on inpatient mortality, length of stay, and 30-day readmission prediction tasks [62]. Unstructured clinical notes combined with structured measurements have

been used for survival analysis of ICU trauma patients [63].

3.3 Problem Definition

For a patient cohort consisting of P patients, the multivariate physiological time series (MPTS) data associated with each patient can be expressed as $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(M)}\} \in \mathbb{R}^{L \times M}$ with $\boldsymbol{x}^{(j)^T} = \{x_1^{(j)}, x_2^{(j)}, \cdots, x_L^{(j)}\}$. M and L represent the number of clinical features and the number of hours after admission, respectively. In addition, for each patient, sequences of clinical notes are used. The true label of each patient's clinical outcome is $y \in \{0, 1\}$ (1 indicates sepsis and 0 indicates non-sepsis). Altogether, each of our datasets can be represented as $\{(\mathbf{X}_i, \mathbf{C}_i, \mathrm{Y}_i) | i = 1, 2, \cdots, P\}$ where $\mathbf{X}_i, \mathbf{C}_i, \mathrm{Y}_i$ are the respective MPTS sequence, available clinical notes within L hours, and the class label for patient i. We formulate sepsis prediction as a binary classification task, for which the goal is to learn a mapping:

$$(\mathbf{X}_i, \mathbf{C}_i) \rightarrow \mathrm{Prob}(\mathrm{Y}_i = 1 | (\mathbf{X}_i, \mathbf{C}_i)),$$

where $i = 1, 2, \cdots, P$. In other words, MPTS data and clinical notes are used simultaneously to predict whether ICU patients admitted through the Emergency Department will develop sepsis.

3.4 Methods

In this section we propose the multimodal Transformer modeling framework. The model structure is illustrated in Figure 3.1.

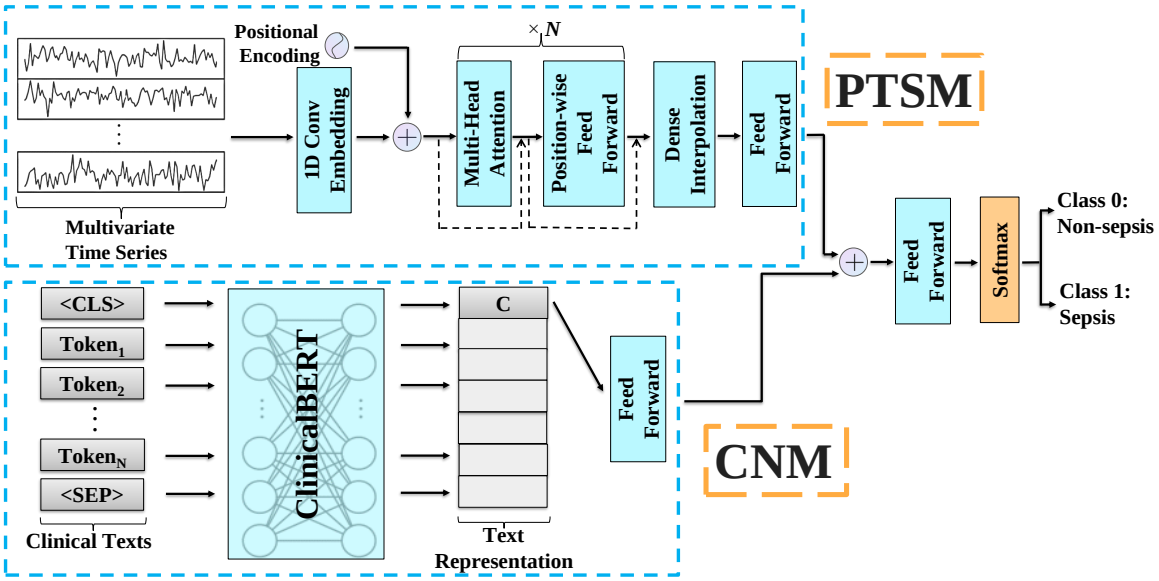


Figure 3.1: An overview of the multimodal Transformer framework. The physiological time series model (PTSM) consists primarily of sequence embeddings, a stack of N Transformer encoder layers (multi-head self-attention sublayer and position-wise FNN sublayer) with residual connection around each sublayer, dense interpolation, and FNN. The clinical notes model (CNM) consists of text representations using ClinicalBERT, and the output $\langle$ CLS $\rangle$ representation is then used to feed into FNN. The output representations from PTSM and CNM are concatenated and fed into FNN, and the final Softmax layer is used for the binary classification task.

3.4.1 Clinical Notes Model (CNM)

The CNM is composed of clinical text representations using ClinicalBERT [54] and a feedforward neural network (FNN). The output $\langle$ CLS $\rangle$ representation following ClinicalBERT is fed into FNN.

Transformer

We begin by introducing Transformer's architecture [64], the foundation of Bidirectional Encoder Representations from Transformers (BERT), for which we use for clinical text representations. In Transformer [64], the self-attention mechanism enables the model to capture both short- and long-term dependencies, and different attention heads

can learn different aspects of attention patterns. In the self-attention layer, an attention function maps a query $\mathbf{Q}$ and a set of key-value pairs $\{\mathbf{K},\mathbf{V}\}$ to an output $\mathbf{O}$. Specifically, a multi-head self-attention sublayer simultaneously transforms the queries, keys and values into H distinct and learnable linear projections, namely

$$\mathbf{Q}^h = \mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}^h = \mathbf{K}\mathbf{W}_h^K, \mathbf{V}^h = \mathbf{V}\mathbf{W}_h^V,$$

where $\mathbf{Q}^h, \mathbf{K}^h, \mathbf{V}^h$ are the respective query matrices, key matrices, and value matrices of the h-th attention head, with $h = \{1, 2, \cdots, H\}$. Here, $\mathbf{W}_h^Q, \mathbf{W}_h^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_h^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ denote learnable parameter matrices and d_{model} is the text embedding dimension. Next, H attention functions are performed in parallel to produce a sequence of vector outputs:

$$\begin{aligned}\mathbf{O}^h &= \text{Attention}(\mathbf{Q}^h, \mathbf{K}^h, \mathbf{V}^h) \\ &= \text{Softmax}\left(\frac{\mathbf{Q}^h {\mathbf{K}^h}^T}{\sqrt{d_k}}\right)\mathbf{V}^h.\end{aligned}$$

Finally, the outputs $\mathbf{O}^1, \mathbf{O}^2, \cdots, \mathbf{O}^H$ are concatenated and linearly projected again to produce the final representation.

Text Representation with ClinicalBERT

BERT is a pre-trained language representation based on the Transformer encoder architecture [50, 64]. BERT and its variants have exhibited outstanding performance in various NLP tasks. In medical contexts, ClinicalBERT develops clinically oriented word representations for clinical NLP tasks. Within ClinicalBERT, each token in clinical notes can be expressed as a sum of corresponding token embeddings, segment embeddings, and position embeddings. When feeding multiple sentences into ClinicalBERT, segment

embeddings identify the sequence that a token is associated with and position embeddings of each token are a learned set of parameters corresponding to the token's position in the input sequence [65]. We use pre-trained ClinicalBERT for contextual representations of clinical notes.

3.4.2 Physiological Time Series Model (PTSM)

Inspired by the model architecture of Transformer [58, 64], the PTSM is composed of sequence embeddings, positional encoding, a stack of N identical Transformer encoder layers, and dense interpolation to incorporate temporal order.

Input Embeddings

In most NLP models, input embeddings are commonly used to map relatively low-dimensional vectors to high-dimensional vectors, which facilitate sequence modeling [66]. For the same reason, a time sequence embedding is required to capture the dependencies among different features without considering the temporal information [58]. A 1D convolutional layer is employed to obtain the K-dimensional embeddings ($K > M$) at each time step.

Positional Encoding

In order to include the MPTS order information, we apply the same sinusoidal functions for the positional encoding layer as [64] to encode the sequential information and add it to the input embeddings of the sequence.

Transformer Encoder

We take advantage of the multi-head self-attention mechanism to capture dependencies of sequences. Similar to [64], we employ 8 parallel attention heads. Following the attention output, a position-wise FNN is applied with two 1D convolutional layers with kernel size 1, and a ReLU activation function in between. A residual connection is employed around each of the two sublayers.

Dense Interpolation

A concise representation of the output sequence from the Transformer encoder layer is needed since we do not make predictions at each time step [67]. A dense interpolation algorithm is applied on learned temporal representations for partial temporal order encoding. Given MPTS data, the pseudocode to perform dense interpolation is shown in Algorithm 1. Let $\mathbf{e}_l \in \mathbb{R}^{d_k}$ represent the intermediate representation following the Trans-

Algorithm 1: Dense Interpolation

Input: time step l, time sequence length L, input embeddings $\mathbf{e}_l$, interpolation coefficient I.
Output: Dense representation $\mathbf{z}$.
for $l = 1$ ***to*** L **do**
  e $= I \times l \ / L$;
  for $i = 1$ ***to*** I **do**
    $r = pow(1 - abs(e - i)/I, 2)$;
    $\mathbf{z}_i = \mathbf{z}_i + r \times \mathbf{e}_l$;
  end
end

former encoder layers. The size of the interpolated embedding vector is $d_k \times I$, where I is the interpolation coefficient. Algorithm 1 mainly focuses on finding the contribution of $\mathbf{e}_l$ to the position i of the final representation $\mathbf{z}$, denoted by r. At each time step l , we obtain e, the relative position in the final vector representation $\mathbf{z}$, and r is computed

as $r = (1 - \frac{|e-i|}{I})^2$. Finally, $\mathbf{z}$ is obtained by matrix multiplication of r and $\mathbf{e}_l$ when we iterate through the time steps of a sequence.

3.4.3 Incorporating PTSM and CNM

The output representations from PTSM and CNM are concatenated, and the combined latent representation is fed into FNN. We use a Softmax layer as the final layer for the binary classification problem and the loss function is given by

$$-(y \cdot \log(\hat{y})) + (1 - y) \cdot \log(1 - \hat{y}),$$

where y and $\hat{y}$ are the true and predicted labels, respectively.

3.5 Datasets

We use the MIMIC-III [45] and eICU-CRD [46] datasets to evaluate our method. MIMIC-III, a publicly available single-center clinical dataset, records $61,532$ ICU stays among $58,976$ hospital admissions, including information on $46,520$ patients from Beth Israel Deaconess Medical Center between 2001 and 2012. The eICU-CRD, a multi-center dataset, consists of health data associated with over $200,000$ admissions to ICUs throughout continental United States between 2014 and 2015. Both datasets contain de-identified data, including patient demographics, vital signs, laboratory measurements, severity of illness, diagnosis, and clinical notes.

3.5.1 Data Preprocessing Pipelines

This section is divided into structured MPTS data preprocessing and unstructured clinical notes preprocessing, respectively.

MPTS Data Preprocessing

For both datasets, patient demographics, vital signs and laboratory measurements are extracted for ICU patients admitted through the Emergency Department. A list of clinically reasonable measurement ranges provided by [68] is used to remove outlier values. In total, we extracted 40 and 38 features from the MIMIC-III and eICU-CRD datasets, respectively. Since data was irregularly sampled, we resample the observation time into hourly bins for each feature. We use the mean value to determine feature values for which there are multiple records within an hour. Missing values are imputed by a combination of forward filling (i.e. using the value of the closest past bin with regard to the missing bin) and then backward filling (i.e. using the value of closest future bin with regard to the missing bin). In addition, we remove patients with hospital admission records of less than 12 hours. We use only the first T hours of MPTS data following patient admission for early sepsis prediction, where $T = 12, 18, 24, 30, 36$. For any patient whose measurement recording hours are less than T, his/her existing last-hour measurements are replicated to T. Otherwise, we truncate his/her measurement hours to T such that the MPTS sequence length for all patients are guaranteed to be the same.

Clinical Notes Preprocessing

We use all the available clinical notes between hour 1 and hour 36 after ICU admission. If we use the first T hours of MPTS data, then all the available notes up to T hours are extracted for each patient. Over each interval of T hours, for each patient we concatenate sequences of notes. Next, common text cleaning techniques are applied such as case normalization, stop words removal, and special characters removal are applied to clean the clinical notes. To avoid potential label leakage, we remove sentences containing "sepsis" or "septic". Finally, the processed notes are fed into ClinicalBERT for text

representations.

3.5.2 Sepsis Labeling

We use the Angus criteria [69], which is an International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM) coding system, to identify sepsis for our datasets. Unlike other sepsis identification methods, it uses the final ICD diagnoses of organ failure and infection rather than feature values from the original datasets, to prevent data leakage issues [41].

3.5.3 Data Statistics

After data preprocessing, we obtain a population of $18,625$ and $60,593$ ICU admissions for the MIMIC-III and eICU-CRD datasets, respectively. The sizes of positive and negative samples identified by the Angus criteria for each dataset are illustrated in Table 1. Based on the ratio of negative and positive samples, our sepsis prediction task can be considered to be an imbalanced classification problem.

Table 3.1: Sample sizes of two datasets.

Datasets	**MIMIC-III**	**eICU-CRD**
Total	18,625	60,593
Negative	11,655	55,926
Positive	6,970	4,667

3.6 Experiments and Results

Our experiments explore: (1) the predictive performance of the multimodal Transformer model on the MIMIC-III and eICU-CRD datasets, (2) the relative importance of

individual components of the model through ablation analysis, and (3) case studies on both clinical notes and MPTS data.

3.6.1 Settings

Both datasets are randomly split, with the training set and testing set of sizes 80% and 20%, respectively. We set aside 20% of the training set for the validation set. Experiments are conducted using the first 12, 18, 24, 30 and 36 hours of patient demographics, vital signs, laboratory measurements, and clinical notes for the Emergency Department patients on both datasets. All experiments were implemented in Pytorch [70] on one NVIDIA Tesla P100 GPU. We minimize the cross entropy loss with the Adam [71] optimizer for training. The hyperparameter search space for each dataset is listed in Table 3.2, where the hyperparameter values in bold indicate the optimal values found for our model using both modalities. Note that the batch size and the sequence length choice is limited by the available GPU memory. We perform grid search for hyperparameter optimization.

Table 3.2: Hyperparameter search space of our model on both datasets. Bold values are the optimal values found using both modalities.

Hyperparameters	**MIMIC-III**	**eICU-CRD**
learning rate	[1e-4,**2e-5**,3e-5,5e-5]	[**1e-5**,2e-5,3e-5,5e-5]
dropout rate	[**0.1**,0.2,0.5]	[**0.1**,0.2,0.5]
batch size	[4,**8**,12]	[4,8,**12**]
activation function	[**ReLU**,SELU,GELU]	[ReLU,SELU,**GELU**]
training epochs	[3,**4**,5]	[3,4,**5**]
sequence length	[256,**512**]	[256,**512**]
♯ of encoder layers N	[3,4,5,**6**]	[3,4,5,**6**]
interpolation coefficient I	[12,**24**,32]	[12,24,**32**]
input embedding dim K	[64,**128**]	[64,**128**]
class weight	[0.5, **0.55**, 0.6, 0.65]	[0.0001,**0.0005**,0.001]

3.6.2 Baselines and Evaluation Metrics

We compare the performance of our model with the following six baselines where the first component (i.e., LSTM, BiLSTM, GRU) is commonly used for time series modeling and the second component (i.e., Word2Vec, FastText, ELMo) is commonly used for text representations in existing literature. Two components are integrated that support two modalities (i.e., time series and clinical notes).

- **LSTM + CNM** [72]
- **BiLSTM + CNM** [73]
- **GRU + CNM** [74]
- **PTSM + Word2Vec** [51]
- **PTSM + FastText** [75]
- **PTSM + ELMo** [76]

We evaluate our model performance in terms of area under the receiver operating characteristic curve (AUROC), F1 score, recall and precision, which are common metrics for imbalanced classification. In addition to the hyperparameters listed in Table 3.2, we fine-tune additional hyperparameters for LSTM (number of layers, hidden units), GRU (number of layers, hidden units), Word2Vec (window size, number of negative samples), FastText (maximum length of word n-gram, number of buckets), and ELMo (bidirectional and number of negative samples) as shown in Table 3.3. The hyperparameter values in bold indicate the optimal values found for our baseline models using both modalities. The number of negative samples is based on the negative sampling algorithm.

Table 3.3: Hyperparameter search space of baselines on both datasets. Bold values are the optimal values found using both modalities.

	Hyperparameters	**MIMIC-III**	**eICU-CRD**
LSTM	♯ of layers	[1,2,**3**,4]	[1,2,3,**4**]
	hidden units	[100,150,**200**]	[100,150,**200**]
	bidirectional	[**Yes**, No]	[**Yes**, No]
GRU	♯ of layers	[1,2,**3**,4]	[1,2,**3**,4]
	hidden units	[100,150,**200**]	[100,150,**200**]
Word2Vec	window size	[5,10,**20**]	[5,**10**,20]
	♯ of negative samples	[**10**,15,20]	[10,**15**,20]
FastText	max length of word n-gram	[2, **5**, 10]	[**2**, 5, 10]
	♯ of buckets	[**1000**,2000,3000]	[1000,**2000**,3000]
ELMo	bidirectional	[**Yes**, No]	[**Yes**, No]
	♯ of negative samples	[10, 15, **20**, 30]	[10, 15, 20, **30**]

3.6.3 Results

The results of our method and all baselines using both modalities on two datasets are shown in Table 3.4 (a) and 3.4 (b), respectively. We can see that our method outperforms all baselines on both datasets on all metrics regardless of hours we use. Compared to LSTM and GRU, PTSM benefited from its self-attention mechanism. Specifically, PTSM has direct access to all of the available data in parallel, which leaves no room for information loss. Furthermore, compared to Word2Vec and FastText, CNM (ClinicalBERT) provides dynamic contextualized word representations instead of static embeddings, which brings about flexible text representations. For ELMo, since it is based on BiLSTM, it may not be able to deal with long-term dependencies as well as Transformer-based CNM. In general, all models performed better when supplied with available MPTS data and clinical notes covering more hours.

Table 3.4: Performance comparison for the MIMIC-III and eICU-CRD datasets between the proposed method and six baselines. Hours represent all the data available including MPTS and clinical notes after admission. Experiments are conducted 5 times with different random seeds. The results are shown in the format of mean and standard deviation.

	Hours	12	18	24	30	36
Baseline 1: LSTM + CNM	AUROC	0.854 ± 0.009	0.867 ± 0.008	0.875 ± 0.008	0.878 ± 0.009	0.884 ± 0.008
	F1 Score	0.846 ± 0.006	0.852 ± 0.007	0.856 ± 0.007	0.857 ± 0.006	0.861 ± 0.006
	Precision	0.797 ± 0.006	0.799 ± 0.007	0.801 ± 0.007	0.802 ± 0.006	0.807 ± 0.006
	Recall	0.901 ± 0.006	0.913 ± 0.007	0.918 ± 0.007	0.921 ± 0.006	0.923 ± 0.006
Baseline 2: BiLSTM + CNM	AUROC	0.861 ± 0.008	0.869 ± 0.008	0.878 ± 0.009	0.886 ± 0.009	0.890 ± 0.008
	F1 Score	0.853 ± 0.009	0.858 ± 0.007	0.862 ± 0.008	0.865 ± 0.009	0.869 ± 0.008
	Precision	0.803 ± 0.009	0.808 ± 0.007	0.811 ± 0.008	0.813 ± 0.009	0.816 ± 0.008
	Recall	0.909 ± 0.009	0.914 ± 0.007	0.920 ± 0.008	0.924 ± 0.009	0.930 ± 0.008
Baseline 3: GRU + CNM	AUROC	0.849 ± 0.011	0.856 ± 0.012	0.864 ± 0.011	0.871 ± 0.012	0.876 ± 0.010
	F1 Score	0.842 ± 0.009	0.844 ± 0.010	0.848 ± 0.009	0.851 ± 0.011	0.853 ± 0.012
	Precision	0.795 ± 0.009	0.797 ± 0.010	0.802 ± 0.009	0.805 ± 0.011	0.806 ± 0.012
	Recall	0.896 ± 0.009	0.898 ± 0.010	0.900 ± 0.009	0.903 ± 0.011	0.906 ± 0.012
Baseline 4: PTSM + Word2Vec	AUROC	0.838 ± 0.008	0.851 ± 0.007	0.859 ± 0.007	0.863 ± 0.007	0.872 ± 0.008
	F1 Score	0.830 ± 0.009	0.836 ± 0.008	0.848 ± 0.007	0.851 ± 0.008	0.855 ± 0.009
	Precision	0.792 ± 0.009	0.794 ± 0.008	0.798 ± 0.007	0.800 ± 0.008	0.801 ± 0.009
	Recall	0.872 ± 0.009	0.882 ± 0.008	0.905 ± 0.007	0.910 ± 0.008	0.916 ± 0.009
Baseline 5: PTSM + FastText	AUROC	0.859 ± 0.007	0.868 ± 0.009	0.875 ± 0.012	0.883 ± 0.010	0.889 ± 0.009
	F1 Score	0.851 ± 0.008	0.854 ± 0.009	0.859 ± 0.007	0.861 ± 0.007	0.865 ± 0.009
	Precision	0.801 ± 0.008	0.804 ± 0.009	0.809 ± 0.007	0.811 ± 0.007	0.815 ± 0.009
	Recall	0.907 ± 0.008	0.911 ± 0.009	0.915 ± 0.007	0.918 ± 0.007	0.922 ± 0.009
Baseline 6: PTSM + ELMo	AUROC	0.863 ± 0.005	0.871 ± 0.007	0.880 ± 0.006	0.889 ± 0.007	0.892 ± 0.006
	F1 Score	0.854 ± 0.006	0.859 ± 0.007	0.863 ± 0.006	0.867 ± 0.008	0.870 ± 0.007
	Precision	0.805 ± 0.006	0.810 ± 0.007	0.814 ± 0.006	0.817 ± 0.008	0.819 ± 0.007
	Recall	0.910 ± 0.006	0.915 ± 0.007	0.918 ± 0.006	0.923 ± 0.008	0.928 ± 0.007
Ours: PTSM + CNM	AUROC	**0.902 ± 0.004**	**0.910 ± 0.005**	**0.917 ± 0.005**	**0.923 ± 0.004**	**0.928 ± 0.004**
	F1 Score	**0.881 ± 0.005**	**0.887 ± 0.006**	**0.894 ± 0.004**	**0.907 ± 0.005**	**0.910 ± 0.004**
	Precision	**0.839 ± 0.005**	**0.845 ± 0.006**	**0.852 ± 0.004**	**0.866 ± 0.005**	**0.869 ± 0.004**
	Recall	**0.928 ± 0.005**	**0.933 ± 0.006**	**0.940 ± 0.004**	**0.951 ± 0.005**	**0.955 ± 0.004**

(a) Comparison results on the MIMIC-III testing set.

	Hours	12	18	24	30	36
Baseline 1: LSTM + CNM	AUROC	0.796 ± 0.012	0.801 ± 0.010	0.816 ± 0.009	0.827 ± 0.010	0.830 ± 0.011
	F1 Score	0.787 ± 0.009	0.792 ± 0.008	0.794 ± 0.008	0.796 ± 0.009	0.798 ± 0.010
	Precision	0.773 ± 0.009	0.779 ± 0.008	0.782 ± 0.008	0.783 ± 0.009	0.786 ± 0.010
	Recall	0.802 ± 0.009	0.805 ± 0.008	0.806 ± 0.008	0.809 ± 0.009	0.810 ± 0.010
Baseline 2: BiLSTM + CNM	AUROC	0.802 ± 0.012	0.809 ± 0.011	0.825 ± 0.009	0.833 ± 0.009	0.851 ± 0.009
	F1 Score	0.790 ± 0.008	0.801 ± 0.009	0.813 ± 0.009	0.820 ± 0.008	0.827 ± 0.008
	Precision	0.778 ± 0.008	0.781 ± 0.009	0.785 ± 0.009	0.787 ± 0.008	0.794 ± 0.008
	Recall	0.802 ± 0.008	0.821 ± 0.009	0.844 ± 0.009	0.855 ± 0.008	0.863 ± 0.008
Baseline 3: GRU + CNM	AUROC	0.791 ± 0.007	0.800 ± 0.008	0.813 ± 0.008	0.824 ± 0.009	0.829 ± 0.008
	F1 Score	0.773 ± 0.008	0.782 ± 0.008	0.786 ± 0.007	0.793 ± 0.006	0.797 ± 0.007
	Precision	0.776 ± 0.008	0.780 ± 0.008	0.783 ± 0.007	0.787 ± 0.006	0.792 ± 0.007
	Recall	0.770 ± 0.008	0.784 ± 0.008	0.789 ± 0.007	0.799 ± 0.006	0.803 ± 0.007
Baseline 4: PTSM + Word2Vec	AUROC	0.787 ± 0.009	0.796 ± 0.009	0.811 ± 0.008	0.824 ± 0.010	0.834 ± 0.009
	F1 Score	0.784 ± 0.008	0.788 ± 0.007	0.804 ± 0.007	0.813 ± 0.008	0.821 ± 0.008
	Precision	0.778 ± 0.008	0.780 ± 0.007	0.784 ± 0.007	0.787 ± 0.008	0.792 ± 0.008
	Recall	0.791 ± 0.008	0.796 ± 0.007	0.824 ± 0.007	0.841 ± 0.008	0.852 ± 0.008
Baseline 5: PTSM + FastText	AUROC	0.814 ± 0.012	0.826 ± 0.011	0.838 ± 0.011	0.846 ± 0.010	0.852 ± 0.011
	F1 Score	0.807 ± 0.009	0.815 ± 0.010	0.822 ± 0.010	0.828 ± 0.009	0.834 ± 0.010
	Precision	0.802 ± 0.009	0.805 ± 0.010	0.808 ± 0.010	0.810 ± 0.009	0.813 ± 0.010
	Recall	0.813 ± 0.009	0.826 ± 0.010	0.837 ± 0.010	0.848 ± 0.009	0.856 ± 0.010
Baseline 6: PTSM + ELMo	AUROC	0.812 ± 0.008	0.821 ± 0.009	0.832 ± 0.007	0.844 ± 0.008	0.849 ± 0.009
	F1 Score	0.808 ± 0.007	0.815 ± 0.006	0.819 ± 0.007	0.824 ± 0.008	0.830 ± 0.007
	Precision	0.803 ± 0.007	0.807 ± 0.006	0.809 ± 0.007	0.811 ± 0.008	0.812 ± 0.007
	Recall	0.814 ± 0.007	0.823 ± 0.006	0.829 ± 0.007	0.837 ± 0.008	0.849 ± 0.007
Ours: PTSM + CNM	AUROC	**0.845 ± 0.006**	**0.852 ± 0.005**	**0.861 ± 0.005**	**0.873 ± 0.006**	**0.882 ± 0.004**
	F1 Score	**0.833 ± 0.005**	**0.840 ± 0.004**	**0.845 ± 0.004**	**0.851 ± 0.004**	**0.857 ± 0.003**
	Precision	**0.802 ± 0.005**	**0.807 ± 0.004**	**0.809 ± 0.004**	**0.814 ± 0.004**	**0.818 ± 0.003**
	Recall	**0.866 ± 0.005**	**0.875 ± 0.004**	**0.884 ± 0.004**	**0.892 ± 0.004**	**0.900 ± 0.003**

(b) Comparison results on the eICU-CRD testing set.

3.6.4 Ablation Analysis

To further study the influence of each individual component of our proposed method, we conduct ablation experiments to investigate the influence of individual model components with different data inputs. The results of ablation analysis on both datasets are presented in Table 3.5a and 3.5b, respectively. First, we consider the performance of applying MPTS data on PTSM only and clinical notes on CNM only. As can be seen from Table 3.5, in general, the model with input of solely MPTS data has better performance than that of solely clinical notes. Next, we utilize both data modalities with only hour 1 MPTS data (admission measurements) and available clinical notes within T hours since admission where $T = 12, 18, 24, 30, 36$. When using both modalities, they can bring about comparable results with those of using MPTS data only. Finally, in terms of our full model using full MPTS data and clinical notes, the performance improves with the available data covering more hours by a margin of $4.3\% - 8.5\%$ on AUROC, $4.1\% - 7.3\%$ on F1 score, $3.3\% - 9.1\%$ on precision, and $3.0\% - 7.9\%$ on recall compared with the "best" model performance when using partial data. The ablation analysis suggests that both MPTS data and clinical notes complement and benefit each other and thus the model with both modalities has better performance than the model with single modality.

3.6.5 Case Studies

We perform case studies to evaluate the uniqueness of each modality in which they may contain information that is inaccessible by the other modality. Figure 3.2 depicts four self-attention mechanisms in our model which help to understand patterns in the clinical notes. In all of the panels, the x-axis represents the query tokens and the y-axis represents the key tokens. In panels (a) and (b), we analyze the medical note "remain intubated and feel periodically very painful with back pain while awake during

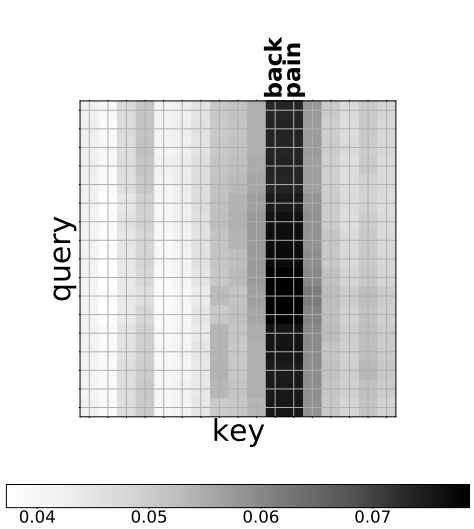
Table 3.5: Ablation analysis on the influence of different components in our model for the MIMIC-III and eICU-CRD datasets. Experiments are conducted 5 times with different random seeds. The results are shown in the format of mean and standard deviation. Note that hour 1 MPTS indicates that only initial measurements are considered as input instead of a series of measurements. Also, the case that hour 1 clinical notes (i.e. admission notes) with increasing available MPTS data is not considered since the available notes for each patient at the initial time is limited.

	Hours	12	18	24	30	36
MPTS on PTSM only	AUROC	0.827 ± 0.009	0.835 ± 0.008	0.839 ± 0.010	0.842 ± 0.007	0.846 ± 0.008
	F1 Score	0.822 ± 0.006	0.830 ± 0.007	0.831 ± 0.006	0.837 ± 0.008	0.838 ± 0.007
	Precision	0.777 ± 0.006	0.784 ± 0.007	0.771 ± 0.007	0.793 ± 0.008	0.778 ± 0.007
	Recall	0.872 ± 0.006	0.882 ± 0.007	0.900 ± 0.006	0.887 ± 0.008	0.907 ± 0.007
Clinical Notes on CNM only	AUROC	0.790 ± 0.008	0.797 ± 0.007	0.806 ± 0.008	0.812 ± 0.009	0.831 ± 0.007
	F1 Score	0.776 ± 0.007	0.789 ± 0.007	0.799 ± 0.009	0.804 ± 0.008	0.823 ± 0.007
	Precision	0.749 ± 0.008	0.740 ± 0.007	0.784 ± 0.009	0.782 ± 0.009	0.792 ± 0.007
	Recall	0.806 ± 0.007	0.846 ± 0.007	0.814 ± 0.009	0.828 ± 0.008	0.856 ± 0.007
Hour 1 MPTS on PTSM + CNM	AUROC	0.836 ± 0.011	0.839 ± 0.010	0.846 ± 0.008	0.862 ± 0.009	0.871 ± 0.009
	F1 Score	0.830 ± 0.009	0.831 ± 0.008	0.833 ± 0.008	0.847 ± 0.009	0.863 ± 0.010
	Precision	0.795 ± 0.009	0.787 ± 0.008	0.789 ± 0.008	0.794 ± 0.009	0.808 ± 0.010
	Recall	0.869 ± 0.009	0.880 ± 0.008	0.883 ± 0.008	0.907 ± 0.009	0.927 ± 0.010
Ours: PTSM + CNM	AUROC	**0.902 ± 0.004**	**0.910 ± 0.005**	**0.917 ± 0.005**	**0.923 ± 0.004**	**0.928 ± 0.004**
	F1 Score	**0.881 ± 0.005**	**0.887 ± 0.006**	**0.894 ± 0.004**	**0.907 ± 0.005**	**0.910 ± 0.004**
	Precision	**0.839 ± 0.005**	**0.845 ± 0.006**	**0.852 ± 0.004**	**0.866 ± 0.005**	**0.869 ± 0.004**
	Recall	**0.928 ± 0.005**	**0.933 ± 0.006**	**0.940 ± 0.004**	**0.951 ± 0.005**	**0.955 ± 0.004**

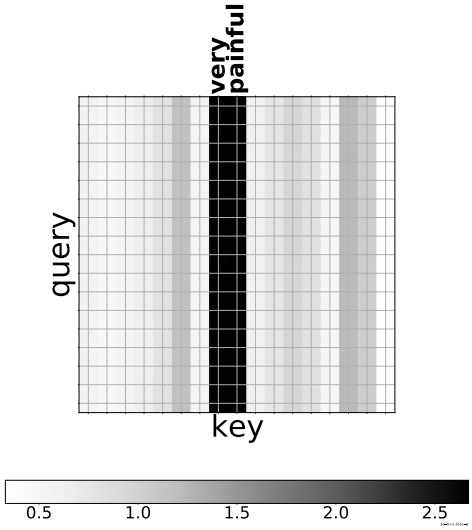
(a) Ablation analysis results on the MIMIC-III testing set.

	Hours	12	18	24	30	36
MPTS on PTSM only	AUROC	0.782 ± 0.006	0.788 ± 0.007	0.793 ± 0.008	0.796 ± 0.009	0.817 ± 0.008
	F1 Score	0.773 ± 0.005	0.776 ± 0.006	0.780 ± 0.006	0.781 ± 0.007	0.799 ± 0.006
	Precision	0.766 ± 0.005	0.750 ± 0.006	0.733 ± 0.007	0.727 ± 0.007	0.757 ± 0.006
	Recall	0.780 ± 0.005	0.803 ± 0.006	0.833 ± 0.006	0.844 ± 0.007	0.847 ± 0.006
Clinical Notes on CNM only	AUROC	0.724 ± 0.008	0.733 ± 0.010	0.748 ± 0.009	0.756 ± 0.007	0.778 ± 0.008
	F1 Score	0.717 ± 0.007	0.721 ± 0.006	0.736 ± 0.006	0.742 ± 0.007	0.761 ± 0.008
	Precision	0.695 ± 0.007	0.692 ± 0.007	0.704 ± 0.006	0.708 ± 0.007	0.719 ± 0.008
	Recall	0.740 ± 0.007	0.752 ± 0.006	0.771 ± 0.006	0.780 ± 0.007	0.808 ± 0.007
Hour 1 MPTS on PTSM + CNM	AUROC	0.794 ± 0.011	0.801 ± 0.009	0.814 ± 0.008	0.831 ± 0.009	0.846 ± 0.008
	F1 Score	0.787 ± 0.008	0.792 ± 0.007	0.805 ± 0.007	0.817 ± 0.008	0.823 ± 0.008
	Precision	0.765 ± 0.008	0.768 ± 0.007	0.777 ± 0.007	0.786 ± 0.008	0.792 ± 0.008
	Recall	0.811 ± 0.008	0.818 ± 0.007	0.836 ± 0.007	0.851 ± 0.008	0.857 ± 0.008
Ours: PTSM + CNM	AUROC	**0.845 ± 0.006**	**0.852 ± 0.005**	**0.861 ± 0.005**	**0.873 ± 0.006**	**0.882 ± 0.004**
	F1 Score	**0.833 ± 0.005**	**0.840 ± 0.004**	**0.845 ± 0.004**	**0.851 ± 0.004**	**0.857 ± 0.003**
	Precision	**0.802 ± 0.005**	**0.807 ± 0.004**	**0.809 ± 0.004**	**0.814 ± 0.004**	**0.818 ± 0.003**
	Recall	**0.866 ± 0.005**	**0.875 ± 0.004**	**0.884 ± 0.004**	**0.892 ± 0.004**	**0.900 ± 0.003**

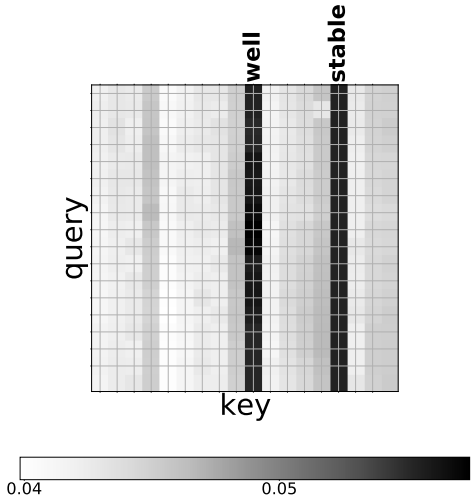
(b) Ablation analysis results on the eICU-CRD testing set.



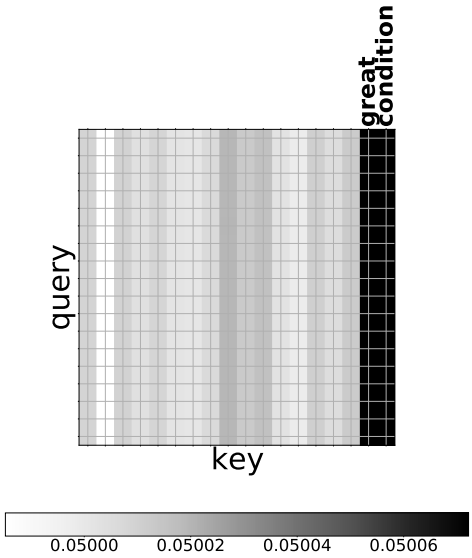
(a) patient with sepsis



(b) patient with sepsis



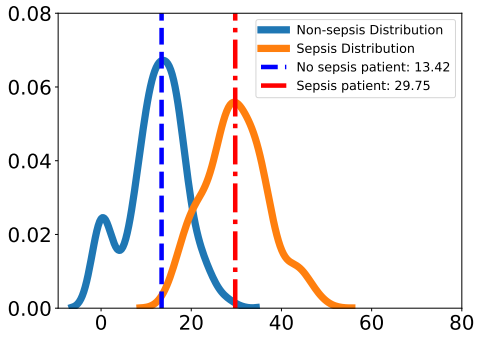
(c) patient without sepsis



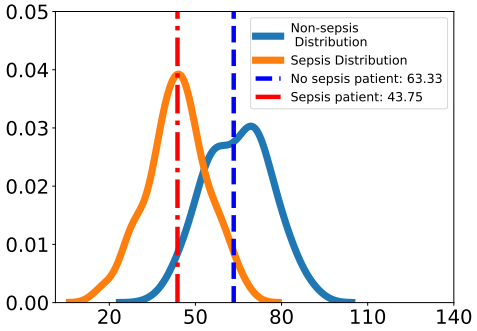
(d) patient without sepsis

Figure 3.2: ClinicalBERT attention mechanism visualization. The x-axis are the query tokens and the y-axis are the key tokens. Panels (a) and (b) are two head attention mechanisms for a patient. The input notes read "remain intubated and feel periodically very painful with back pain while awake during mechanical ventilation". Panels (a) and (b) extract "back pain" and "very painful" as prominent patterns from the two heads, respectively, which provides insight on the patient's critically ill condition. Similarly, panels (c) and (d) are two head attention mechanisms for a patient without sepsis. The input notes include "feel comfortable and tolerating cpap well and vital signs keep stable overall great condition". "Well" and "stable" stand out in panel (c) and "great condition" emerges in panel (d). All of those words are strong indications that the patient is in a relatively benign condition.

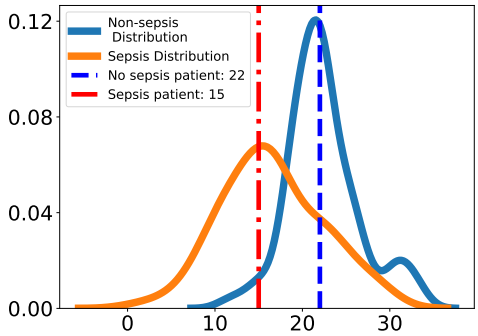
mechanical ventilation" from a patient with sepsis. Panels (a) and (b) are two different head attention mechanisms. Panel (a) indicates "back pain" and panel (b) extracts "very painful" as prominent patterns, respectively. Similarly, panels (c) and (d) are two head attention mechanisms for a patient that ends up with no sepsis. The input note is "feel comfortable and tolerating cpap well and vital signs keep stable overall great condition". CNM finds "well", "stable" and "great condition" in panels (c) and (d), respectively. Both "very painful" and "great condition" help in understanding the patients' conditions and strongly correlate with the final sepsis outcomes. The indications from extracted patterns to patient outcomes show the effectiveness of the ClinicalBERT representations for clinical notes.



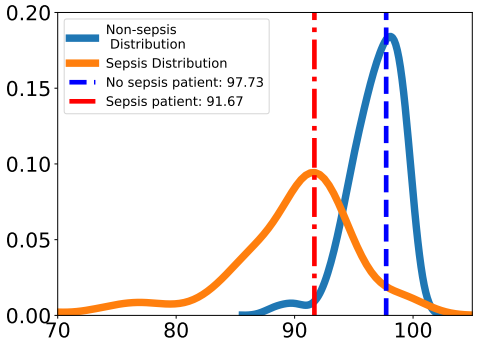
(a) Respiratory Rates



(b) Diastolic Blood Pressure



(c) Bicarbonate Levels



(d) Oxygen Saturation

Figure 3.3: Density plots of features. The blue and orange curves are density curves of corresponding features. The blue curve represents no sepsis, and the orange represents sepsis. The dashed vertical lines shows the two patients' feature values.

Then, we compare some physiological feature values from MPTS data in Figure 3.3, which plots the univariate distributions of selected features for sepsis and non-sepsis patients, respectively. The orange and blue curves are density curves of observed components of features. The orange curves represent the density curves of sepsis, and the blue ones represent no sepsis. Dashed vertical lines are two patients' corresponding measurement values, who were correctly classified by the proposed model (PTSM + CNM) while misclassified by CNM only. Case studies suggest that single modality does not contain all the possible information that benefits the final prediction. Consequently, using both MPTS data and clinical notes can help obtain more information, which is conducive to the better predictive performance of the model.

3.7 Conclusion

In this chapter, we incorporate multivariate physiological time series data and clinical notes with Transformer for early prediction of sepsis. Comprehensive experiments are conducted on two large critical care datasets, including baseline comparison, ablation analysis, and case studies. Our results demonstrate the effectiveness of our method when using both data modalities, which consistently outperforms competitive baselines on all metrics. Further analysis, and specifically to include clinicians' treatment measures in the input data, are worth exploring.

Chapter 4

Enhancing Transformer Efficiency for Multivariate Time Series Classification

4.1 Introduction

Time series (TS) data is ubiquitous, occurring in healthcare [77, 78], stock market [79], astronomy [80], and many other domains [81, 82]. With the advance of sensing techniques, TS classification across wide-ranging domains has gained much interest during the past decade [83, 84].

The availability of the UCR/UEA time series benchmark datasets [84] has led to an abundance of TS classification algorithms [85, 86, 87, 88, 89]. The classification accuracy has been the key metric used to evaluate existing methods [90]. However, the high accuracy of these algorithms often comes with the cost of high computational complexity [91]. From common preconceptions in natural language processing (NLP) and computer vision (CV), in order to achieve high accuracy, training top performing models

with millions/billions of parameters is a computationally intensive task, requiring days or weeks on many parallel GPUs or TPUs. However, such intensive training makes the model difficult to retrain for further improvement on performance. Likewise, for large-scale time series data with high dimensionality or long sequence length, it is challenging to maintain the balance between the predictive accuracy and training efficiency.

In this work, we propose a method to investigate the relationship between model efficiency and its effectiveness, as well as its complexity for MTS classification. The model architecture is based on Transformer and Fourier transform. We use 18 benchmark MTS datasets for evaluation. Comprehensive experiments are conducted on all datasets, including ablation study of each module of the network and module-by-module pruning in terms of accuracy, training speed, and model size. Experimental results demonstrate the competitive performance of our proposed architecture compared with current state-of-the-art methods. Ablation studies identify the main contributors to the predictive performance, such as multi-head self-attention and Fourier transform. In addition, module-wise pruning of the network reveals the trade-off between model efficiency and effectiveness, as well as model efficiency and complexity. Finally, we conduct Pareto analysis to examine the trade-off between efficiency and performance.

The main contributions of this chapter are highlighted as follows:

(1) To the best of our knowledge, this is the first chapter to perform Pareto analysis to investigate the relationship between efficiency and accuracy.

(2) Through module-by-module pruning, comprehensive experimental results indicate an evident trade-off between model efficiency and its effectiveness, as well as its complexity.

(3) We employ Pareto analysis to investigate the relationship between model efficiency and performance. Such analysis methods can provide general guidance for re-

searchers on how to select efficient model configurations, which can be applied to any model architecture.

The remainder of this chapter is organized as follows. Section 4.2 describes related work of Transformer and Fourier transform on time series analysis and existing methods on model efficiency improvement. The network architecture is outlined in Section 4.3. Section 4.4 discusses datasets and experiments on 18 benchmark datasets, including ablation studies, module-wise pruning and Pareto efficiency visualization. Finally, our conclusions are presented in Section 4.5.

4.2 Related Work

Neural Networks for Time Series Classification. Currently, most TS classification algorithms can be divided into three categories: feature-based [92], distance-based [93], and neural network based methods [83]. Here, we focus only on neural network based methods. Since the advancements of deep learning, two popular frameworks, CNN and RNN, are widely applied in TS classification tasks. [94] combined Fully Convolutional Networks (FCN) and Residual Networks (ResNet) for univariate time series classification. [95] developed a group-constrained method, which combines a CNN with an RNN. More recent works such as InceptionTime [96], TapNet [97], and TST [89] are proposed for TS classification. For additional deep learning methods, we refer readers to [83].

Fourier Transform in Time Series. The Fourier transform (FT) has been an important tool in time series analysis for decades [98], and is widely used for applications such as anomaly detection [99], periodicity detection [100], and similarity measures [101]. The FT converts a TS from time domain to frequency domain, and uses Fourier coeffi-

cients to represent the original data. For the TS classification task, FTs have been used indirectly in disparate applications. For instance, [102] utilizes the FT to filter noisy data for vegetation type classification, and [103] uses the FT as a feature extraction technique to classify electroencephalography (EEG) data. However, none of the above methods apply the FT directly to TS classification, particularly in the context of neural networks. In contrast, we aim to apply the discrete FT and its inverse as modules of a deep learning framework. The unparameterized FT can reduce the computational cost of the network to some extent.

Transformer Networks for Time Series Classification. With the exemplary performance of the Transformer architecture [64] in NLP and CV, researchers in the time series community began exploring Transformers in TS classification in specific domains [104, 63]. More recent works have generalized Transformer frameworks for MTS classification. [89] adopts a Transformer encoder architecture for unsupervised representation learning of MTS. [105] explored an extension of the current Transformer architecture by gating, which merges two towers for MTS classification. In contrast, we propose to generalize a mixing framework which utilizes both Transformer and FT. By replacing some self-attention sublayers with FT, the computational complexity can be reduced.

Model Training Efficiency. Due to the increasing size of both models and training data, many works have focused on improving model training efficiency through parameter reduction, such as DenseNet [106] and EfficientNet [107], training speed improvement including NFNets [108] and BotNet [109], or both [110]. One of the most common techniques to improve network efficiency is model pruning. Early works focused on non-structured methods. For instance, [111, 112] proposed to remove individual weight values. Recent works focused more on structured methods, such as channel weight pruning based

on l_1 norm [113].

4.3 Methodology

In this section, we present our network architecture, which contains all of the modules for potential model pruning. The overall model structure is illustrated in Figure 4.1.

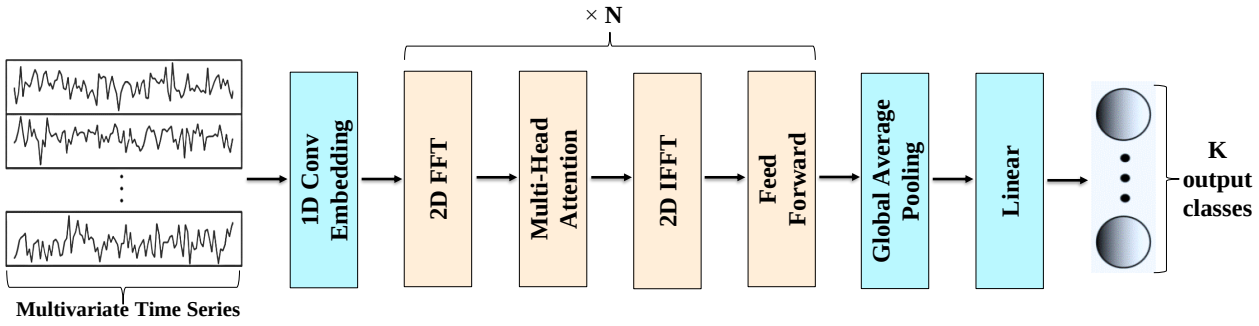


Figure 4.1: An overview of the full model framework. Our architecture is based on Transformer and Fourier transform. Following the sequence embedding, we apply a 2D discrete Fourier transform (particularly Fast Fourier transform) to convert the TS features from the time domain to the frequency domain, a multi-head self-attention layer, and a 2D inverse discrete Fourier transform to map the features back to the time domain. Then we employ a Global Average Pooling (GAP) layer to average the output of the MTS over the entire time dimension. Finally, a Softmax layer is used for the multi-class MTS classification task.

Input Embeddings. Input embeddings are commonly used in NLP models, which map relatively low-dimensional vectors to high-dimensional vectors to facilitate sequence modeling [66]. Correspondingly, an embedding for TS sequence is required to capture the dependencies among different features without considering the temporal information [58]. Our framework employs a 1D convolutional layer to obtain the K-dimensional embeddings at each time step.

Discrete Fourier Transform. The Fourier transform decomposes a function of time into its constituent frequencies. For clarity, we first consider the 1D Discrete Fourier

transform (DFT). Given a sequence of complex numbers $x(n)$ with $0 \leq n \leq N-1$, the 1D DFT is defined by

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} x(n) \cdot W_N^{kn},\ 0 \leq k \leq N-1,$$

where $W_N^{kn} = e^{-\frac{2\pi i}{N} kn}$. Given the DFT $X(k)$, the original sequence can be recovered by the inverse DFT (IDFT)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{\frac{2\pi i}{N} kn},\ 0 \leq n \leq N-1.$$

The 2D DFT is a direct extension of the 1D DFT, obtained by alternately performing the 1D DFT on the row and column dimensions. Given a 2D signal $x(m,n)$ with $0 \leq m \leq M-1, 0 \leq n \leq N-1$, the 2D DFT is given by

$$X(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) \cdot e^{-2\pi j(\frac{km}{M} + \frac{ln}{N})}.$$

Similar to the 1D IDFT, the 2D DFT is invertible via the 2D IDFT,

$$x(m,n) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} X(k,l) \cdot e^{2\pi j(\frac{km}{M} + \frac{ln}{N})}.$$

To compute the DFT efficiently, the Fast Fourier Transform (FFT) algorithm takes advantage of the periodicity and symmetry properties of W_N^{kn} such that the computational complexity of the DFT reduces from $O(N^2)$ to $O(NlogN)$, regardless of dimension.

Multi-head Attention. The multi-head attention (MHA) mechanism, the major component of the Transformer architecture [64], allows the model to jointly attend to information from different representation subspaces at different positions. MHA is defined

as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \cdots, \text{head}_h)W^O,$$

where $Q, K, V \in \mathbb{R}^{n \times d_{model}}$ are input embedding matrices, n is the sequence length, d_{model} is the embedding dimension, and h is the number of heads. Each head i is defined as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) = \text{softmax}\left(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_k}}\right)VW_i^V,$$

where $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}, W_i^O \in \mathbb{R}^{hdv \times d_{model}}$ are parameter matrices to be learned.

Global Average Pooling. Global average pooling involves calculating the average value of all of the elements in a feature map. It is mainly used to reduce the amount of learnable parameters.

Batch Normalization. Instead of using layer normalization in Transformer-related architectures in NLP, we consider the necessity of applying batch normalization to each block shown in Figure 3.1. Compared to layer normalization, batch normalization can mitigate the effect of outlier values in time series data, which does not appear in text representations.

Activation Function. Using the same activation function as the original Transformer architecture [64], we consider the necessity of applying the activation function *gelu* for each module shown in Figure 3.1.

Feedforward Neural Network. A position-wise feedforward neural network (FNN) is applied with two 1D convolutional layers with kernel size 1, and a *gelu* activation function in between.

4.4 Experiments

In this section, we describe benchmark MTS datasets [84] used for experimental evaluation, the experimental setup, and corresponding results.

4.4.1 Datasets

We select a set of 18 publicly available benchmark datasets from the UCR/UEA classification archive: AtrialFibrillation (AF), BasicMotions (BM), Cricket (CR), DuckDuckGeese (DDG), Epilepsy (EP), EthanolConcentration (EC), ERing (ER), FingerMovements (FM), HandMovementDirection (HMD), Handwriting (HW), Heartbeat (HB), Libras (LIB), NATOPS (NATO), PEMS-SF (PEMS), RacketSports (RS), SelfRegulationSCP1 (SRS1), SelfRegulationSCP2 (SRS2), and UWaveGestureLibrary (UW). The main characteristics of each dataset are summarised in Table 4.1. All of the datasets have been split into training and testing sets by default. Thus, there are no preprocessing steps for these data. The predictive performance on all datasets is evaluated in terms of accuracy.

4.4.2 Setup

We set aside 20% of the default training set for the validation set, which we used to select the best collection of hyperparameters. All experiments were implemented in Pytorch [70] on one GTX 1080 Ti GPU. We minimized the cross entropy loss with the Adam [71] optimizer for training. The hyperparameter search space for each dataset is listed in Table 4.2. Note that the batch size choice is limited by the available GPU memory.

Table 4.1: Summary of the 18 UCR/UEA datasets used in experimentation.

Dataset	Code	Train Size	Test Size	Dimensions	Length	Classes
AtrialFibrillation	AF	15	15	2	640	3
BasicMotions	BM	40	40	6	100	4
Cricket	CR	108	72	6	1197	12
DuckDuckGeese	DDG	50	50	1345	270	5
Epilepsy	EP	137	138	3	206	4
EthanolConcentration	EC	261	263	3	1751	4
ERing	ER	30	270	4	65	6
FingerMovements	FM	316	100	28	50	2
HandMovementDirection	HMD	160	74	10	400	4
Handwriting	HW	150	850	3	152	26
Heartbeat	HB	204	205	61	405	2
Libras	LIB	180	180	2	45	15
NATOPS	NATO	180	180	24	51	6
PEMS-SF	PEMS	267	173	963	144	7
RacketSports	RS	151	152	6	30	4
SelfRegulationSCP1	SRS1	268	293	6	896	2
SelfRegulationSCP2	SRS2	200	180	7	1152	2
UWaveGestureLibrary	UW	120	320	3	315	8

4.4.3 Module Settings

Based on Section 4.3, we define the following eight modules of the network for further analysis: input embedding (EMBED), fast Fourier transform (FFT), inverse fast Fourier transform (IFFT), multi-head attention (MHA), feedforward neural network (FFN), global average pooling (GAP), batch normalization (BN), and activation function (ACT). The corresponding abbreviations of each module are shown in parentheses.

4.4.4 Ablation Study

First, we conduct ablation studies to analyze the contributions of each module on the predictive performance. The contribution of each module is obtained when a module is removed from the full network while other modules remain intact. The fine-tuned results on 18 datasets are shown in Table 4.3. Starting from Column 4, the smaller the accuracy

Table 4.2: Hyperparameter search space of the model on each dataset. If the number of layers of a module is equal to 0, then this module is removed in the pruned model.

Hyperparameters	Search Space
learning rate	[1e-3, 5e-3, 1e-4, 5e-4, 1e-5, 5e-5]
dropout rate	[0.1, 0.2, 0.3]
batch size	[8, 16, 32]
♯ of heads	[4, 8, 16]
♯ of FFT layers	[0, 1, 2, 3, 4]
♯ of IFFT layers	[0, 1, 2, 3, 4]
♯ of MHA layers	[0, 1, 2, 3, 4]
♯ of Feedforward layers	[0, 1, 2, 3, 4]

is, the larger the module's contribution is, and vice versa. The accuracy of each dataset for the unpruned model (Table 4.3 Column 3) is competitive with current state-of-the-art methods [84]. Among eight modules, it can be seen that MHA and FFT contribute most to the predictive performance on 10 out of the 18 datasets and 9 out of the 18 datasets, respectively. For MTS data, the correlations between different dimensions across all time steps are important to consider. Hence, the MHA is able to catch different feature correlations, and influence the accuracy to a large extent. The FFT, as the core of signal processing and more generalized time series, extracts frequency information embedded in data, which provides a more straightforward representation compared to the original data in the time-domain. In contrast, we observe that EMBED, BN, and ACT contribute least to the predictive performance on 11 out of the 18 datasets, 5 out of the 18 datasets, and 13 out of the 18 datasets, respectively. Although these operations are important for the training of the model, they influence the testing accuracy marginally compared with MHA and FFT.

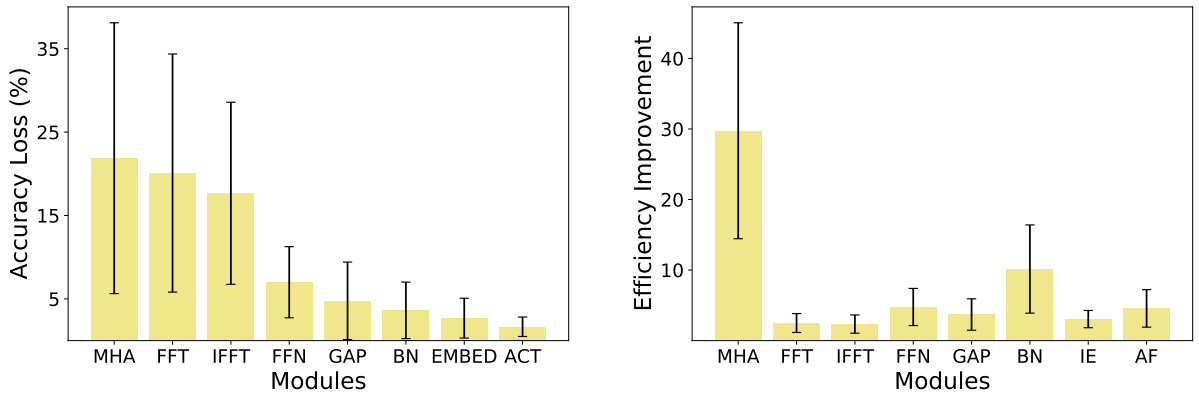
Table 4.3: Ablation study in the testing accuracy loss on 18 datasets by removing each module at a time while leaving others the same. Each experiment is conducted 5 times with different random seeds. The results are shown in the format of mean and standard deviation. Column 2 shows the accuracy of the full model with all modules included. Columns 3 to 10 represent the accuracy when the module in that column is removed from the model. Bold indicates that the module contributes most to the loss in accuracy and underlining indicates that the module contributes least to the loss in accuracy when the module is removed.

Dataset	Acc.	Unpruned	EMBED	FFT	IFFT	MHA	FFN	GAP	BN	ACT
AF	Mean	0.667	0.600	**0.400**	0.467	**0.400**	0.667	0.533	0.600	0.667
	Std.	0.003	0.005	0.005	0.004	0.003	0.006	0.006	0.004	0.003
BM	Mean	0.975	0.950	**0.725**	0.775	0.750	0.900	0.925	0.900	0.950
	Std.	0.008	0.010	0.012	0.009	0.012	0.010	0.014	0.009	0.011
CR	Mean	0.987	0.958	0.875	0.861	**0.833**	0.889	0.944	0.972	0.944
	Std.	0.007	0.009	0.012	0.008	0.012	0.006	0.009	0.012	0.008
DDG	Mean	0.580	0.580	0.440	0.420	**0.380**	0.520	0.560	0.560	0.580
	Std.	0.016	0.017	0.020	0.016	0.014	0.016	0.016	0.014	0.016
EP	Mean	0.986	0.978	**0.891**	0.913	0.899	0.949	0.971	0.956	0.971
	Std.	0.014	0.013	0.016	0.014	0.014	0.012	0.014	0.013	0.015
EC	Mean	0.456	0.445	0.376	0.395	**0.365**	0.418	0.441	0.445	0.452
	Std.	0.003	0.002	0.003	0.003	0.004	0.002	0.004	0.003	0.002
ER	Mean	0.963	0.956	0.896	0.889	**0.885**	0.892	0.948	0.952	0.956
	Std.	0.006	0.007	0.006	0.006	0.008	0.005	0.006	0.007	0.005
FM	Mean	0.640	0.620	**0.490**	0.520	0.500	0.600	0.590	0.610	0.620
	Std.	0.009	0.008	0.007	0.008	0.010	0.008	0.009	0.010	0.011
HMD	Mean	0.486	0.446	0.365	0.351	**0.338**	0.406	0.459	0.432	0.473
	Std.	0.018	0.016	0.020	0.017	0.018	0.019	0.018	0.016	0.020
HW	Mean	0.529	0.514	0.471	0.473	**0.468**	0.506	0.506	0.512	0.514
	Std.	0.006	0.007	0.006	0.005	0.007	0.007	0.008	0.007	0.006
HB	Mean	0.771	0.766	**0.683**	0.707	0.688	0.751	0.756	0.766	0.756
	Std.	0.014	0.015	0.014	0.017	0.015	0.016	0.014	0.015	0.016
LIB	Mean	0.917	0.906	**0.822**	0.827	0.839	0.889	0.894	0.906	0.911
	Std.	0.009	0.011	0.012	0.010	0.012	0.013	0.011	0.009	0.010
NATO	Mean	0.844	0.833	**0.728**	0.739	0.750	0.772	0.811	0.833	0.833
	Std.	0.005	0.004	0.005	0.007	0.006	0.005	0.006	0.004	0.006
PEMS	Mean	0.908	0.884	0.815	0.809	**0.803**	0.867	0.879	0.896	0.896
	Std.	0.013	0.012	0.014	0.016	0.014	0.013	0.013	0.014	0.012
RS	Mean	0.914	0.901	**0.796**	0.816	0.803	0.855	0.908	0.901	0.908
	Std.	0.021	0.020	0.020	0.018	0.019	0.021	0.020	0.021	0.019
SRS1	Mean	0.915	0.894	0.836	0.823	**0.819**	0.853	0.887	0.894	0.901
	Std.	0.005	0.007	0.006	0.006	0.005	0.007	0.006	0.005	0.005
SRS2	Mean	0.600	0.594	**0.522**	0.533	0.516	0.578	0.583	0.588	0.594
	Std.	0.002	0.003	0.002	0.001	0.004	0.002	0.003	0.003	0.002
UW	Mean	0.922	0.906	0.844	0.850	**0.841**	0.875	0.894	0.897	0.903
	Std.	0.006	0.008	0.009	0.006	0.007	0.008	0.006	0.007	0.007

To clearly demonstrate the influence of each module on the predictive performance and efficiency of the network, the averaged testing accuracy loss and the corresponding efficiency improvement for each module (compared with the unpruned model) over all datasets are presented in Figure 4.2. Here, efficiency is defined as the product of training time per epoch and the amount of learnable parameters. The higher the product, the lower the efficiency is. In consideration of highly diversed datasets with respect to sequence length, number of samples, and dimensionality, the average loss in accuracy for each module demonstrates a high variance from Figure 4.2a as the performance loss extent can vary depending on dataset characteristics. The modules MHA, FFT, and IFFT demonstrate a notable influence on the model performance on average (21.9%, 20.1%, and 17.7% loss in accuracy respectively). For modules like BN, EMBED and ACT, removing them bring about minimal accuracy loss compared to other modules (3.6%, 2.7%, and 1.6% respectively). Meanwhile, comparing Figure 4.2a and Figure 4.2b, the module which has larger impact on the predictive performance does not indicate that removing it can bring about more efficiency improvement. For instance, the computationally inexpensive FFT influences the predictive performance to a large extent. In contrast, although the computational cost of BN is high, its contribution to the performance is marginal.

4.4.5 Module-by-Module Pruning

Next, we explore the relationship between efficiency (defined the same as Section 4.4.4) and effectiveness (predictive performance). Based on the contribution of each module on the performance loss shown in Figure 4.2a, we perform module-by-module pruning by following the order of modules from the most significant contributor to the least significant contributor (MHA, FFT, IFFT, FFN, GAP, BN, EMBED, ACT) to accuracy. We evalu-

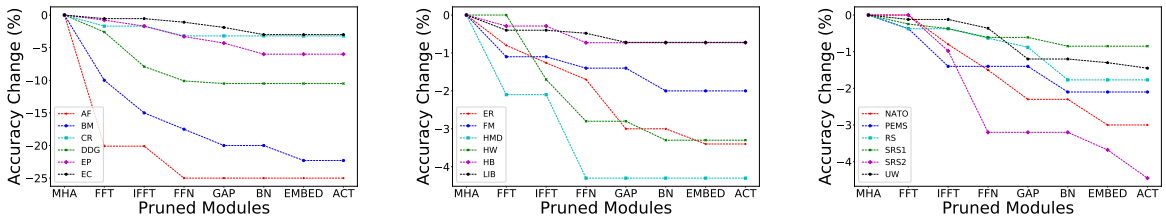


(a) (b)

Figure 4.2: (a) represents the average testing accuracy loss across all datasets while removing one module at a time and other modules remain in the network. Modules MHA, FFT, and IFFT bring about larger influence on the predictive performance due to the high percentage of accuracy loss when removing them. In comparison, BN, EMBED, and ACT bring about marginal influence on the predictive performance compared with other modules. (b) represents the corresponding average efficiency improvement across all datasets when one module is removed from the network while other modules keep intact.

ate such pruning effect in two aspects: (1) effectiveness: testing accuracy; (2) efficiency: average training time per epoch in seconds and the number of learnable parameters. Due to limited space, we only show some datasets' testing accuracy in Table 4.4 and their efficiency results in Figure 4.4. We observe that after removing the entire MHA module, the number of learnable parameters shrinks drastically, so as the accuracy (Table 4.4 Column 4). The representation capability of the pruned network, which has fewer parameters, is damaged since the amount of parameters is a key aspect to the network representation. Furthermore, the pace of accuracy loss and parameter reduction removal of subsequent modules slows down as FFT/IFFT has no learnable parameters. For the remaining modules, the number of parameters they carry is much fewer than the MHA module. Based on Figure 4.2a, their effects on the predictive performance are moderate. Hence, the curves in Figure 4.4 are relatively flat following MHA. We further investigate

the extent of change in accuracy of module-wise pruning on all datasets, as shown in Figure 4.3. We notice that the performance variation in different datasets vary widely. For datasets such as AF, BM, and DDG, the model pruning has a great impact on their performance. This may be due to very limit amount of training samples. Conversely, for datasets like HB, LIB, and SRS1, the model pruning brings little effect after removing the MHA module (within 1%).



(a) (b) (c)

Figure 4.3: Change in accuracy (%) from module-by-module pruning across all datasets. The order of datasets shown from (a) to (c) correspond to Table 4.3.

Table 4.4: Module-wise pruning results of datasets EC, NATO, FM and SRS1. The results from Column 3 (MHA) to Column 10 (AF) with regard to accuracy represent that the module in that column is removed from the model architecture. Experiments are conducted 5 times with different random seeds. The accuracy results are shown in the format of mean and standard deviation. Bold represents that the module brings about much accuracy loss compared to the unpruned model. Following MHA, the accuracy decreasing trend remains stable.

Dataset	Acc.	Unpruned	MHA	FFT	IFFT	FFN	GAP	BN	IE	AF
EC	Mean	0.456	**0.365**	0.363	0.363	0.361	0.358	0.354	0.354	0.354
	Std.	0.003	0.004	0.004	0.002	0.004	0.003	0.003	0.003	0.003
NATO	Mean	0.844	**0.750**	0.750	0.744	0.739	0.733	0.733	0.728	0.728
	Std.	0.005	0.006	0.003	0.004	0.006	0.005	0.006	0.004	0.005
FM	Mean	0.640	**0.500**	0.495	0.495	0.493	0.493	0.490	0.490	0.490
	Std.	0.009	0.010	0.011	0.010	0.008	0.009	0.011	0.010	0.011
SRS1	Mean	0.915	**0.819**	0.817	0.816	0.814	0.814	0.812	0.812	0.812
	Std.	0.005	0.005	0.003	0.003	0.0.004	0.006	0.003	0.004	0.005

Overall, based on the above module-by-module pruning scheme, we observe that as

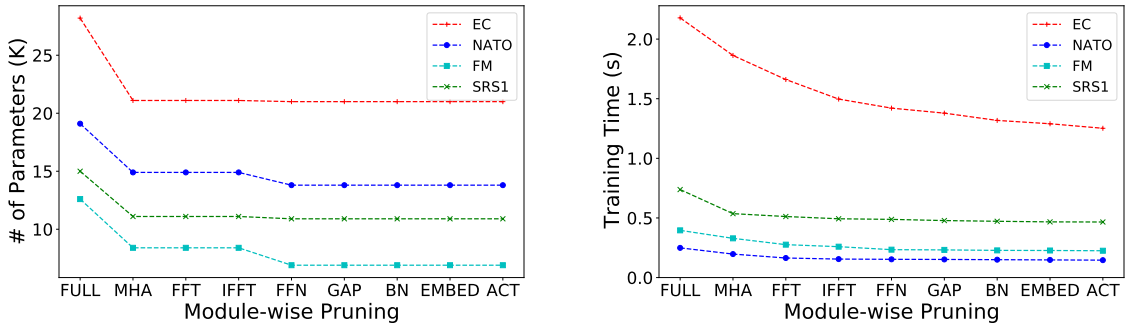


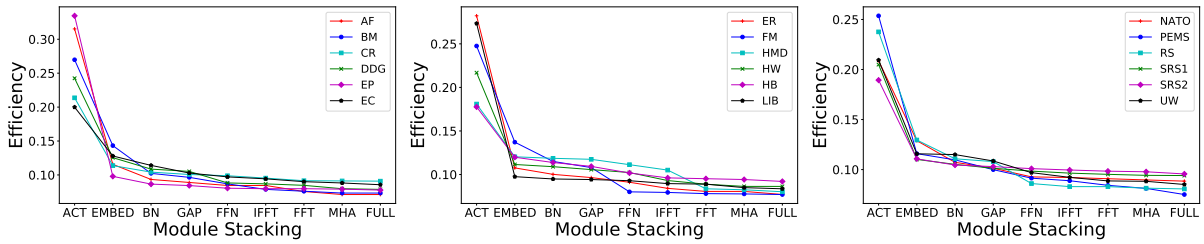
Figure 4.4: Module-wise results for changes in terms of number of parameters and training time per epoch on four datasets: EC, NATO, FM, SRS1.

the effectiveness (predictive performance) of the network increases, the corresponding efficiency (training speed and model size) generally decreases. The evident cost–benefit trade-off between efficiency and effectiveness provides a key question to researchers on how to find efficient model settings while maintaining the "equilibrium" between these two aspects. This problem will be discussed in Section 4.4.7.

4.4.6 Efficiency vs. Complexity

Here, we explore the relationship between network efficiency and complexity. In general, the more complex a model is, the less efficient it is. The network's efficiency is defined in the same way as previous sections, in terms of the training time and the number of parameters. Meanwhile, we define the complexity of the model as the stacking of modules. Contrary to model pruning, we stack each module based on their influence on the predictive performance, from the least significant contributor to the most significant contributor (ACT, EMBED, BN, GAP, FFN, IFFT, FFT, MHA) to accuracy. Our empirical results in Figure 4.5 shed light on the trade-off between model efficiency and complexity. As can be seen in Figure 4.5, as more modules are stacked over the

network, the corresponding computational efficiency decreases. All datasets illustrate similar trends.



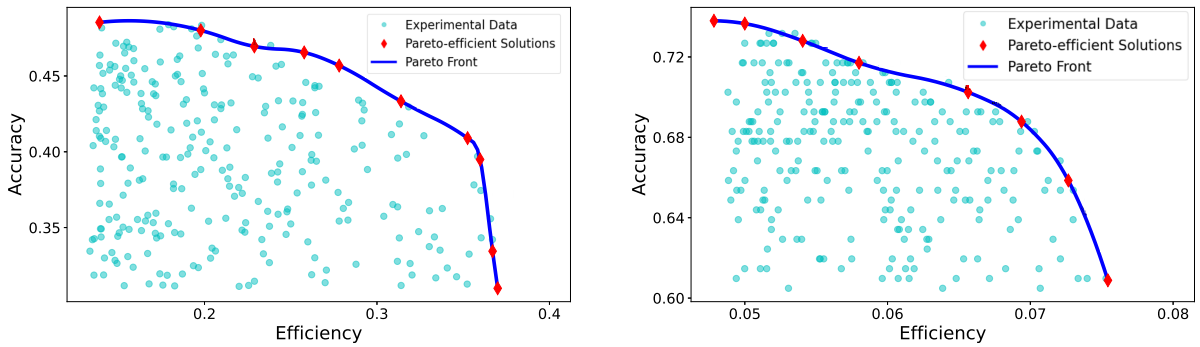
(a) (b) (c)

Figure 4.5: Trade-off between network efficiency and complexity across all datasets. Due to the notable differences of dataset sizes, the computation of efficiency is normalized for each dataset. The order of datasets shown from (a) to (c) correspond to Table 4.3.

4.4.7 Pareto Analysis for Trade-off Exploration between Efficiency and Performance/Effectiveness

We define the model efficiency in terms of the reciprocal of the product between training time per epoch and the number of parameters. Thus, the higher the reciprocal, the higher the efficiency. To explore the relationship between model efficiency and performance, we employ Pareto analysis [114]. Pareto efficiency represents a state for which improving the performance as measured by one criterion would worsen the performance as measured by another criterion. We choose the *FingerMovements* and *Heartbeat* datasets to obtain the Pareto frontiers, where the set of points on the front correspond to Pareto-efficient solutions. We have two objectives: (1) maximize the efficiency; (2) maximize the accuracy. Figure 4.6 shows the result of Pareto fronts for both datasets in blue, where the red points are Pareto-efficient solutions. The scattered cyan points are randomly sampled experimental data from all different configurations. The Pareto analysis provides us with a principled approach for choosing efficient network settings, while

exploring the trade-off between efficiency and performance. Specifically, we can identify the extent of computational resources that is required in order for a model to achieve a certain performance. Conversely, we can identify how well a model can perform, given a certain amount of resources.



(a) *FingerMovements*. (b) *Heartbeat*.

Figure 4.6: Pareto efficiency visualization of the *FingerMovements* and *Heartbeat* datasets. The scattered cyan points, the marked red points, and the blue curve represent randomly sampled experimental data, Pareto-efficient solutions, and Pareto efficient frontiers.

4.5 Discussion

In this work, we propose a methodology to investigate the relationship between model efficiency and effectiveness, as well as its complexity. The method is performed on a mixing network based on Transformer and Fourier transform for MTS classification. Extensive experiments are conducted on 18 MTS datasets, including ablation studies on different modules of the network, module-by-module pruning evaluated in terms of the predictive performance, training speed, and the number of learnable parameters. The network achieves competitive performance compared to current best-performing methods. Ablation studies indicate that self-attention and Fourier transform are the largest contributors that influence the model performance across all datasets. Furthermore,

through sequential pruning of each module, we observed the efficiency–effectiveness and the efficiency–complexity trade-offs of the network. Through Pareto analysis, we show how to choose efficient settings of the network, while investigating the performance–efficiency trade-off through visualization of the Pareto fronts. We note that for far more complex models applied to large-scale data, due to finite computational resources, it is not practical to consider all possible configurations of the model and perform experiments. In these cases, given a reasonable number of experiments, techniques like regression can be used to generate massive random model settings and corresponding model performance. Pareto analysis can then be performed to evaluate the efficiency-performance trade-off, to guide researchers to adjust the model settings to improve the efficiency and effectiveness accordingly.

Chapter 5

Predicting the Need for Blood Transfusion in Intensive Care Units with Reinforcement Learning

5.1 Introduction

In critically ill patients, anemia and coagulopathy are common and associated with poor outcomes, such as increased risk of mortality, myocardial infarction, and thrombosis [115]. Transfusion of blood and blood products remains a cornerstone of anemia and coagulopathy treatment in critical care. Clinically, physicians make transfusion decisions mainly based on a patient's hemoglobin level and symptoms of anemia. However, due to urgency in the Intensive Care Units (ICU), physicians may not able to comprehensively evaluate all indicators of a patient such as demographics (e.g., age, weight, etc.), medical history (e.g., high blood pressure, diabetes, etc.), and laboratory parameters (e.g., creatinine, hemoglobin, etc.), which can play significant roles in the properness of making a decision about transfusion at a certain time [116]. However, inappropriate decisions

on blood transfusion such as the dosage and the type of blood product may even deteriorate the patient status. Thus, developing efficient decision support tools is critical to determining optimal treatment strategies in terms of the need for blood transfusion of each patient and improving the patients' clinical outcomes such as improved survival rates [63].

The majority of recent works considered the problem of blood transfusion prediction as a binary classification task [117, 118], i.e., whether the patient will require a blood transfusion during hospitalization. In practice, however, the transfusion decision that a physician makes at time t, based on patients' current situation, can influence the patients' subsequent condition and hence the physician's decision at time $t + 1$. Such dynamic status change makes blood transfusion a sequential decision making problem rather than purely a classification task.

In this work, we explore the use of an off-policy batch reinforcement learning (RL) algorithm, namely, the discretized Batch Constrained Q-learning (BCQ) [119] with different state representations and reward functions to provide clinical decision support for the need of blood transfusion for ICU patients. Specifically, we consider transfusion of three common types of blood products: red blood cells (RBC), platelets (PLT), and fresh frozen plasma (FFP). We use two critical care datasets: the Medical Information Mart for Intensive Care III (MIMIC-III) [45] and the UCSF. In order to evaluate the performance of the learned policy from sequential patient observations, our experiments are fourfold.

First, we use weighted importance sampling (WIS) [120] for off-policy evaluation. Second, we compare the RL policy recommendations against the true policies implemented by the hospital regarding whether the patient should receive blood transfusion at each time step using observed patient trajectories along with undertaken interventions. This is based on the assumption that physicians are knowledgeable and experienced to

make wise transfusion decisions. Third, we integrate TL to the RL algorithm to improve the original learned policy on the UCSF dataset (target domain) in terms of WIS and accuracy using the knowledge from the MIMIC-III dataset (source domain). Finally, we investigate about how can RL agents assist physicians further optimize real-time treatment strategies on blood transfusion based on the fact that transfusion does not always improve patients' clinical outcomes [121]. We conduct policy simulations from transferred RL policies to illustrate that blending the RL with what physicians follow could lead to better transfusion strategies and improving patients' short-term (decreased acuity scores) and long-term (decreased mortality) clinical outcomes on the UCSF dataset.

The main contributions of this chapter are highlighted as follows:

(1) To the best of our knowledge, this is the first chapter to use RL-based approach with different patient state encoding and reward function designs to deal with the blood transfusion policy recommendations in real-world critical care datasets.

(2) Experimental results show that TL, together with RL, can improve the transfusion policy learning on the data-scarce UCSF dataset using external knowledge from the MIMIC-III dataset. Specifically, compared to performances without TL, the matching accuracy between the learned policy and the true hospital policy improves up to 17.02%. Furthermore, the improvements of jump-start and asymptotic performances in WIS are up to 18.94% and 21.63%, respectively.

(3) Simulations from transferred RL policies on the UCSF dataset demonstrate both improved short-term and long-term clinical outcomes of ICU patients. Concretely, the overall estimated 28-day mortality rate reduces by 2.74% and the decreased acuity rate during patients' hospital stay reduces by 1.18% compared to the ground truth UCSF transfusion policies.

The remainder of this chapter is organized as follows. Section 5.2 describes related work. The preliminary background is briefly discussed in Section 5.3. Section 5.4 describes the datasets we use for evaluation. The methods we use are outlined in Section 5.5. Experiments and results are discussed in Section 5.6. Finally, our conclusions are presented in Section 5.7.

5.2 Related Work

In this section, we review related work on broad applications of RL in healthcare domains, TL approaches and applications in the context of deep RL, as well as existing methods for blood transfusion prediction.

5.2.1 RL in healthcare

RL is popular paradigm for solving sequential decision making problems with sampled, evaluative and delayed feedback simultaneously, and applies broadly in many disciplines, including games [122], robotics control [123], and biological data analysis [124]. Such distinctive features of RL make it a suitable candidate for developing dynamic treatment regimes (DTRs) that may improve the long-term outcome of patients. For example, cancer treatment, a naturally sequential evolutionary process, is a major objective of RL for DTR application. Various RL techniques have been applied to improve different aspects of cancer treatment [125]. For other DTR applicatons such as HIV treatment [126], sepsis treatment [127], and the need for mechanical ventilation [128], we refer readers to [129] for a comprehensive survey of applications of RL techniques in healthcare domains.

5.2.2 TL in deep RL

With the broad prospects of deep RL in different domains, TL has become an important technique to deal with various challenges faced by deep RL, which aims at accelerating the learning process and improving the performance of RL agents by transferring knowledge from external expertise. A significant volume of literature on a wide variety TL approaches in the context of deep RL focused on different aspects of transferring knowledge such as reward shaping [130], transfer from demonstrations [131], policy transfer [132], and inter-task mapping [133]. Recent years have witnessed the remarkable progress that TL combined with deep RL techniques. Such an integration has achieved notable success in applications such as robotics control [134] and game playing [135]. It also demonstrates promising prospects in domains like health informatics [136] and transportation systems [137].

5.2.3 Blood Transfusion Prediction

In recent years, researchers have exploited the use of machine learning (ML) methods on the problem of blood transfusion prediction. Supervised learning methods such as logistic regression [138], extreme Gradient Boosting (XGBOOST) [116], random forests [117] or neural networks [118], are applied to predict a binary decision: whether or not a patient will need a transfusion during the hospital stay. Unsupervised switching state autoregressive models on vital signs [139] are trained to predict whether transfusion was performed at each one-hour interval of the patient's stay in the hospital. All previous works formulated the blood transfusion prediction as a classification task. In contrast, we propose to use deep RL methods with different state representations of patients and reward functions, in combination with TL techniques, to directly provide sequential treatment recommendations for blood transfusion, and improve ICU patients' clinical outcomes.

5.3 Preliminaries

In this section, we briefly introduce the typical RL problem formulation via Markov decision process (MDP) and value-based deep RL, and TL from the RL perspective.

5.3.1 RL and MDP Formulation

RL studies sequential decision making processes, generally framed in terms of MDP. A MDP is a 5-tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where each element is defined as follows:

(1) $\mathcal{S}$: a finite state space that the patient is in state $s_t \in \mathcal{S}$ at time t.

(2) $\mathcal{A}$: a finite action space that the RL agent takes action $a_t \in \mathcal{A}$, which influences the next state s_{t+1}.

(3) $p(s_{t+1}|s_t, a_t)$: the dynamics of the system, which is the probability of the next state given the current state and action.

(4) $r(s_t, a_t, s_{t+1})$: the immediate reward after the transition from s_t to s_{t+1} due to action a_t.

(5) $\gamma \in [0, 1]$: the discount factor, which relates the rewards to the time domain and determines the relative weight in the distant future relative to those in the immediate future.

The purpose of a RL agent is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$, i.e., a mapping from a given state $s \in \mathcal{S}$ to a distribution over actions, that maximizes the expected accumulated reward:

$$R^{\pi}(s_t) = \lim_{T \to \infty} \mathbb{E}_{s_{t+1}|s_t,\pi} \sum_{t=1}^{T} \gamma^t r(s_t, a_t, s_{t+1})$$

over time horizon T.

5.3.2 Value-based Deep RL

Value-based deep RL methods are used when we use a deep neural network to approximate the value function. A standard algorithm, deep Q-Network (DQN) [140], uses a deep convolutional neural network architecture for optimal action-value (known as Q) function approximation. During learning, the Q-learning [141] update is applied:

$$\mathcal{L}(\theta) = L_\kappa\Big(r_{t+1} + \gamma \max_{a_{t+1}} Q_{\theta'}(s_{t+1}, a_{t+1}) - Q_\theta(s_t, a_t)\Big),$$

where L_κ is the Huber loss [142]:

$$L_\kappa(\delta) = \begin{cases} \frac{1}{2}\delta^2 & \text{if } |\delta| \leq \kappa, \\ \kappa(|\delta| - \frac{1}{2}\kappa) & \text{otherwise.} \end{cases}$$

The target network $Q_{\theta'}$ are updated infrequently, where θ' is updated to θ after a set number of learning steps. The Huber loss is minimized over replay buffer [143]. In healthcare settings, the dataset is fixed, and there are no further interactions with the environment (here, the patient). Hence, the off-policy batch-mode deep RL fits naturally.

5.3.3 TL in the Context of RL

Given one or more source domains $\mathcal{M}_s$ and one target domain $\mathcal{M}_t$, TL aims to optimize a policy from π to π^* for the target domain $\mathcal{M}_t$ by leveraging exterior knowledge $\mathcal{D}_s$ from $\mathcal{M}_s$, as well as interior knowledge $\mathcal{D}_t$ from $\mathcal{M}_t$. Here, $\pi = \phi(\mathcal{D}_s \sim \mathcal{M}_s, \mathcal{D}_t \sim \mathcal{M}_t)$, which is a function mapping $\mathcal{S}^t \rightarrow \mathcal{A}^t$ from the states to actions for the target domain $\mathcal{M}_t$. In our problem setting, we have $|\mathcal{M}_s| = |\mathcal{M}_t| = 1$ and knowledge can transfer between two RL agents within the same domain.

5.4 Datasets

Our patient cohorts are constructed from two datasets: the MIMIC-III (v1.4) and the UCSF dataset. MIMIC-III is a freely available single-center database of critical care data from over 58,000 hospital admissions, including information on 46,520 patients from Beth Israel Deaconess Medical Center between 2001 and 2012. The UCSF dataset, collected from the San Francisco General Hospital and Trauma Center, contains 2,190 highest level trauma activation patients admitted to the Level I trauma center. Both datasets contain de-identified data, including patient demographics, time-stamped measurements from bedside monitoring of vitals, clinical laboratory test results, as well as diagnosis and observations charted by healthcare providers. Both datasets were preprocessed in the same way, including the cohort selection criteria, raw data preprocessing, action space, and reward designs.

5.4.1 Cohort Selection

From both datasets, we first selected adult patients over the age of 18. Then, patients with less than 24-hour ICU stay or more than 168-hour ICU stay were excluded such that we could focus on patients where transfusion status was likely to impact recovery. After filtering by these criteria, we obtained a final cohort of 15,418 and 2,190 patients for the MIMIC-III and the UCSF datasets, respectively. Summary statistics of patient cohorts are summarized in Table 5.1. Here, for each blood transfusion task, patients are considered in two groups: (1) get transfusions at least once during hospitalization (Trans.); (2) do not receive any transfusion during hospitalization (No Trans.).

Table 5.1: Dataset Statistics.

Intervention	MIMIC-III Trans. / No Trans.	UCSF Trans. / No Trans.
RBC transfusion	8199 / 7219	1572 / 618
PLT transfusion	1772 / 13646	1160 / 1030
FFP transfusion	3215 / 12203	1358 / 832

5.4.2 Data Preprocessing

For each patient, we chose vital signs (e.g., heart rate, body temperature, respiration rate) and lab values (e.g., creatinine, hemoglobin, arterial pH) commonly reviewed by clinicians that change over time. Vital signs such as heart rate and temperature are taken several times within an hour, while laboratory tests such as arterial pH and creatinine are administered every few hours as needed. Following [144], this wide discrepancy in measurement frequency for time-varying continuous features is consolidated into means at 4-hour intervals. A list of clinically reasonable measurement ranges provided by [68] is used to remove outlier values for each feature. For the remaining missing values, we applied MICE [145] data imputation. After imputation, each feature's raw data is preprocessed independently by z-scoring across all patients such that the resulting data of each column has zero mean and unit variance. In addition, we extracted some demographic features (e.g., age at admission, admitting weight, gender) for each patient. All demographic features with static values of extracted patient cohorts were fully present. In total, we extracted 42 (4 static and 38 time-varying) features from the MIMIC-III dataset and 38 (9 static and 29 time-varying) features from the UCSF dataset.

5.4.3 Action Space

We define a binary action space for the need of transfusion in a 4-hour window. The action $a_t \in \mathcal{A}$ at each time step is chosen from $a_t \in \{0, 1\}$, which indicates having

the patient receiving the transfusion or not (1 indicates presence of transfusion, and 0 indicates absence of transfusion). This discrete action space is suitable for the transfusion of all three blood products (RBC, PLT, and FFP). We choose to use this action space due to the complexity and variations of patients' conditions in clinical practice, and its common definition in existing literature for the blood transfusion prediction task.

5.4.4 Reward Design

Our discrete reward functions are defined in two different ways.

(1) $R1$: Since patients' survival is physicians' major objective in critical care, we used the long-term clinical outcome, 28-day mortality status to define the reward. At the terminal time step of each patient's trajectory, we assign a positive reward $+10$ to patients who survived 28 days after ICU admission and a negative reward -10 as a penalty for those who were deceased before 28 days after ICU admission. For all intermediate time steps (including the starting time step), the rewards are all assigned to 0 since final outcomes of patients are unknown before therapeutic procedures ended.

(2) $R2$: During patients' hospitalization, in addition to their final survival, we value short-term outcomes after some treatments by observing an improvement or deterioration of patient status. The acuity score computed at each time step is used to estimate patients' severity of illness and reflect patients' conditions. We use the Sequential Organ Failure Assessment (SOFA) Score [146], a common acuity score, which is suitable to assess both critically ill ICU patients and trauma patients. For the starting time step, we assign the reward 0 for each patient since the status changed cannot be observed. For all intermediate time steps, the reward function

is defined as follows:

$$r_{t+1} = \begin{cases} +1 & \text{if } s_{t+1}^{SOFA} < s_t^{SOFA}, \\ -1 & \text{if } s_{t+1}^{SOFA} > s_t^{SOFA}, \\ 0 & \text{otherwise.} \end{cases}$$

This reward function penalizes increasing SOFA scores from s_t to s_{t+1} (deteriorated conditions). If SOFA scores decrease from s_t to s_{t+1}, a positive reward is assigned (improved conditions). Otherwise, there is no change in patients' condition, and a reward 0 will be given. At the terminal time step, we use the 28-day mortality to design the reward, which follows the same way as the first reward definition.

5.5 Methods

In this section, we present the overall framework with three phases, including (I) **representation phase** (patient state representations), (II) **learning phase** (disretized BCQ), and (III) **transfer phase** (Q-value transfer and weight transfer from the MIMIC-III to the UCSF). The framework overview is shown in Figure 5.1.

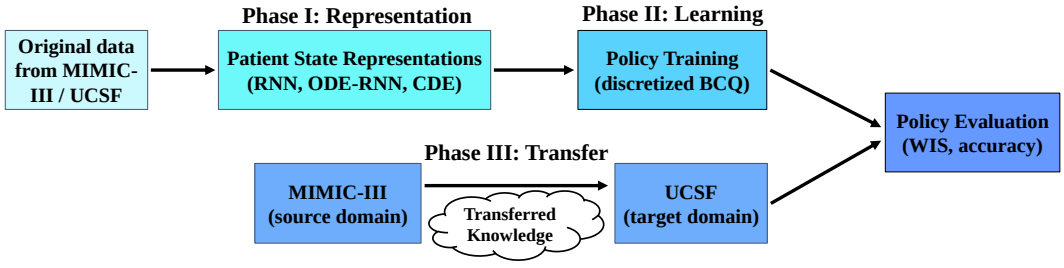
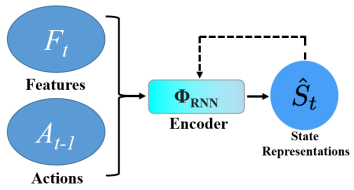


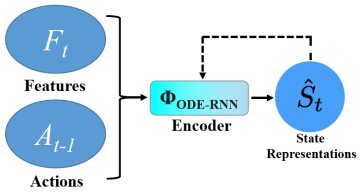
Figure 5.1: An overview of the three-phase framework.

5.5.1 Patient State Representations

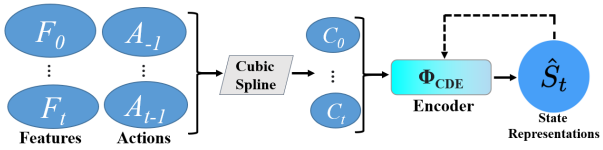
We represent patient states via an encoder architecture on both datasets. With a batch of observed patient trajectories, containing transitions between temporal data at time t and $t+1$ with treatment action at time t, as well as static demographics, an encoding function $\Phi : \mathcal{F}_{0:t}, \mathcal{A}_{0:t-1} \rightarrow \hat{S}_t$ is required to learn the patient state representation. Here, $\mathcal{F}_{0:t}$ represents all feature values from admission to time t and $\mathcal{A}_{0:t-1}$ represents actions taken from admission to time $t-1$. Three recurrent architectures are used to represent patients on both datasets, including basic Recurrent Neural Network [147] (RNN), generalized RNN with Ordinary Differential Equations [148] (ODE-RNN) and neural Controlled Differential Equations [149] (CDE). These approach architectures used to construct state representations are depicted in Figure 5.2.



(a) RNN encoder architecture.



(b) ODE-RNN encoder architecture.



(c) CDE encoder architecture.

Figure 5.2: Three recurrent architectures for patient state representations.

RNN: The RNN processes variable-length sequences by utilizing a recurrent hidden state, which is activated by features propagated from the previous time step. In our setting, given the current feature value F_t, it is concatenated with the previous action A_{t-1}, and passed into the RNN along with the hidden state representation from the previous time step $\hat{S}_{t-1}$, resulting in the current hidden state representation $\hat{S}_t$.

ODE-RNN: An ODE-RNN generalizes RNNs to have continuous-time hidden dynamics defined by ODEs. The main difference between ODE-RNN and basic RNN is that the recurrent hidden state updates based on an ODE between feature observations instead of being fixed.

CDE: A neural CDE is the continuous analogue of an RNN. It is similar to ODE-RNNs in terms of the temporal modeling dynamics by parameterizing the time derivative of a hidden state. Different from ODE-RNN, the hidden states in CDEs evolve smoothly as a function of time. A natural cubic spline interpolation is used to achieve continuous dependency on the data throughout the entire latent trajectory. Then, the network operates on pre-computed cubic spline coefficients instead of real feature values. At $t = 0$, the value for the initial latent space is is computed by a linear map on the inputs.

5.5.2 Discretized Batch Constrained Q-learning

We train RL policies on each of the learned state representations described in Section 5.5.1. We seek to learn a policy to select the optimal transfusion action using the state representation: $A_t \sim \pi(\hat{S}_t|\mathcal{F}_{0:t}, \mathcal{A}_{0:t-1})$. In our setting, we learn a policy via an off-policy batch RL algorithm, namely, discretized BCQ [119]. This method overcomes the issue of extrapolation errors, which occur in standard off-policy algorithms such as DQN. The discretized BCQ mainly include a state-conditioned model G_ω, a Q-network Q_θ, and a target Q-network $Q_{\theta'}$. The model G_ω is constructed using behavior cloning,

which is trained in the way of supervised learning with cross-entropy loss. The Q-network Q_θ is updated as follows:

$$\mathcal{L}(\theta) = L_\kappa\left(r_{t+1} + \gamma \max_{a_{t+1}|G_\omega(a_{t+1}|s_{t+1})/\max_{\hat{a}} G_\omega(\hat{a}|s_{t+1})>\tau} Q' - Q\right)$$

where $Q' = Q_{\theta'}(s_{t+1}, a_{t+1})$ and $Q = Q_\theta(s_t, a_t)$. Here, the threshold τ is used to select actions with higher probability, a constraint for state-action pairs. The final policy learned is based on the greedy behavior of τ:

$$\pi(s) = \operatorname*{argmax}_{a_t|G_\omega(a_t|s_t)/\max_{\hat{a}} G_\omega(\hat{a}|s_t)>\tau} Q_\theta(s_t, a_t).$$

See [119] for a more detailed algorithm description.

5.5.3 Transfer discretized BCQ

In consideration of the amount of data in our proprietary and public datasets, we explore the use of TL from the MIMIC-III (expert model) to the UCSF (learner model). We consider two ways of transfer: Q-value transfer and weight transfer.

Q-value Transfer (BCQ-QVT): The expert model learns its policies using the discretized BCQ algorithm, and selects actions based on its own Q-values. The learner model uses the Q-values from the expert model to adjust its network parameters in the direction that may attain better performance than before and help speed up convergence. The Q-values from the expert model are used in the loss function to guide the learner model, resulting in the following formulation of the loss function:

$$\mathcal{L}(\theta) = L_\kappa\left(r_{t+1} + \gamma \max_{a_{t+1}|G_\omega(a_{t+1}|s_{t+1})/\max_{\hat{a}} G_\omega(\hat{a}|s_{t+1})>\tau} Q' - Q + Q''\right)$$

where $Q' = Q_{\theta'}(s_{t+1}, a_{t+1})$, $Q = Q_{\theta}(s_t, a_t)$, and $Q'' = Q_{\exp,\theta''}(s_t, a_t)$ (Q-values from the expert model with parameter set θ'').

Weight Transfer (BCQ-W): We consider either retraining layers or a combination of retraining and re-initializing layers during transfer. Specifically, retraining layers involves initializing layers with the weights of a pre-trained policy and continuing to update these weights with backpropagation. Re-initializing layers involves randomly initializing the weights for a layer, rather than using the pre-trained weights.

5.6 Experiments and Results

Our experiments explore: (1) RL off-policy evaluation via WIS, (2) the matching accuracy between RL policy recommendations and ground truth actions performed by the hospital, (3) a combination of TL and RL from the MIMIC-III to the UCSF, (4) policy simulation from the transferred RL policy on the UCSF dataset. All the reported results and analysis through the remainder of this Section are provided using only the testing subset of the patient cohort.

Table 5.2: Hyperparameter search space for tuning of classification algorithms on both datasets.

	Tuning parameters	Search Space
LR	Inverse of regularization strength	$[1e{-}3, 1e{-}2, 1e{-}1, 1, 10, 100, 1000]$
RF	Number of trees in the forest	$[25, 50, 75, 100, 125, 150, 175, 200]$
	Maximum depth of the tree	$[2, 5, 8, 10, 12, 15, 20]$
XGBOOST	Number of trees in the forest	$[25, 50, 75, 100, 125, 150, 175, 200\}$
	Maximum depth of the tree	$\{2, 5, 8, 10, 12, 15, 20]$
MLP	Hidden layer size	$[16, 32, 64, 128, 256, 512]$
	Batch size	$[8, 16, 32, 64, 128, 256]$
	Activation function	$[\text{ReLU}, \tanh, \text{Sigmoid}]$
	Optimizer	[SGD, Adam]
	Learning rate	$[1e{-}4, 1e{-}3, 1e{-}2, 1e{-}1]$

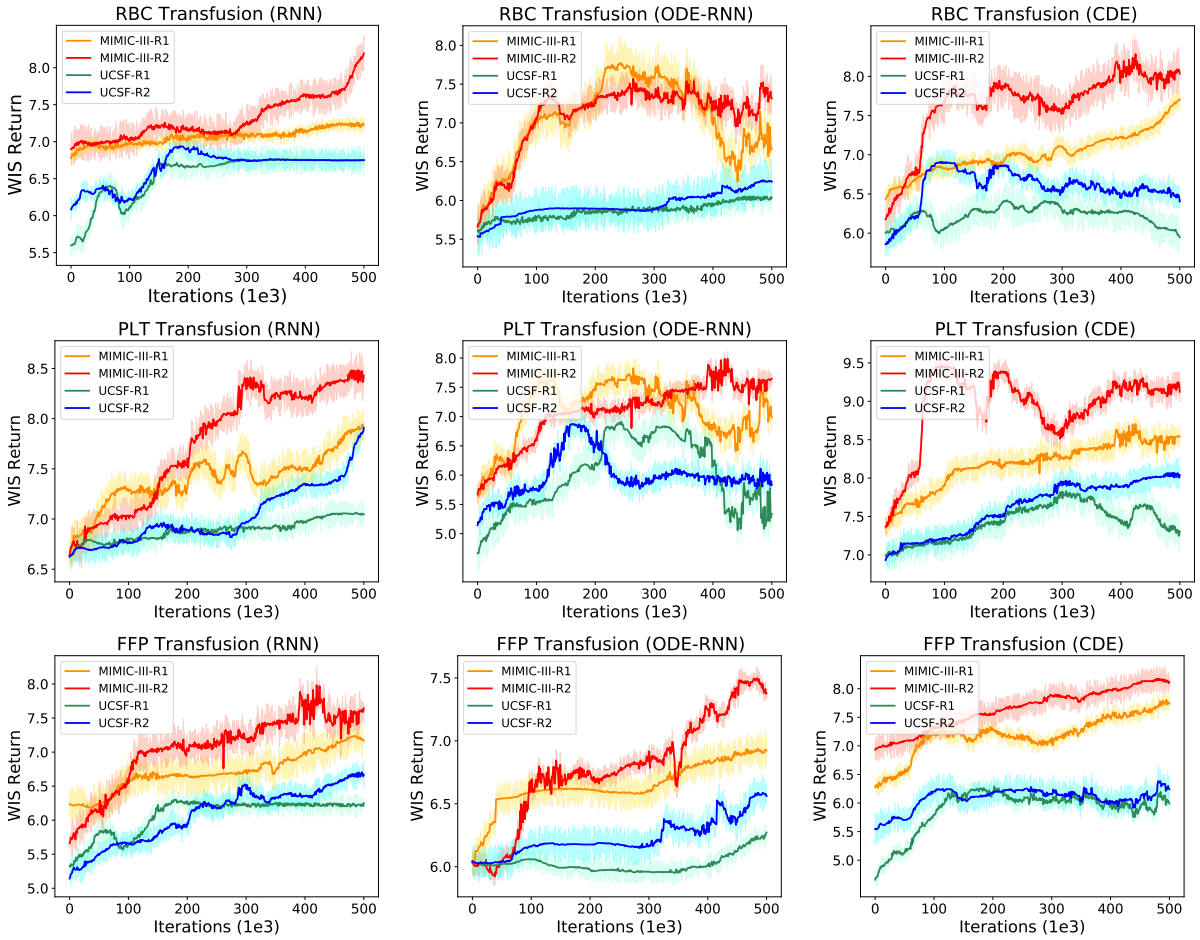


Figure 5.3: WIS evaluation of policies (learning curves) with 3 representation approaches (RNN, ODE-RNN, and CDE) and 2 reward functions on three transfusion tasks. All policies are trained from a replay buffer comprised of the training batch of patient trajectories for 500k iterations, evaluating the learned policy every 1000 iterations on the testing set of both datasets. The displayed results are averaged over 5 random seeds. The shaded area measures a single standard deviation across seeds. Across all tasks, representations with RNN and CDE along with the R2 reward function generally have better policy learning curves compared to those using ODE-RNN and the R1 reward function. Furthermore, the policy learned on the MIMIC-III dataset is far more superior to the UCSF dataset according to the WIS return value and the stability of learning curves.

Table 5.3: Hyperparameter search space for tuning of discrete BCQ algorithm on both datasets.

Tuning parameters	Search Space
Number of nodes per layer in Q-network	[32, 64, 128]
Batch size	[8, 16, 32, 64, 128, 256, 512]
Optimizer	[SGD, Adam]
Discount factor γ	[0.97, 0.975, 0.98, 0.985, 0.99, 0.995]
Target Q-network update frequency	[1k, 2k, 4k, 8k training iterations]
Learning rate	[1e−6, 5e−6, 1e−5, 5e−5, 1e−4, 5e−4, 1e−3, 5e−3]
Threshold τ	[0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]
Huber loss κ	[0.8, 0.9, 1.0, 1.1, 1.2]

5.6.1 Experimental Settings

Training Infrastructure: All of the experiments regarding state representations and policy learning were implemented in Pytorch on one NVIDIA Tesla P100 GPU.

Data Splitting: For both datasets, we randomly split the data in the ratio 70 : 15 : 15. That is, 70% data will go to the training set, 15% to the validation set and remaining 15% to the test set.

Evaluation Metrics:

(I) ***Accuracy***: When evaluating the closeness of a match between actions taken by RL agents and ground truth actions performed by physicians, we measured the performance in terms of accuracy. In this respect, we also consider a few classification models for comparison, which are commonly used in blood transfusion prediction tasks, including logistic regression (LR), random forest (RF), eXtreme Gradient Boosting (XGBOOST), and multi-layer perceptron (MLP). All experiments of these classification algorithms were implemented using scikit-learn.

(II) ***WIS***: When evaluating off-policy learning from the RL perspective, we use WIS, a means to correct the mismatch between the probabilities of a trajectory under

the behavior and target policies (learned policy using discretized BCQ), which is computed by: $R^{WIS} = \frac{\sum_{n=1}^{N} R_n w_n}{\sum_{n=1}^{N} w_n}$, where w_n is the per-trajectory importance sampling weight, a fraction between the target policy π and the behavior policy μ, and R_n is the empirical outcome of trajectory n. Here, the behavior policy used in WIS was behavior cloning, a 2-layer fully connected (FC) network with ReLU activation functions in between. It was trained with cross-entropy loss.

Architecture Details for Patient State Representations:

(I) ***RNN***: A 3-layer RNN is used to estimate the encoding function Φ_{RNN}. The first two layers are FC layers with ReLU activation functions, followed by a gated recurrent unit layer. The dimension of the hidden state was selected by grid search among $\{32, 64, 128, 256\}$.

(II) ***ODE-RNN***: A gated recurrent unit with 100 units is used to estimate the encoding function $\Phi_{\text{ODE-RNN}}$. The hidden states between feature observations are modeled by a neural ODE, parameterized by a 2-layer MLP with 100 hidden units. The adaptive step size is using the fifth-order *dopri5* solver from the torchdiffeq package.

(III) ***CDE***: To estimate the encoding function Φ_{CDE}, a neural CDE is used, parameterized by a MLP with three hidden layers, each of which has 100 hidden units. For the hidden layers, we use ReLU activation functions, and a tanh activation function for the final layer.

In order for efficient transfer of discretized BCQ from the MIMIC-III to the UCSF, the output vector dimension is the same when we construct patient representations on both datasets.

Policy Training: The discretized BCQ algorithm is used to train the policies. In our

BCQ, the Q-network is a 3-layer FC network. We used a uniformly sampled replay buffer for training, which comprised of the training batch of patient trajectories for 500k iterations. Then, we evaluated the learned policy every 1000 iterations using the testing subset of the data.

Hyperparameters: The hyperparameter search spaces for classification algorithms and discretized BCQ policy training are listed in Table 5.2 and Table 5.3, respectively. All three transfusion tasks use the same set of search space on both datasets. We perform grid search for hyperparameter optimization.

5.6.2 Off-Policy Evaluation

We evaluate the learned policy from the discretized BCQ algorithm with WIS. Specifically, for each transfusion task, we consider the influence of state representations and reward mechanisms on policy learning. Figure 5.3 presents all combinations of 3 state presentations and 2 reward mechanisms on both datasets for each task. Based on our reward design, the magnitude of the WIS value should lie in the ± 10 possible range. Suppose that the behavior policy is close to the actual hospital policy, which can be considered as "experts" most of the time. Then, a higher WIS value corresponds to more closeness between the learned policy and the behavior policy, indicating that the learned policy is more effective. From Figure 5.3, it is evident that regardless of the RL setting (i.e., different encoders and rewards), the learned policies on the MIMIC-III dataset far outperform the ones learned on the UCSF dataset, primarily due to differences in batch sample sizes. Furthermore, the performance of transfusion policies is generally superior when using R2 mechanism to the ones using R1 mechanism based on the policy learning curves in Figure 5.3. Finally, the learned policies from patient state representations encoded by RNN and CDE outperform ODE-RNN. The learning curves

of RNN and CDE demonstrate an overall steady growth with some small oscillations. For ODE-RNN, however, the learning curves show more volatility. Overall, oscillations and drops in performance during the intermediate iterations signify the impacts of encoded state representations, reward design mechanisms, and batch sample sizes on the policy learning of offline RL agents.

Table 5.4: Accuracy comparison on the testing subsets between actions taken by RL agents / classification models and ground truth actions implemented by the hospital. Experiments are conducted with 5 random initializations. The results are shown in the format of mean and standard deviation. Note that RNN-R1 represents the setting using the RNN state representation with reward function R1. Symbols for other settings in the descretized BCQ algorithm are similar.

		RBC transfusion		***PLT transfusion***		***FFP transfusion***	
		MIMIC-III	UCSF	MIMIC-III	UCSF	MIMIC-III	UCSF
Discretized BCQ Algorithm	RNN-R1	0.82 ± 0.02	0.67 ± 0.03	0.89 ± 0.03	0.70 ± 0.02	0.85 ± 0.01	0.71 ± 0.02
	RNN-R2	0.84 ± 0.01	0.68 ± 0.02	0.90 ± 0.02	0.73 ± 0.03	0.90 ± 0.02	0.72 ± 0.03
	ODE-RNN-R1	0.80 ± 0.03	0.63 ± 0.02	0.86 ± 0.02	0.68 ± 0.02	0.87 ± 0.01	0.70 ± 0.02
	ODE-RNN-R2	0.81 ± 0.02	0.63 ± 0.01	0.87 ± 0.01	0.70 ± 0.02	0.86 ± 0.01	0.70 ± 0.02
	CDE-R1	0.84 ± 0.02	0.69 ± 0.02	0.89 ± 0.02	0.72 ± 0.01	0.83 ± 0.02	0.71 ± 0.02
	CDE-R2	**0.85 ± 0.02**	0.71 ± 0.01	0.89 ± 0.01	0.73 ± 0.02	0.90 ± 0.02	0.71 ± 0.01
Classification Algorithms	LR	0.73 ± 0.02	0.69 ± 0.01	0.76 ± 0.02	0.75 ± 0.02	0.79 ± 0.01	0.73 ± 0.01
	RF	0.84 ± 0.01	0.80 ± 0.02	0.89 ± 0.01	0.87 ± 0.01	0.91 ± 0.01	**0.84 ± 0.02**
	XGBOOST	**0.85 ± 0.02**	**0.82 ± 0.01**	**0.92 ± 0.01**	0.88 ± 0.02	**0.93 ± 0.02**	0.83 ± 0.01
	MLP	0.82 ± 0.03	0.79 ± 0.02	0.89 ± 0.01	**0.89 ± 0.02**	0.90 ± 0.02	0.82 ± 0.01

5.6.3 Degree of Matching between RL agents and Physicians' Decisions

In addition to evaluating the learned policy using standard RL off-policy evaluation via WIS, we further compare the discrete BCQ algorithm's recommendations with respect to blood transfusion against the true policy implemented by the hospital. In addition, 4 classification models (LR, RF, XGBOOST, and MLP) are considered for accuracy comparison. From the RL perspective, in consideration of the performance influence using different state representations and reward designs, we consider all combinations

of representation approaches and reward design mechanisms and evaluate their performance. We report all the results on three transfusion tasks, as summarized in Table 5.4. For the MIMIC-III dataset, some settings of the discretized BCQ algorithm (RNN-R2 and CDE-R2) regarding action recommendations achieved comparable performance to most classification algorithms. Comparing performances across all RL settings in terms of state representations and rewards, policies learned from representations encoded by RNN and CDE outperform those using ODE-RNN, indicating that the representations from ODE-RNN did not adequately encode sufficient information to learn a policy from the batch mode, perhaps due to limited data. Furthermore, using R2 mechanism generally performs better than using R1 mechanism. This may be primarily due to the fact that R2 mechanism is more clinically guided such that it may reflect patients' real-time condition change better than using R1 mechanism. For the UCSF dataset, the accuracy performance of RL agents far underperform classification models regardless of the RL setting, mainly due to its tiny data size, making the RL agent more difficult to learn. A potential solution using knowledge from external expertise such as the MIMIC-III to overcome this issue will be discussed in Section 5.6.4.

5.6.4 Transfer RL

Due to the poor performance of policy learning on the UCSF dataset, we use external knowledge from MIMIC-III to improve its performances in terms of WIS and accuracy. As discussed in Section 5.5.3, we consider three types of transfer and evaluate their performance: BCQ-QVT, BCQ-WT (retraining all layers without re-initializing any layers), and BCQ-WTR (re-initializing the FC layers and retraining all layers). Based on the results from Section 5.6.2 and Section 5.6.3, we only consider the RL settings using R2 mechanism with state representations RNN and CDE. Results of policy learning

on the UCSF dataset with and without TL are presented in Figure 5.4 and Table 5.5. All three transfer methods yielded better performance than the original policy learning on the UCSF dataset in terms of WIS and accuracy. Specifically, from Figure 5.4, all transfer methods show better jump-start performance (i.e., the initial performance of the agent) and asymptotic performance (i.e., the ultimate performance of the agent) compared to the original policy learning curves without TL. In particular, across all three tasks, the jump-start WIS return improves up to 18.94% and the asymptotic WIS return improves up to 21.63% on average. For some transfusion tasks such as FFP transfusion with RNN-R2 and PLT transfusion with RNN-R2, some transfer methods can help reduce the oscillations, resulting in more steady growth learning curves. For some other tasks, transfer methods like BCQ-WT even bring about larger oscillations, indicating more unstable policy learning compared to the performances without TL. This may be due to weights of different scales between two datasets and pre-trained policies from the MIMIC-III dataset may get stuck in local optima. In Table 5.5, among three transfusion tasks, the degree of matching between the transferred policy and the ground truth policy has significantly improved to different extents. On average, the accuracy improves up to 17.02%. Generally, both evaluation metrics demonstrate the effectiveness of TL on offline agent policy learning.

Table 5.5: Accuracy comparison on the UCSF testing subset between actions taken by RL agents ground truth actions implemented by the hospital with and without TL. Experiments are conducted with 5 random initializations. The results are shown in the format of mean and standard deviation. The accuracy improves to varying extents after transfer.

	No Transfer (RNN-R2)	No Transfer (CDE-R2)	BCQ-QVT (RNN-R2)	BCQ-QVT (CDE-R2)	BCQ-WT (RNN-R2)	BCQ-WT (CDE-R2)	BCQ-WTR (RNN-R2)	BCQ-WTR (CDE-R2)
RBC transfusion	0.68 ± 0.02	0.71 ± 0.01	0.78 ± 0.01	**0.81 ± 0.02**	0.74 ± 0.03	0.75 ± 0.02	0.79 ± 0.02	0.80 ± 0.02
PLT transfusion	0.73 ± 0.03	0.73 ± 0.02	0.82 ± 0.02	0.81 ± 0.01	0.78 ± 0.03	0.78 ± 0.02	**0.85 ± 0.02**	**0.85 ± 0.02**
FFP transfusion	0.72 ± 0.03	0.71 ± 0.01	0.79 ± 0.01	**0.82 ± 0.02**	0.78 ± 0.03	0.79 ± 0.02	**0.82 ± 0.01**	**0.82 ± 0.02**

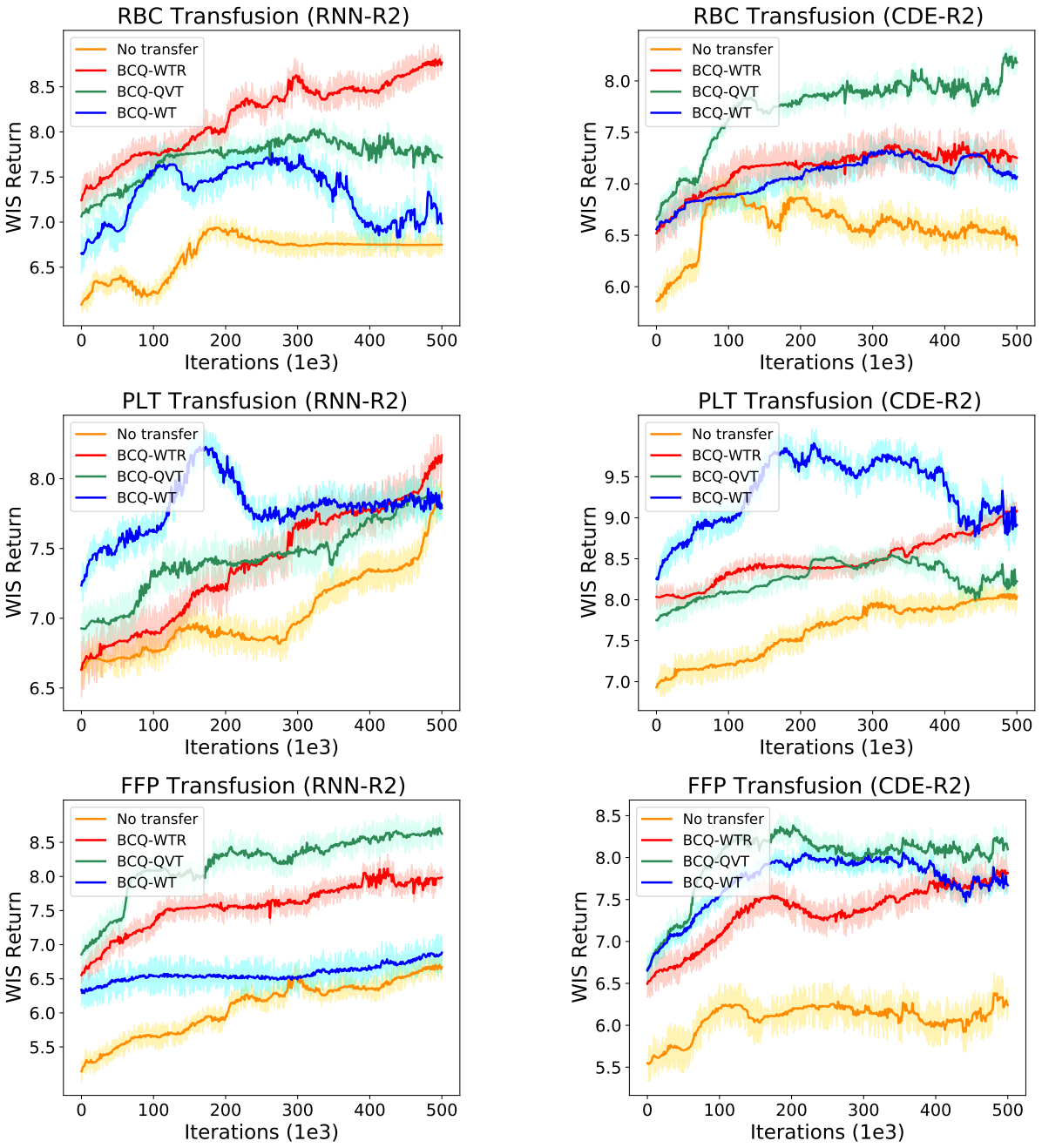


Figure 5.4: WIS evaluation of policies (learning curves) on the UCSF testing subset with RNN-R2 and CDE-R2 RL settings with and without TL on three transfusion tasks. The displayed results are averaged over 5 random seeds. The shaded area measures a single standard deviation across seeds. Among three tasks, all transfer methods show better jump-start and asymptotic performance compared to the original policy learning curves without TL. For RL settings like RNN-R2, transfer can help reduce the oscillations in FFP and PLT transfusion tasks.

5.6.5 Policy Simulation

All the above analysis are based on the assumption that the actual physician decision making regarding blood transfusion can improve patients' clinical outcomes. However, in practice, physicians may not be able to always make optimal transfusion decisions due to insufficient communication with patients and an incomplete understanding of patients' conditions and medical history due to urgency in the ICU. Furthermore, transfusion may not always improve patients' outcomes, resulting in higher risk of mortality and morbidity. Here, we consider patients' short-term (decreased acuity scores) and long-term (increased survival rates) clinical outcomes. In our two real-world datasets, we calculate the Pearson correlation coefficients between a decision of giving transfusion and clinical outcomes on all tasks, as shown in Table 5.6. For the MIMIC-III dataset, transfusion does

Table 5.6: Pearson correlation between transfusion decision and clinical outcomes. LT and ST are short for long-term and short-term, respectively.

Intervention	MIMIC-III LT / ST	UCSF LT / ST
RBC transfusion	0.35 / 0.29	-0.21 / 0.08
PLT transfusion	0.38 / 0.23	-0.25 / -0.12
FFP transfusion	0.32 / 0.31	-0.33 / -0.19

improve patients' both long-term and short-term outcomes. For the UCSF dataset, however, the negative coefficients indicate that not all transfusion decisions improve patients' conditions. Hence, actual physicians' real-time treatment strategies of blood transfusion may require further optimization, probably by assistance from RL agents. Hence, we extract patients from the UCSF dataset that receive transfusions during their hospital stay but decease within 28 days after their admission. Then, we perform policy simulations from the transferred transfusion policies on the UCSF dataset to model the environment of real-time transfusion decision changes and corresponding patients' outcomes. Here, we

use the MIMIC-III dataset to help with the simulation process. We group patients from the MIMIC-III dataset and the selected patients from the UCSF dataset by clustering patients according to their temporal value changes and static demographics. Then, by using patients from the same cluster on the MIMIC-III dataset who survived in 28 days after ICU admission as a control group, we conducted simulations by changing the real-time transfusion policy on the selected UCSF patient cohorts. The estimated mortality rate over all UCSF patients decreases from the actual 16.48% to 13.74%. Similarly, we select patients with worsening conditions during their hospital stay on the UCSF dataset and conduct simulations via transferred policies. The estimated decreased acuity rate over all UCSF patients decreases from the actual 9.13% to 7.95%. Here, all the results are averaged over three transfusion tasks. This is an important finding which supports decision making tools have significant potential to improve patient outcomes.

5.7 Conclusions and Future Work

In this work, we utilized an off-policy batch reinforcement learning algorithm, discretized BCQ, to tackle policy recommendations for blood transfusion in ICUs. We conduct experiments on two real-world datasets with different patient state encoding and reward function mechanisms. Our results demonstrate that using appropriate state representations like RNN and CDE, along with proper reward designs like R2 can provide reasonably well policy training. Furthermore, an integration of TL and RL can help improve the policy learning on a data-scarce dataset to a large extent. As a decision support tool, the learned policy by RL agents may serve as an auxiliary advice for physicians in emergency, and thus potentially assist physicians to optimize the real-time treatment strategies on blood transfusion. Hence, blending the RL with real physicians' decisions using available patient information could lead to better transfusion strategies

and improving patients' clinical outcomes. Possible directions for future work include exploring different patient state representations for better policy learning, extending the action space to include continuous transfusion dosages, using more principled approach to the design of the rewards such as inverse RL, adopting different evaluation methods like doubly robust evaluation, as well as applying the method to other unexplored clinical decision making problems that may fit the RL setting.

Chapter 6

Are Large Language Models Ready for Healthcare? A Comparative Study on Clinical Language Understanding

6.1 Introduction

Recent advancements in clinical language understanding hold the potential to revolutionize healthcare by facilitating the development of intelligent systems that support decision-making [150, 151], expedite diagnostics [152, 3], and improve patient care [153]. Such systems could assist healthcare professionals in managing the ever-growing body of medical literature, interpreting complex patient records, and developing personalized treatment plans [154, 155]. State-of-the-art large language models (LLMs) like OpenAI's GPT-3.5 and GPT-4 [156], and Google AI's Bard [157], have gained significant attention for their remarkable performance across diverse natural language understanding

tasks, such as sentiment analysis, machine translation, text summarization, and question-answering [158, 159, 160]. However, a comprehensive evaluation of their effectiveness in the specialized healthcare domain, with its unique challenges and complexities, remains necessary.

The healthcare domain presents distinct challenges, including handling specialized medical terminology, managing the ambiguity and variability of clinical language, and meeting the high demands for reliability and accuracy in critical tasks. Although existing research has explored the application of LLMs in healthcare, the focus has typically been on a limited set of tasks or learning strategies. For example, studies have investigated tasks like medical concept extraction, patient cohort identification, and drug-drug interaction prediction, primarily relying on supervised learning approaches [161, 162, 163]. In this study, we broaden this scope by evaluating LLMs on various clinical language understanding tasks, including natural language inference (NLI), document classification, semantic textual similarity (STS), question-answering (QA), named entity recognition (NER), and relation extraction.

Furthermore, the exploration of learning strategies such as few-shot learning, transfer learning, and unsupervised learning in the healthcare domain has been relatively limited. Similarly, the impact of diverse prompting techniques on improving model performance in clinical tasks has not been extensively examined, leaving room for a comprehensive comparative study.

In this study, we aim to bridge this gap by evaluating the performance of state-of-the-art LLMs on a range of clinical language understanding tasks. LLMs offer the exciting prospect of in-context few-shot learning via prompting, enabling task completion without fine-tuning separate language model checkpoints for each new challenge. In this context, we propose a novel prompting strategy called self-questioning prompting (SQP) to enhance these models' effectiveness across various tasks. Our empirical evaluations

demonstrate the potential of SQP as a promising technique for improving LLMs in the healthcare domain. Furthermore, by pinpointing tasks where the models excel and those where they struggle, we highlight the need for addressing specific challenges such as wording ambiguity, lack of context, and negation handling, while emphasizing the importance of responsible LLM implementation and collaboration with domain experts in healthcare settings.

In summary, our contributions are threefold:

(1) To the best of our knowledge, this is the first comparative study to investigate the effectiveness of state-of-the-art LLMs on a variety of clinical language understanding tasks with diverse learning strategies and prompting strategies.

(2) We introduce a novel prompting strategy, namely self-questioning prompting, which aims to enhance the performance of LLMs by encouraging the generation of informative questions and answers and prompting a deeper understanding of the medical scenarios being described.

(3) Our error analysis on the most challenging task common to all models highlights the unique challenges each model faces, including wording ambiguity, lack of context, and negation, emphasizing the need for a cautious approach when employing LLMs in healthcare as a supplement to human expertise.

Generalizable Insights about Machine Learning in the Context of Healthcare

Our study presents a comprehensive evaluation of state-of-the-art LLMs in the healthcare domain, examining their capabilities and limitations across a variety of clinical language understanding tasks. We develop and demonstrate the efficacy of our self-

questioning prompting (SQP) strategy, which involves generating context-specific questions and answers to guide the model towards a better understanding of clinical scenarios. This tailored learning approach significantly enhances LLM performance in healthcare-focused tasks. Our in-depth error analysis on the most challenging task shared by all models uncovers unique difficulties encountered by each model, such as wording ambiguity, lack of context, and negation issues. These findings emphasize the need for a cautious approach when implementing LLMs in healthcare as a complement to human expertise. We underscore the importance of integrating domain-specific knowledge, fostering collaborations among researchers, practitioners, and domain experts, and employing task-oriented prompting techniques like SQP. By addressing these challenges and harnessing the potential benefits of LLMs, we can contribute to improved patient care and clinical decision-making in healthcare settings.

6.2 Related Work

In this section, we review the relevant literature on large language models applied to clinical language understanding tasks in healthcare, as well as existing prompting strategies.

6.2.1 Large Language Models in Healthcare

The advent of the Transformer architecture [64] revolutionized the field of natural language processing, paving the way for the development of large-scale pre-trained language models such as base BERT [50] and RoBERTa [164]. In the healthcare domain, domain-specific adaptations of BERT, such as BioBERT [53] and ClinicalBERT [54], have been introduced to tackle various clinical language understanding tasks. More recently, GPT-3.5 and its successor GPT-4, launched by OpenAI [156], as well as Bard, devel-

oped by Google AI [157], have emerged as state-of-the-art LLMs, showcasing impressive capabilities in a wide range of applications, including healthcare [165, 166, 167, 168].

Clinical language understanding is a critical aspect of healthcare informatics, focused on extracting meaningful information from diverse sources, such as electronic health records [169], scientific articles [170], and patient-authored text data [171]. This domain encompasses various tasks, including NER [172], relation extraction [173], NLI [174], STS [175], document classification [176], and QA [177]. Prior work has demonstrated the effectiveness of domain-specific models in achieving improved performance on these tasks compared to general-purpose counterparts [178, 179, 180]. However, challenges posed by complex medical terminologies, the need for precise inference, and the reliance on domain-specific knowledge can limit their effectiveness [181]. In this work, we address some of these limitations by conducting a comprehensive evaluation of state-of-the-art LLMs on a diverse set of clinical language understanding tasks, focusing on their performance and applicability within healthcare settings.

6.2.2 Prompting Strategies

Prompting strategies, often used in conjunction with few-shot or zero-shot learning [182, 183], guide and refine the behavior of LLMs to improve performance on various tasks. In these learning paradigms, LLMs are conditioned on a limited number of examples in the form of prompts, enabling them to generalize and perform well on the target task. Standard prompting techniques [182] involve providing an LLM with a clear and concise prompt, often in the form of a question or statement, which directs the model towards the desired output. Another approach, known as chain-of-thought prompting [184, 183], leverages a series of interconnected prompts to generate complex reasoning or multi-step outputs. While these existing prompting strategies have shown

considerable success, their effectiveness can be limited by the quality and informativeness of the prompts [185], which may not always capture the intricate nuances of specialized domains like healthcare. Motivated by these limitations, we propose a novel prompting strategy called self-questioning prompting (SQP). SQP aims to enhance the performance of LLMs by generating informative questions and answers related to the given clinical scenarios, thus addressing the unique challenges of the healthcare domain and contributing to improved task-specific performance.

6.3 Self-Questioning Prompting

Complex problems can be daunting, but they can often be solved by breaking them down into smaller parts and asking questions to clarify understanding and explore different aspects. Inspired by this human-like reasoning process, we introduce a novel method called self-questioning prompting (SQP) for LLMs. SQP aims to enhance model performance by encouraging models to be more aware of their own thinking processes, enabling them to better understand relevant concepts and develop deeper comprehension. This is achieved through the generation of targeted questions and answers that provide additional context and clarification, ultimately leading to improved performance on various tasks. The general construction process of SQP for a task, as shown in Figure 6.1, involves identifying key information in the input text, generating targeted questions to clarify understanding, using the questions and answers to enrich the context of the task prompt, and tailoring the strategy to meet the unique output requirements of each task.

In Table 6.1, we compare the proposed SQP with existing prompting methods, such as standard prompting and chain-of-thought prompting, highlighting the differences in guidelines and purposes for each strategy. Subsequently, we present the SQP templates for six clinical language understanding tasks, with the core self-questioning process high-

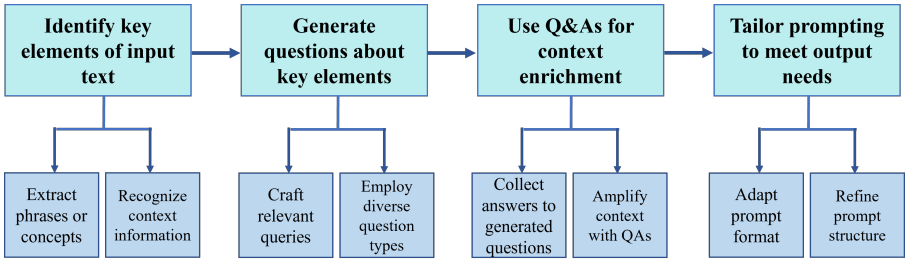


Figure 6.1: Construction process of self-questioning prompting (SQP).

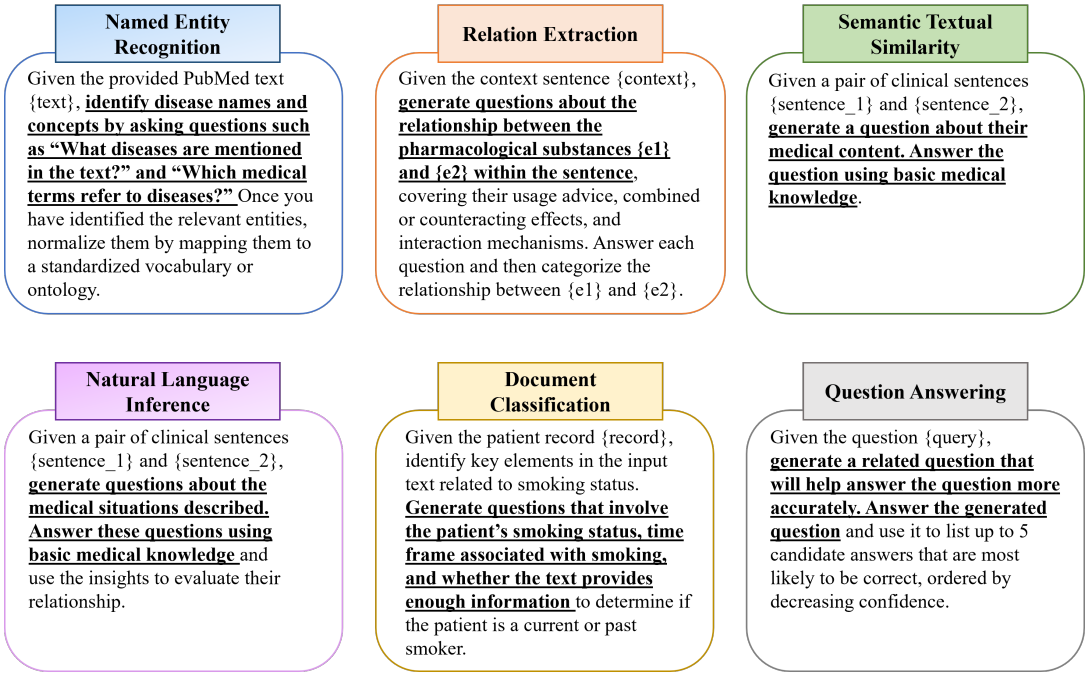


Figure 6.2: Self-questioning prompting (SQP) templates for six clinical language understanding tasks, with the core self-questioning process underscored and bolded. These components represent the generation of targeted questions and answers, guiding the model's reasoning and enhancing task performance.

lighted in each template, as shown in Figure 6.2. These underscored and bold parts illustrate how SQP generates targeted questions and answers related to the tasks, which guide the model's reasoning, leading to improved task performance. By incorporating this self-questioning process into the prompts, SQP enables the model to utilize its knowledge more effectively and adapt to a wide range of clinical tasks.

Table 6.1: Comparison among standard prompting, chain-of-thought prompting, and self-questioning prompting.

Prompting Strategy	**Guideline**	**Purpose**
Standard	Use a direct, concise prompt for the desired task.	To obtain a direct response from the model.
Chain-of-Thought	Create interconnected prompts guiding the model through logical reasoning.	To engage the model's reasoning by breaking down complex tasks.
Self-Questioning	Generate targeted questions and use answers to guide the task response.	To deepen the model's understanding and enhance performance.

6.4 Datasets

We utilize a wide range of biomedical and clinical language understanding datasets for our experiments. These datasets encompass various tasks, including NER (NCBI-Disease [186] and BC5CDR-Chem [187]), relation extraction (i2b2 2010-Relation [188] and SemEval 2013-DDI [189]), STS (BIOSSES [190]), NLI (MedNLI [174]), document classification (i2b2 2006-Smoking [191]), and QA (bioASQ 10b-Factoid [192]). Among these tasks, STS (BIOSSES) is a regression task, while the rest are classification tasks. Table 6.2 offers a comprehensive overview of the tasks and datasets. For NER tasks, we adopt the BIO tagging scheme, where 'B' represents the beginning of an entity, 'I' signifies the continuation of an entity, and 'O' denotes the absence of an entity. The

output column in Table 6.2 presents specific classes, scores, or tagging schemes associated with each task.

For relation extraction, SemEval 2013-DDI requires identifying one of the following labels: Advice, Effect, Mechanism, or Int. In the case of i2b2 2010-Relation, it necessitates predicting relationships such as Treatment Improves Medical Problem (TrIP), Treatment Worsens Medical Problem (TrWP), Treatment Causes Medical Problem (TrCP), Treatment is Administered for Medical Problem (TrAP), Treatment is Not Administered because of Medical Problem (TrNAP), Test Reveals Medical Problem (TeRP), Test Conducted to Investigate Medical Problem (TeCP), or Medical Problem Indicates Medical Problem (PIP).

Table 6.2: Overview of biomedical/clinical language understanding tasks and datasets.

Task	**Dataset**	**Output**	**Metric**
Named Entity Recognition	NCBI-Disease, BC5CDR-Chemical	BIO tagging for diseases and chemicals	Micro F1
Relation Extraction	i2b2 2010-Relation, SemEval 2013-DDI	relations between entities	Micro F1, Macro F1
Semantic Textual Similarity	BIOSSES	similarity scores from 0 (different) to 4 (identical)	Pearson Correlation
Natural Language Inference	MedNLI	entailment, neutral, contradiction	Accuracy
Document Classification	i2b2 2006-Smoking	current smoker, past smoker, smoker, non-smoker, unknown	Micro F1
Question-Answering	bioASQ 10b-Factoid	factoid answers	Mean Reciprocal Rank, Lenient Accuracy

6.5 Experiments

In this section, we outline the experimental setup and evaluation procedure used to evaluate the performance of various LLMs on tasks related to biomedical and clinical text comprehension and analysis.

6.5.1 Experimental Setup

We investigate various prompting strategies for state-of-the-art LLMs, employing N-shot learning techniques on diverse clinical language understanding tasks.

Large Language Models. We assess the performance of three state-of-the-art LLMs, each offering unique capabilities and strengths. First, we examine GPT-3.5, an advanced model developed by OpenAI, known for its remarkable language understanding and generation capabilities. Next, we investigate GPT-4, an even more powerful successor to GPT-3.5, designed to push the boundaries of natural language processing further. Finally, we explore Bard, an innovative language model launched by Google AI. By comparing these models, we aim to gain insights into their performance on clinical language understanding tasks.

Prompting Strategies. We employ three prompting strategies to optimize the performance of LLMs on each task: standard prompting, chain-of-thought prompting, and our proposed self-questioning prompting. Standard prompting serves as the baseline, while chain-of-thought and self-questioning prompting techniques are investigated to assess their potential impact on model performance.

N-Shot Learning. We explore N-shot learning for LLMs, focusing on zero-shot and 5-shot learning scenarios. Zero-shot learning refers to the situation where the model has not been exposed to any labeled examples during training and is expected to generalize to the task without prior knowledge. In contrast, 5-shot learning involves the model receiving a small amount of labeled data, consisting of five few-shot exemplars from the training set, to facilitate its adaptation to the task. We evaluate the model's performance in both zero-shot and 5-shot learning settings to understand its ability to generalize and adapt to different tasks in biomedical and clinical domains.

6.5.2 Evaluation Procedure

To assess the performance for each task, we first create an evaluation set by randomly selecting 50 instances from the test set. In the case of zero-shot learning, we directly evaluate the model's performance on this evaluation set. For 5-shot learning, we enhance the model with five few-shot exemplars, which are randomly chosen from the training set. The model's performance is then assessed using the same evaluation set as in the zero-shot learning scenario.

6.6 Results

In this section, we present a comprehensive analysis of the performance of the LLMs (i.e., Bard, GPT-3.5, and GPT-4) on clinical language understanding tasks. We begin by comparing the overall performance of these models, followed by an examination of the effectiveness of various prompting strategies. Next, we delve into a detailed task-by-task analysis, providing insights into the models' strengths and weaknesses across different tasks. Finally, we conduct a case study on error analysis, investigating common error types and the potential improvements brought about by advanced prompting techniques.

6.6.1 Overall Performance Comparison

In our study, we evaluate the performance of Bard, GPT-3.5, and GPT-4 on various clinical benchmark datasets spanning multiple tasks. We employ different prompting strategies, including standard, chain-of-thought, and self-questioning, as well as N-shot learning with N equal to 0 and 5. Table 6.3 summarizes the experimental results.

We observe that GPT-4 generally outperforms Bard and GPT-3.5 in tasks involving the identification and classification of specific information within text, such as NLI

Table 6.3: Performance comparison of Bard, GPT-3.5, and GPT-4 with different prompting strategies (standard, chain-of-thought, and self-questioning) and N-shot learning (N = 0, 5) on clinical benchmark datasets. randomly sampled evaluation data from the test set. Our results show that GPT-4 outperforms Bard and GPT-3.5 in tasks that involve identification and classification of specific information within text, while Bard achieves higher accuracy than GPT-3.5 and GPT-4 on tasks that require a more factual understanding of the text. Additionally, self-questioning prompting consistently achieves the best performance on the majority of tasks. The best results for each dataset are highlighted in bold.

Model	**NCBI-Disease** *Micro F1*	**BC5CDR-Chemical** *Micro F1*	**i2b2 2010-Relation** *Micro F1*	**SemEval 2013-DDI** *Macro F1*	**BIOSSES** *Pear.*	**MedNLI** *Acc.*	**i2b2 2006-Smoking** *Micro F1*	**BioASQ 10b-Factoid** *MRR*	*Len. Acc.*
Bard									
w/ zero-shot StP	0.911	0.947	0.720	0.490	0.401	0.580	0.780	0.800	0.820
w/ 5-shot StP	0.933	0.972	0.900	0.528	0.449	0.640	0.820	0.845	0.880
w/ zero-shot CoTP	0.946	0.972	0.660	0.525	0.565	0.580	0.760	**0.887**	**0.920**
w/ 5-shot CoTP	0.955	0.977	0.900	0.709	0.602	0.720	0.800	0.880	0.900
w/ zero-shot SQP	0.956	0.977	0.760	0.566	0.576	0.760	0.760	0.850	0.860
w/ 5-shot SQP	0.960	0.983	**0.940**	0.772	0.601	0.760	0.820	0.860	0.860
GPT-3.5									
w/ zero-shot StP	0.918	0.939	0.780	0.360	0.805	0.700	0.680	0.707	0.720
w/ 5-shot StP	0.947	0.967	0.840	0.531	0.828	0.780	0.780	0.710	0.740
w/ zero-shot CoTP	0.955	0.977	0.680	0.404	0.875	0.740	0.680	0.743	0.800
w/ 5-shot CoTP	0.967	0.977	0.840	0.548	0.873	0.740	0.740	0.761	0.820
w/ zero-shot SQP	0.963	0.974	0.860	0.529	0.873	0.760	0.720	0.720	0.740
w/ 5-shot SQP	0.970	0.983	0.860	0.620	0.892	0.820	0.820	0.747	0.780
GPT-4									
w/ zero-shot StP	0.968	0.976	0.860	0.428	0.820	0.800	**0.900**	0.795	0.820
w/ 5-shot StP	0.975	0.989	0.860	0.502	0.848	0.840	0.880	0.815	0.840
w/ zero-shot CoTP	0.981	0.994	0.860	0.509	0.875	0.840	0.860	0.805	0.840
w/ 5-shot CoTP	0.984	0.994	0.880	0.544	0.897	0.800	0.860	0.852	0.880
w/ zero-shot SQP	**0.985**	0.992	0.920	0.595	0.889	**0.860**	**0.900**	0.844	0.900
w/ 5-shot SQP	0.984	**0.995**	0.920	**0.798**	**0.916**	**0.860**	0.860	0.873	0.900

Note: Acc. = Accuracy; CoTP = Chain-of-Thought Prompting; Len. Acc. = Lenient Accuracy; MRR = Mean Reciprocal Rank; Pear. = Pearson Correlation; StP = Standard Prompting.

(MedNLI), NER (NCBI-Disease, BC5CDR-Chemical), and STS (BIOSSES). In the realm of document classification, a task that involves assigning predefined categories to entire documents, GPT-4 also surpasses GPT-3.5 and Bard on the i2b2 2006-Smoking dataset. In relation extraction, GPT-4 outperforms both Bard and GPT-3.5 on the SemEval 2013-DDI dataset, while Bard demonstrates superior performance in the i2b2 2010-Relation dataset. Additionally, Bard excels in tasks that require a more factual understanding of the text, such as QA (BioASQ 10b-Factoid).

Regarding prompting strategies, self-questioning consistently outperforms standard prompting and exhibits competitive performance when compared to chain-of-thought prompting across all settings. Our findings suggest that self-questioning is a promising approach for enhancing the performance of LLMs, achieving the best performance for the majority of tasks, except for QA (BioASQ 10b-Factoid).

Furthermore, our study demonstrates that 5-shot learning generally leads to improved performance across all tasks when compared to zero-shot learning, although not universally. This finding indicates that incorporating even a modest amount of task-specific training data can substantially enhance the effectiveness of pre-trained LLMs.

6.6.2 Prompting Strategies Comparison

We evaluate the performance of different prompting strategies, specifically standard prompting, self-questioning prompting (SQP), and chain-of-thought prompting (CoTP) on both zero-shot and 5-shot learning settings across various models and datasets. Figure 6.3 presents the averaged performance comparison over all datasets, under the assumption that datasets and evaluation metrics are equally important and directly comparable. We observe that self-questioning prompting consistently yields the best performance compared to standard and chain-of-thought prompting. In addition, GPT-4

excels among the models, demonstrating the highest overall performance.

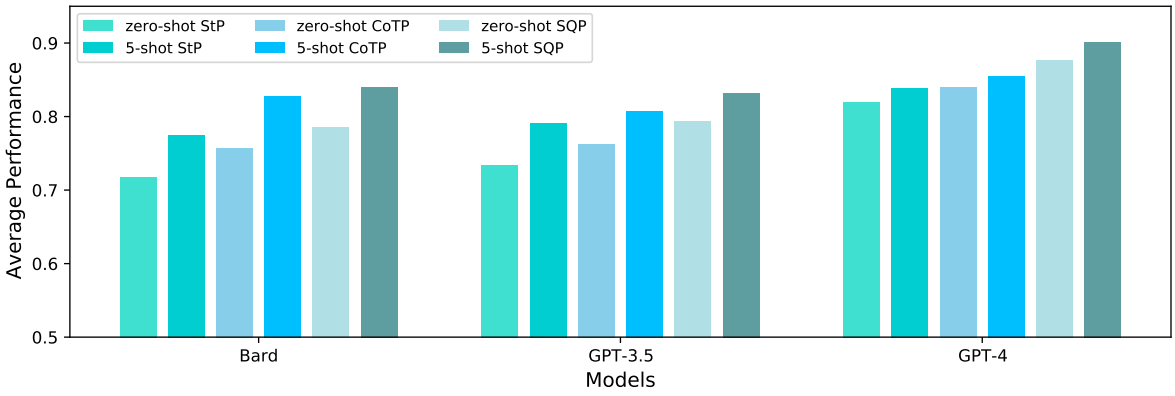


Figure 6.3: Average performance comparison of three prompting methods in zero-shot and 5-shot learning settings across Bard, GPT-3.5, and GPT-4 models. Performance values are averaged across all datasets, assuming equal importance for datasets and evaluation metrics, as well as direct comparability. The self-questioning prompting method consistently outperforms standard and chain-of-thought prompting, and GPT-4 excels among the models.

Table 6.4 and Table 6.5 demonstrate performance improvements of prompting strategies over multiple datasets and models under zero-shot and 5-shot settings, respectively, using standard prompting as a baseline. In the zero-shot learning setting (Table 6.4), self-questioning prompting achieves the highest improvement in the majority of tasks, with improvements ranging from 4.9% to 46.9% across different datasets.

In the 5-shot learning setting (Table 6.5), self-questioning prompting leads to the highest improvement in most tasks, with improvements ranging from 2.9% to 59.0%. In both settings, we also observe some instances where chain-of-thought or self-questioning prompting yields negative values, such as relation extraction (i2b2 2010-Relation) and document classification (i2b2 2006-Smoking), indicating inferior performance compared to standard prompting. This could be due to the specific nature of certain tasks, where the additional context or complexity introduced by the alternative prompting strategies might not contribute to better understanding or performance. It might also be possible

that the model's capacity is insufficient to take advantage of the additional information provided by the alternative prompting strategies in some cases.

Overall, self-questioning prompting generally outperforms other prompting strategies across different models and datasets in both zero-shot and 5-shot learning settings, despite occasional inferior performance in specific tasks. This suggests that self-questioning prompting can be a promising technique for improving performance in the domain of clinical language understanding. Furthermore, GPT-4 emerges as the top-performing model, emphasizing the potential for various applications in the clinical domain.

Table 6.4: Comparison of zero-shot learning performance improvements (in %) for different models and prompting techniques on multiple datasets, with standard prompting as the baseline. Bold values indicate the highest improvement for each dataset across models and prompting strategies, while negative values signify inferior performance. Self-questioning prompting leads to the largest improvement in the majority of tasks.

Dataset	**Metric**	**Bard**		**GPT-3.5**		**GPT-4**	
		CoTP	SQP	CoTP	SQP	CoTP	SQP
NCBI-Disease	*Micro F1*	3.8	**4.9**	4.0	**4.9**	1.3	1.8
BC5CDR-Chemical	*Micro F1*	2.6	3.2	**4.0**	3.7	1.8	1.6
i2b2 2010-Relation	*Micro F1*	−8.3	5.6	−12.8	**10.3**	0.0	7.0
SemEval 2013-DDI	*Macro F1*	7.1	15.5	12.2	**46.9**	18.9	39.0
BIOSSES	*Pear.*	40.9	**43.6**	8.7	8.4	6.7	8.4
MedNLI	*Acc.*	0.0	**31.0**	5.7	8.6	5.0	7.5
i2b2 2006-Smoking	Micro F1	−2.6	−2.6	0.0	**5.9**	−4.4	0.0
BioASQ 10b-Factoid	*MRR*	**10.9**	6.3	5.1	1.8	1.3	6.2
BioASQ 10b-Factoid	*Len. Acc.*	**12.2**	4.9	11.1	2.8	2.4	9.8

6.6.3 Task-by-Task Analysis

To delve deeper into the specific characteristics and challenges associated with each task (i.e., NER, relation extraction, STS, NLI, document classification, and QA), we

Table 6.5: Comparison of 5-shot learning performance improvements (in %) for different models and prompting techniques on multiple datasets, with standard prompting as the baseline. Bold values indicate the highest improvement for each dataset across models and prompting strategies, while negative values signify inferior performance. Self-questioning prompting leads to the highest improvement in 6 out of 8 tasks, followed by chain-of-thought prompting with 2 largest improvements.

Dataset	**Metric**	**Bard**		**GPT-3.5**		**GPT-4**	
		CoTP	SQP	CoTP	SQP	CoTP	SQP
NCBI-Disease	*Micro F1*	2.4	**2.9**	2.1	2.4	0.9	0.9
BC5CDR-Chemical	*Micro F1*	0.5	1.1	1.0	**1.7**	0.5	0.6
i2b2 2010-Relation	*Micro F1*	0.0	4.4	0.0	2.4	2.3	**7.0**
SemEval 2013-DDI	*Macro F1*	34.3	46.2	3.2	16.8	8.4	**59.0**
BIOSSES	*Pear.*	**34.1**	33.9	5.4	7.7	5.8	8.0
MedNLI	*Acc.*	12.5	**18.8**	−5.1	5.1	−4.8	2.4
i2b2 2006-Smoking	Micro F1	−2.4	0.0	−5.1	**5.1**	−2.3	−2.3
BioASQ 10b-Factoid	*MRR*	4.1	1.8	**7.2**	5.2	4.5	7.1
BioASQ 10b-Factoid	*Len. Acc.*	2.3	−2.3	**10.8**	5.4	4.8	7.1

individually analyze the results, aiming to better understand the underlying factors that contribute to model performance and identify areas for potential improvement or further investigation.

Named Entity Recognition Task. In the NER task, we focus on two datasets: NCBI-Disease and BC5CDR-Chemical. Employing the BIO tagging scheme, we evaluate model performance using the micro F1 metric. NER tasks in the biomedical domain pose unique challenges due to specialized terminology, complex entity names, and frequent use of abbreviations. Our results indicate that, compared to standard prompting, self-questioning prompting leads to average improvements of 3.9% and 2.8% in zero-shot learning for NCBI-Disease and BC5CDR-Chemical, respectively. In the 5-shot setting, the average improvements are 2.1% and 1.1%, respectively. Moreover, GPT-4 demonstrates the most significant performance boost compared to Bard and GPT-3.5.

We also conduct a qualitative analysis by examining specific examples from the datasets, such as the term "aromatic ring" in the BC5CDR-Chemical dataset, which is often incorrectly predicted as "B-Chemical" (beginning of a chemical entity) instead of "O" (outside of any entity) by the models. This error might occur because the term "aromatic ring" refers to a structural feature commonly found in chemical compounds, leading models to associate it with chemical entities and misclassify it. This example highlights the challenges faced by the models in accurately recognizing entities, particularly when dealing with terms that have strong associations with specific entity types. It also demonstrates the potential limitations of prompting strategies in addressing these challenges, as models may still struggle to disambiguate such terms, despite employing different prompting techniques.

Relation Extraction Task. In the relation extraction task involving the i2b2 2010-Relation and SemEval 2013-DDI datasets, we evaluate our model's performance using micro F1 and macro F1 scores, respectively. Our study reveals that self-questioning prompting leads to average improvements of 7.6% and 33.8% in zero-shot learning for the i2b2 2010-Relation and SemEval 2013-DDI datasets, respectively. In the 5-shot setting, the average improvements are 4.6% and 40.7%, respectively. GPT-4 demonstrates more significant performance improvement compared to Bard and GPT-3.5.

For our qualitative analysis, we examine a challenging example from the i2b2 2010-Relation dataset, where the models struggle to identify the correct relationship between "Elavil" and "stabbing left-sided chest pain". The gold label indicates "TrWP" (Treatment Worsens Medical Problem), but all models incorrectly predict it as "TrAP" (Treatment is Administered for Medical Problem). This misclassification may arise from the models' inability to recognize that the patient still experiences severe pain despite taking Elavil. This example highlights the difficulties encountered by the models in accurately identifying nuanced relationships in complex biomedical texts. Incorporating domain-

specific knowledge could help to better capture the subtleties of such relationships.

Semantic Textual Similarity Task. In the STS task, we focus on the BIOSSES dataset and evaluate our model's performance using Pearson correlation. Our study reveals that self-questioning prompting leads to average improvements of 20.1% and 16.5% in zero-shot and 5-shot settings, respectively. GPT-4 outperforms Bard and GPT-3.5 across all settings.

Taking a closer look, we examine a pair of sentences with a gold label similarity score of 0.2, indicating high dissimilarity. The first sentence discusses the specific effect of mutant K-Ras on tumor progression, while the second sentence refers to an important advance in lung cancer research without mentioning any specific details. However, the average score predicted by models, regardless of the setting, is 2.0. This discrepancy may arise from the models' difficulty in grasping the distinct contexts in which the sentences are written. The models might be misled by the presence of related keywords such as "tumor" and "cancer", leading to an overestimation of the similarity score. This example demonstrates the challenge faced by the models in accurately gauging the semantic similarity of sentences when the underlying context or focus differs, despite the presence of shared terminology.

Natural Language Inference Task. In the NLI task, we focus on the MedNLI dataset and evaluate our model's performance using accuracy. On average, self-questioning prompting improves the model performance by 15.7% and 8.8% for zero-shot and 5-shot settings, respectively, with GPT-4 consistently outperforming Bard and GPT-3.5 across all settings.

We further investigate a pair of sentences where the gold label is "contradiction". The first sentence states that the patient was transferred to the Neonatal Intensive Care Unit for observation, while the second sentence claims that the patient had an uneventful course. Despite the gold label, none of the models ever predict the true label, opting

for "neutral" or "entailment" instead. The models may focus on the absence of explicit negations or conflicting keywords, leading them to overlook the more subtle contradiction. These findings highlight the need to enhance model capabilities to better understand implicit and nuanced relationships between sentences, thereby enabling more accurate predictions in complex real-world clinical scenarios.

Document Classification Task. In the document classification task, we focus on the i2b2 2006-Smoking dataset and evaluate our model's performance using micro F1. Our analysis reveals that self-questioning prompting leads to average improvements of 1.1% and 0.9% for zero-shot and 5-shot settings, respectively. GPT-4 consistently delivers superior performance to Bard and GPT-3.5 in all experimental settings.

During our qualitative assessment, we investigate a patient record containing the sentence "He is a heavy smoker and drinks 2-3 shots per day at times". All models classify the patient as a "CURRENT SMOKER", while the patient is, in fact, a past smoker, as indicated by the subsequent descriptions of medications and the patient's improved condition. This misclassification may occur because the models focus on the explicit mention of smoking habits in the sentence, neglecting the broader context provided by the entire document. This instance highlights the need for models to take a more comprehensive approach in interpreting clinical documents by considering the overall context, rather than relying solely on individual textual cues.

Question-Answering Task. In the QA task using the bioASQ 10b-Factoid dataset, we evaluate our model with MRR and lenient accuracy. For MRR, self-questioning prompting leads to average improvements of 4.8% and 4.7% for zero-shot and 5-shot settings, respectively. For lenient accuracy, the improvements are 5.8% and 3.4%, respectively. GPT-4 consistently outperforms Bard and GPT-3.5 across all settings.

During our qualitative exploration, we analyzed an example question: "What is the major sequence determinant for nucleosome positioning?" The correct answer is "G+C

content"; however, the top answer from models is "DNA sequence". This misclassification might occur because the models capture the broader context related to nucleosome positioning but fail to recognize the specific determinant, namely G+C content. The models may rely on more general associations between DNA sequences and nucleosome positioning, resulting in a less precise answer. This example underscores the necessity for models to identify fine-grained details in biomedical questions and deliver more accurate and specific responses.

6.6.4 Case Study: Error Analysis

We conduct a comprehensive error analysis on relation extraction (SemEval 2013-DDI), the most challenging task shared by all LLMs. This task is identified by calculating the median performance across all settings for a robust representation. We investigate common error types and provide illustrative examples, examining the influence of prompting strategies and N-shot learning on the models' performance. This analysis highlights each model's strengths, limitations, and the role of experimental settings in improving clinical language understanding tasks.

Table 6.6: Average error type distribution for SemEval 2013-DDI across Bard, GPT-3.5, and GPT-4. Wording Ambiguity is the most common error for Bard, Lack of Context for GPT-3.5, and Negation and Qualification for GPT-4.

Error Type	**Description**	**Error Proportion (%)**		
		Bard	GPT-3.5	GPT-4
Wording Ambiguity	unclear wording	**32**	23	24
Lack of Context	incomplete context usage	25	**31**	19
Complex Interactions	multiple drug interactions	19	12	14
Negation and Qualification	Misinterpreting negation/qualification	8	27	**25**
Co-reference Resolution	Misidentifying co-references	16	7	18

Table 6.6 presents the average error type distribution for the SemEval 2013-DDI task across Bard, GPT-3.5, and GPT-4. The average proportions are calculated by aggregating error frequencies for each error type across all settings and then dividing by the total number of errors for each model. The most common error type for Bard is Wording Ambiguity, accounting for 32% of its errors, which may stem from the inherent complexity of clinical language or insufficient training data for specific drug relations. In contrast, GPT-3.5 struggles the most with Lack of Context, comprising 31% of its errors, suggesting the model's difficulty in grasping the broader context of the input text. GPT-4's top error is Negation and Qualification, making up 25% of its errors, possibly due to the model's limitations in understanding and processing negations and qualifications within the clinical domain. This analysis highlights the unique challenges each model faces in the relation extraction task, emphasizing the need for targeted interventions and tailored strategies to address these specific areas for improvement.

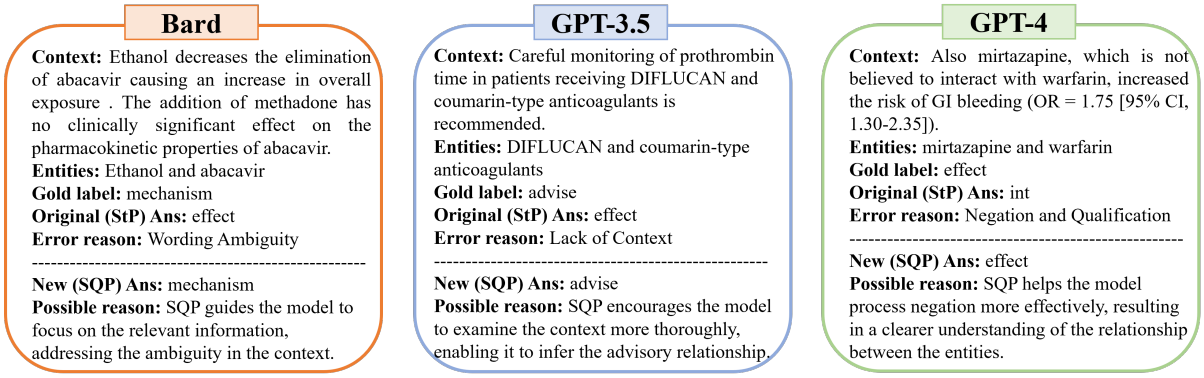


Figure 6.4: Error correction examples using self-questioning prompting (SQP) for Bard, GPT-3.5, and GPT-4 in the SemEval 2013-DDI dataset, compared to standard prompting (StP). Each example showcases the top error for each model and how SQP addresses these challenges. As this chapter primarily focuses on the effectiveness of SQP, chain-of-thought prompting is not presented in these examples.

Specific examples presented in Figure 6.4 illustrate the challenges faced by each model and how self-questioning prompting (SQP) can effectively improve their performance.

SQP demonstrates its flexibility and adaptability across various model architectures by mitigating distinct error types and refining predictions. Bard sees improvements in addressing Wording Ambiguity, GPT-3.5 benefits from enhanced context utilization, and GPT-4's understanding of negation is strengthened. These examples emphasize the significance of harnessing advanced prompting techniques like SQP to bolster model performance and reveal the multifaceted challenges faced by LLMs in relation extraction tasks, particularly within the clinical domain.

Our findings highlight the potential of advanced prompting techniques, such as self-questioning prompting, in addressing model-specific errors and enhancing overall performance. These insights can be extended to various clinical language understanding tasks, guiding future research to develop more robust, accurate, and reliable models capable of processing complex clinical information and improving patient care.

6.7 Discussion

In this study, we have conducted a comprehensive evaluation of state-of-the-art large language models in the healthcare domain, including GPT-3.5, GPT-4, and Bard. We have examined the capabilities and limitations of these leading large language models across various clinical language understanding tasks such as NER, relation extraction, and QA. Our findings suggest that while LLMs have made substantial progress in understanding clinical language and achieving competitive performance across these tasks, they still exhibit notable limitations and challenges. Some of these challenges include the varying confidence levels of their responses and the difficulty in determining the trustworthiness of their generated information without human validation. Consequently, our study emphasizes the importance of using LLMs with caution as a supplement to existing workflows rather than as a replacement for human expertise. To effectively implement

LLMs, clinical practitioners should employ task-specific learning strategies and prompting techniques, such as SQP, carefully designing and selecting prompts that guide the model towards better understanding and generation of relevant responses. Collaboration with experts during the development and fine-tuning of LLMs is essential to ensure accurate capture of domain-specific knowledge and sensitivity to clinical language nuances. Additionally, clinicians should be aware of the limitations and potential biases in LLMs and ensure that a human expert verifies the information they produce. By adopting a cautious approach, healthcare professionals can harness the potential of LLMs responsibly and effectively, ultimately contributing to improved patient care.

Limitations While this study presents meaningful observations and sheds light on the role of large language models in the healthcare domain, there are some limitations to our work. Our study focuses on a select group of state-of-the-art LLMs, which may limit the generalizability of our findings to other models or future iterations. The performance of the proposed SQP strategy may vary depending on the tasks, prompting setup, and input-output exemplars used, suggesting that further research into alternative prompting strategies or other techniques is warranted. Our evaluation is based on a set of clinical language understanding tasks and may not cover all possible use cases in the healthcare domain, necessitating further investigation into other tasks or subdomains. Lastly, ethical and legal considerations, such as patient privacy, data security, and potential biases, are not explicitly addressed in this study. Future work should explore these aspects to ensure the responsible and effective application of LLMs in healthcare settings.

Chapter 7

Conclusion and Future Work

This dissertation introduces the strides we have made in applying machine learning (ML) to healthcare. Our work sheds light on critical challenges, presents innovative solutions, and offers key insights, all contributing to the potential enhancement of patient outcomes and the optimization of healthcare delivery. Specifically, the empirical analysis on the prediction of multiple organ failure in trauma patients sheds light on the impact of different ML configurations. The findings highlight the importance of classifier choice in achieving better performance and reveal the trade-off between model complexity and performance variation.

The proposed multimodal Transformer model for early sepsis prediction demonstrates its effectiveness over competitive baselines. By integrating physiological time series data and clinical notes, the model shows promise in identifying sepsis cases at an early stage, enabling timely interventions and potentially reducing mortality rates.

The development of an efficient model for multivariate time series classification addresses the computational challenges associated with large-scale time series datasets. The proposed methodology, based on module-wise pruning and Pareto analysis, successfully identifies the relationship between model efficiency and accuracy, providing insights for

designing computationally efficient ML models without compromising performance.

The application of reinforcement learning in predicting the need for blood transfusion in intensive care units provides a valuable decision support tool. The proposed off-policy batch reinforcement learning algorithm demonstrates its potential in making transfusion recommendations based on observed patient trajectories, contributing to more effective treatment strategies and optimizing resource allocation.

The comparative study on the readiness of large language models (LLMs) for healthcare highlights their potential for clinical language understanding tasks. The evaluation of state-of-the-art LLMs and the introduction of a novel prompting strategy shed light on the importance of task-specific learning strategies to maximize their effectiveness in healthcare applications.

While this dissertation addresses some key challenges in the field of ML applications in healthcare, there remain important avenues for future research to explore. Some potential areas of future work include:

(1) **Multimodal Learning**: Further investigation into the integration of multiple data modalities, such as text, images, and physiological signals, can enhance the performance and interpretability of ML models in healthcare. This involves developing robust algorithms and architectures that can effectively handle the challenges associated with heterogeneous data.

(2) **Ethical and Responsible Use of ML**: Continued efforts are needed to address ethical considerations, including privacy, security, fairness, and bias mitigation, when deploying ML models in healthcare. Future research should focus on developing methods and techniques to ensure transparency, interpretability, and accountability of ML models, enabling healthcare professionals to trust and understand the decision-making process.

(3) **Scalability and Generalizability**: Validating and evaluating ML models on diverse patient populations and healthcare settings is crucial to ensure their effectiveness and reliability. Collaborations with healthcare institutions and the collection of large-scale, diverse datasets will enable the scalability and generalizability of ML models.

(4) **Real-Time Decision Support**: Further development of real-time ML-based decision support tools can assist healthcare professionals in making timely and accurate decisions. This involves integrating ML models into clinical workflows and providing real-time insights and recommendations for personalized patient care.

Bibliography

[1] Y. Zhao, *Data Mining in Neuroscience and Healthcare*. University of California, Santa Barbara, 2021.

[2] Y. Wang, Y. Zhao, R. Callcut, and L. Petzold, *Empirical analysis of machine learning configurations for prediction of multiple organ failure in trauma patients*, *arXiv preprint arXiv:2103.10929* (2021).

[3] Y. Wang, Y. Zhao, R. Callcut, and L. Petzold, *Integrating physiological time series and clinical notes with transformer for early prediction of sepsis*, *arXiv preprint arXiv:2203.14469* (2022).

[4] Y. Wang, Y. Zhao, and L. Petzold, *Enhancing transformer efficiency for multivariate time series classification*, *arXiv preprint arXiv:2203.14472* (2022).

[5] Y. Wang, Y. Zhao, and L. Petzold, *Predicting the need for blood transfusion in intensive care units with reinforcement learning*, in *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–10, 2022.

[6] Y. Wang, Y. Zhao, and L. Petzold, *Are large language models ready for healthcare? a comparative study on clinical language understanding*, *arXiv preprint arXiv:2304.05368* (2023).

[7] V.-P. Harjola, W. Mullens, M. Banaszewski, J. Bauersachs, H.-P. Brunner-La Rocca, O. Chioncel, S. P. Collins, W. Doehner, G. S. Filippatos, A. J. Flammer, *et. al.*, *Organ dysfunction, injury and failure in acute heart failure: from pathophysiology to diagnosis and management. a review on behalf of the acute heart failure committee of the heart failure association (hfa) of the european society of cardiology (esc)*, *European journal of heart failure* **19** (2017), no. 7 821–836.

[8] Z.-K. Wang, R.-J. Chen, S.-L. Wang, G.-W. Li, Z.-Z. Zhu, Q. Huang, Z.-L. Chen, F.-C. Chen, L. Deng, X.-P. Lan, *et. al.*, *Clinical application of a novel diagnostic scheme including pancreatic β-cell dysfunction for traumatic multiple organ dysfunction syndrome*, *Molecular medicine reports* **17** (2018), no. 1 683–693.

[9] R. M. Durham, J. Moran, J. E. Mazuski, M. J. Shapiro, A. E. Baue, and L. M. Flint, *Multiple organ failure in trauma patients*, *Journal of Trauma and Acute Care Surgery* **55** (2003), no. 4 608–616.

[10] A. Ulvik, R. Kvåle, T. Wentzel-Larsen, and H. Flaatten, *Multiple organ failure after trauma affects even long-term survival and functional status*, *Critical Care* **11** (2007), no. 5 1–8.

[11] P. S. Barie, L. J. Hydo, and E. Fischer, *A prospective comparison of two multiple organ dysfunction/failure scoring systems for prediction of mortality in critical surgical illness.*, *The Journal of trauma* **37** (1994), no. 4 660–666.

[12] D. Peres Bota, C. Melot, F. Lopes Ferreira, V. Nguyen Ba, and J.-L. Vincent, *The multiple organ dysfunction score (mods) versus the sequential organ failure assessment (sofa) score in outcome prediction*, *Intensive care medicine* **28** (2002) 1619–1624.

[13] D. C. Dewar, A. White, J. Attia, S. M. Tarrant, K. L. King, and Z. J. Balogh, *Comparison of postinjury multiple-organ failure scoring systems: Denver versus sequential organ failure assessment*, *Journal of trauma and acute care surgery* **77** (2014), no. 4 624–629.

[14] L. Hutchings, P. Watkinson, J. D. Young, and K. Willett, *Defining multiple organ failure after major trauma: a comparison of the denver, sequential organ failure assessment and marshall scoring systems*, *The journal of trauma and acute care surgery* **82** (2017), no. 3 534.

[15] A. Sauaia, F. A. Moore, E. E. Moore, J. M. Norris, D. C. Lezotte, and R. F. Hamman, *Multiple organ failure can be predicted as early as 12 hours after injury*, *Journal of Trauma and Acute Care Surgery* **45** (1998), no. 2 291–303.

[16] J. A. Vogel, M. M. Liao, E. Hopkins, N. Seleno, R. L. Byyny, E. E. Moore, C. Gravitz, and J. S. Haukoos, *Prediction of postinjury multiple-organ failure in the emergency department: development of the denver emergency department trauma organ failure score*, *The journal of trauma and acute care surgery* **76** (2014), no. 1 140.

[17] Z. Obermeyer and E. J. Emanuel, *Predicting the future—big data, machine learning, and clinical medicine*, *The New England journal of medicine* **375** (2016), no. 13 1216.

[18] J. A. Cruz and D. S. Wishart, *Applications of machine learning in cancer prediction and prognosis*, *Cancer informatics* **2** (2006) 117693510600200030.

[19] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, *Machine learning applications in cancer prognosis and prediction*, *Computational and structural biotechnology journal* **13** (2015) 8–17.

[20] H. Asri, H. Mousannif, H. Al Moatassime, and T. Noel, *Using machine learning algorithms for breast cancer risk prediction and diagnosis*, *Procedia Computer Science* **83** (2016) 1064–1069.

[21] K. Sharma, A. Kaur, and S. Gujral, *Brain tumor detection based on machine learning algorithms*, *International Journal of Computer Applications* **103** (2014), no. 1.

[22] Z. Wang, G. Yu, Y. Kang, Y. Zhao, and Q. Qu, *Breast tumor detection in digital mammography based on extreme learning machine*, *Neurocomputing* **128** (2014) 175–184.

[23] M. De Bruijne, *Machine learning approaches in medical image analysis: From detection to diagnosis*, 2016.

[24] Y. Wang, Y. Zhao, and L. Petzold, *An empirical study on the robustness of the segment anything model (sam)*, *arXiv preprint arXiv:2305.06422* (2023).

[25] C. R. Farrar and K. Worden, *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

[26] K. Worden and G. Manson, *The application of machine learning to structural health monitoring*, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **365** (2007), no. 1851 515–537.

[27] Z. Ahmed, K. Mohamed, S. Zeeshan, and X. Dong, *Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine*, *Database* **2020** (2020).

[28] K. J. Janssen, A. R. T. Donders, F. E. Harrell Jr, Y. Vergouwe, Q. Chen, D. E. Grobbee, and K. G. Moons, *Missing covariate data in medical research: to impute is better than to ignore*, *Journal of clinical epidemiology* **63** (2010), no. 7 721–727.

[29] E. Tuba, I. Strumberger, T. Bezdan, N. Bacanin, and M. Tuba, *Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine*, *Procedia Computer Science* **162** (2019) 307–315.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *Smote: synthetic minority over-sampling technique*, *Journal of artificial intelligence research* **16** (2002) 321–357.

[31] I. Mani and I. Zhang, *knn approach to unbalanced data distributions: a case study involving information extraction*, in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126, pp. 1–7, ICML, 2003.

[32] G. E. Batista, A. L. Bazzan, M. C. Monard, *et. al.*, *Balancing training data for automated annotation of keywords: a case study.*, in *WOB*, pp. 10–18, 2003.

[33] H. Dağ, K. Sayin, I. Yenidoğan, S. Albayrak, and C. Acar, *Comparison of feature selection algorithms for medical data*, in *2012 International Symposium on Innovations in Intelligent Systems and Applications*, pp. 1–5, IEEE, 2012.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et. al.*, *Scikit-learn: Machine learning in python*, *the Journal of machine Learning research* **12** (2011) 2825–2830.

[35] J. Bakker, P. Gris, M. Coffernils, R. J. Kahn, and J.-L. Vincent, *Serial blood lactate levels can predict the development of multiple organ failure following septic shock*, *The American journal of surgery* **171** (1996), no. 2 221–226.

[36] G. I. Papachristou, V. Muddana, D. Yadav, M. O'connell, M. K. Sanders, A. Slivka, and D. C. Whitcomb, *Comparison of bisap, ranson's, apache-ii, and ctsi scores in predicting organ failure, complications, and mortality in acute pancreatitis*, *Official journal of the American College of Gastroenterology— ACG* **105** (2010), no. 2 435–441.

[37] W. Chen, T.-Y. Liu, Y. Lan, Z.-M. Ma, and H. Li, *Ranking measures and loss functions in learning to rank*, *Advances in Neural Information Processing Systems* **22** (2009).

[38] M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, *et. al.*, *The third international consensus definitions for sepsis and septic shock (sepsis-3)*, *Jama* **315** (2016), no. 8 801–810.

[39] V. Liu, G. J. Escobar, J. D. Greene, J. Soule, A. Whippy, D. C. Angus, and T. J. Iwashyna, *Hospital deaths in patients with sepsis from 2 independent cohorts*, *Jama* **312** (2014), no. 1 90–92.

[40] T. Desautels, J. Calvert, J. Hoffman, M. Jay, Y. Kerem, L. Shieh, D. Shimabukuro, U. Chettipally, M. D. Feldman, C. Barton, *et. al.*, *Prediction of sepsis in the intensive care unit with minimal electronic health record data: a machine learning approach*, *JMIR medical informatics* **4** (2016), no. 3 e5909.

[41] M. Saqib, Y. Sha, and M. D. Wang, *Early prediction of sepsis in emr records using traditional ml techniques and deep learning lstm networks*, in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4038–4041, IEEE, 2018.

[42] A. J. Masino, M. C. Harris, D. Forsyth, S. Ostapenko, L. Srinivasan, C. P. Bonafide, F. Balamuth, M. Schmatz, and R. W. Grundmeier, *Machine learning models for early sepsis recognition in the neonatal intensive care unit using readily available electronic health record data*, *PloS one* **14** (2019), no. 2 e0212665.

[43] Z. C. Lipton, D. Kale, and R. Wetzel, *Directly modeling missing data in sequences with rnns: Improved classification of clinical time series*, in *Machine learning for healthcare conference*, pp. 253–270, PMLR, 2016.

[44] J. Feng, C. Shaib, and F. Rudzicz, *Explainable clinical decision support from text*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1478–1489, 2020.

[45] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, *Mimic-iii, a freely accessible critical care database*, *Scientific data* **3** (2016), no. 1 1–9.

[46] T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, *The eicu collaborative research database, a freely available multi-center database for critical care research*, *Scientific data* **5** (2018), no. 1 1–13.

[47] J. Patrick and M. Li, *High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge*, *Journal of the American Medical Informatics Association* **17** (2010), no. 5 524–527.

[48] L. Deleger, K. Molnar, G. Savova, F. Xia, T. Lingren, Q. Li, K. Marsolo, A. Jegga, M. Kaiser, L. Stoutenborough, *et. al.*, *Large-scale evaluation of automated clinical note de-identification and its impact on information extraction*, *Journal of the American Medical Informatics Association* **20** (2013), no. 1 84–94.

[49] T. R. Goodwin and S. M. Harabagiu, *Medical question answering for clinical decision support*, in *Proceedings of the 25th ACM international on conference on information and knowledge management*, pp. 297–306, 2016.

[50] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, *arXiv preprint arXiv:1810.04805* (2018).

[51] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, *arXiv preprint arXiv:1310.4546* (2013).

[52] J. Pennington, R. Socher, and C. D. Manning, *Glove: Global vectors for word representation*, in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[53] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, *Biobert: a pre-trained biomedical language representation model for biomedical text mining*, *Bioinformatics* **36** (2020), no. 4 1234–1240.

[54] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, *Publicly available clinical bert embeddings*, *arXiv preprint arXiv:1904.03323* (2019).

[55] Z. Liu, L. Wu, and M. Hauskrecht, *Modeling clinical time series using gaussian process sequences*, in *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 623–631, SIAM, 2013.

[56] Z. Liu and M. Hauskrecht, *Clinical time series prediction: Toward a hierarchical dynamical system framework*, *Artificial intelligence in medicine* **65** (2015), no. 1 5–18.

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

[58] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias, *Attend and diagnose: Clinical time series analysis using attention models*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[59] Y.-H. H. Tsai, P. P. Liang, A. Zadeh, L.-P. Morency, and R. Salakhutdinov, *Learning factorized multimodal representations*, *arXiv preprint arXiv:1806.06176* (2018).

[60] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, *Multimodal machine learning: A survey and taxonomy*, *IEEE transactions on pattern analysis and machine intelligence* **41** (2018), no. 2 423–443.

[61] Y. Xu, S. Biswal, S. R. Deshpande, K. O. Maher, and J. Sun, *Raim: Recurrent attentive and intensive model of multimodal patient monitoring data*, in *Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*, pp. 2565–2573, 2018.

[62] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, *et. al.*, *Scalable and accurate deep learning with electronic health records*, *NPJ Digital Medicine* **1** (2018), no. 1 1–10.

[63] Y. Zhao, Q. Hong, X. Zhang, Y. Deng, Y. Wang, and L. Petzold, *Bertsurv: Bert-based survival models for predicting outcomes of trauma patients*, *arXiv preprint arXiv:2103.10928* (2021).

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, *arXiv preprint arXiv:1706.03762* (2017).

[65] K. Huang, J. Altosaar, and R. Ranganath, *Clinicalbert: Modeling clinical notes and predicting hospital readmission*, *arXiv preprint arXiv:1904.05342* (2019).

[66] Y. Kim, *Convolutional neural networks for sentence classification*, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, 2014.

[67] A. Trask, D. Gilmore, and M. Russell, *Modeling order in neural word embeddings at scale*, in *International Conference on Machine Learning*, pp. 2266–2275, PMLR, 2015.

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

[69] D. C. Angus, W. T. Linde-Zwirble, J. Lidicker, G. Clermont, J. Carcillo, and M. R. Pinsky, *Epidemiology of severe sepsis in the united states: analysis of incidence, outcome, and associated costs of care*, *Read Online: Critical Care Medicine— Society of Critical Care Medicine* **29** (2001), no. 7 1303–1310.

[70] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et. al.*, *Pytorch: An imperative style, high-performance deep learning library*, *arXiv preprint arXiv:1912.01703* (2019).

[71] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, *arXiv preprint arXiv:1412.6980* (2014).

[72] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural computation* **9** (1997), no. 8 1735–1780.

[73] M. Schuster and K. K. Paliwal, *Bidirectional recurrent neural networks*, *IEEE transactions on Signal Processing* **45** (1997), no. 11 2673–2681.

[74] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, *On the properties of neural machine translation: Encoder-decoder approaches*, *arXiv preprint arXiv:1409.1259* (2014).

[75] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, *Enriching word vectors with subword information*, *Transactions of the Association for Computational Linguistics* **5** (2017) 135–146.

[76] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, *Deep contextualized word representations*, *arXiv preprint arXiv:1802.05365* (2018).

[77] H. L. Li-wei, R. P. Adams, L. Mayaud, G. B. Moody, A. Malhotra, R. G. Mark, and S. Nemati, *A physiological time series dynamics-based approach to patient monitoring and outcome prediction*, *IEEE journal of biomedical and health informatics* **19** (2014), no. 3 1068–1076.

[78] Y. Zhao, Y. Wang, J. Liu, H. Xia, Z. Xu, Q. Hong, Z. Zhou, and L. Petzold, *Empirical quantitative analysis of covid-19 forecasting models*, in *2021 International Conference on Data Mining Workshops (ICDMW)*, pp. 517–526, IEEE, 2021.

[79] H. Liu and Z. Long, *An improved deep learning model for predicting stock market price time series*, *Digital Signal Processing* **102** (2020) 102741.

[80] T.-c. Fu, *A review on time series data mining*, *Engineering Applications of Artificial Intelligence* **24** (2011), no. 1 164–181.

[81] B. Gao, X. Li, W. L. Woo, and G. yun Tian, *Physics-based image segmentation using first order statistical properties and genetic algorithm for inductive thermography imaging*, *IEEE Transactions on Image Processing* **27** (2017), no. 5 2160–2175.

[82] B. Hu, B. Gao, W. L. Woo, L. Ruan, J. Jin, Y. Yang, and Y. Yu, *A lightweight spatial and temporal multi-feature fusion network for defect detection*, *IEEE Transactions on Image Processing* **30** (2020) 472–486.

[83] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, *Deep learning for time series classification: a review*, *Data mining and knowledge discovery* **33** (2019), no. 4 917–963.

[84] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, *The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances*, *Data Mining and Knowledge Discovery* **35** (2021), no. 2 401–449.

[85] M. Hüsken and P. Stagge, *Recurrent neural networks for time series classification*, *Neurocomputing* **50** (2003) 223–235.

[86] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, *Convolutional neural networks for time series classification*, *Journal of Systems Engineering and Electronics* **28** (2017), no. 1 162–169.

[87] J. Lines, S. Taylor, and A. Bagnall, *Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles*, *ACM transactions on knowledge discovery from data* **12** (2018), no. 5.

[88] A. Dempster, F. Petitjean, and G. I. Webb, *Rocket: exceptionally fast and accurate time series classification using random convolutional kernels*, *Data Mining and Knowledge Discovery* **34** (2020), no. 5 1454–1495.

[89] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, *A transformer-based framework for multivariate time series representation learning*, in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2114–2124, 2021.

[90] J. Lines and A. Bagnall, *Time series classification with ensembles of elastic distance measures*, *Data Mining and Knowledge Discovery* **29** (2015) 565–592.

[91] P. Schäfer, *Scalable time series classification*, *Data Mining and Knowledge Discovery* **30** (2016), no. 5 1273–1298.

[92] B. D. Fulcher and N. S. Jones, *Highly comparative feature-based time-series classification*, *IEEE Transactions on Knowledge and Data Engineering* **26** (2014), no. 12 3026–3037.

[93] A. Abanda, U. Mori, and J. A. Lozano, *A review on distance based time series classification*, *Data Mining and Knowledge Discovery* **33** (2019), no. 2 378–412.

[94] Z. Wang, W. Yan, and T. Oates, *Time series classification from scratch with deep neural networks: A strong baseline*, in *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585, IEEE, 2017.

[95] S. Lin and G. C. Runger, *Gcrnn: Group-constrained convolutional recurrent neural network*, *IEEE transactions on neural networks and learning systems* **29** (2017), no. 10 4709–4718.

[96] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, *Inceptiontime: Finding alexnet for time series classification*, *Data Mining and Knowledge Discovery* **34** (2020), no. 6 1936–1962.

[97] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, *Tapnet: Multivariate time series classification with attentional prototypical network*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6845–6852, 2020.

[98] P. Bloomfield, *Fourier analysis of time series: an introduction.* John Wiley & Sons, 2004.

[99] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, *Time-series anomaly detection service at microsoft*, in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3009–3017, 2019.

[100] T. Puech, M. Boussard, A. D'Amato, and G. Millerand, *A fully automated periodicity detection in time series*, in *Advanced Analytics and Learning on Temporal Data: 4th ECML PKDD Workshop, AALTD 2019, Würzburg, Germany, September 20, 2019, Revised Selected Papers 4*, pp. 43–54, Springer, 2019.

[101] G. J. Janacek, A. J. Bagnall, and M. Powell, *A likelihood ratio distance measure for the similarity between the fourier transform of time series*, in *Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005. Proceedings 9*, pp. 737–743, Springer, 2005.

[102] R. Geerken, B. Zaitchik, and J. Evans, *Classifying rangeland vegetation type and coverage from ndvi time series using fourier filtered cycle similarity*, *International Journal of Remote Sensing* **26** (2005), no. 24 5535–5554.

[103] K. Samiee, P. Kovacs, and M. Gabbouj, *Epileptic seizure classification of eeg time-series using rational discrete short-time fourier transform*, *IEEE transactions on Biomedical Engineering* **62** (2014), no. 2 541–552.

[104] J. Oh, J. Wang, and J. Wiens, *Learning to exploit invariances in clinical time-series data using sequence transformer networks*, in *Machine learning for healthcare conference*, pp. 332–347, PMLR, 2018.

[105] M. Liu, S. Ren, S. Ma, J. Jiao, Y. Chen, Z. Wang, and W. Song, *Gated transformer networks for multivariate time series classification*, *arXiv preprint arXiv:2103.14438* (2021).

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

[107] M. Tan and Q. Le, *Efficientnet: Rethinking model scaling for convolutional neural networks*, in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[108] A. Brock, S. De, S. L. Smith, and K. Simonyan, *High-performance large-scale image recognition without normalization*, in *International Conference on Machine Learning*, pp. 1059–1071, PMLR, 2021.

[109] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, *Bottleneck transformers for visual recognition*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16519–16529, 2021.

[110] M. Tan and Q. Le, *Efficientnetv2: Smaller models and faster training*, in *International conference on machine learning*, pp. 10096–10106, PMLR, 2021.

[111] Y. LeCun, J. Denker, and S. Solla, *Optimal brain damage*, *Advances in neural information processing systems* **2** (1990).

[112] S. Han, J. Pool, J. Tran, and W. Dally, *Learning both weights and connections for efficient neural network*, *Advances in neural information processing systems* **28** (2015).

[113] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, *Pruning filters for efficient convnets*, *arXiv preprint arXiv:1608.08710* (2016).

[114] Y. Censor, *Pareto optimality in multiobjective problems*, *Applied Mathematics and Optimization* **4** (1977), no. 1 41–59.

[115] A. Shah, S. Oczkowski, C. Aubron, A. P. Vlaar, J. C. Dionne, E. T. T. Force, S. de Bruin, M. Wijnberge, M. Antonelli, P. Aries, *et. al.*, *Transfusion in critical care: Past, present and future*, *Transfusion Medicine* **30** (2020), no. 6 418–432.

[116] L.-P. Liu, Q.-Y. Zhao, J. Wu, Y.-W. Luo, H. Dong, Z.-W. Chen, R. Gui, and Y.-J. Wang, *Machine learning for the prediction of red blood cell transfusion in patients during or after liver transplantation surgery*, *Frontiers in medicine* **8** (2021) 81.

[117] A. Mitterecker, A. Hofmann, K. M. Trentino, A. Lloyd, M. F. Leahy, K. Schwarzbauer, T. Tschoellitsch, C. Böck, S. Hochreiter, and J. Meier, *Machine learning–based prediction of transfusion*, *Transfusion* **60** (2020), no. 9 1977–1986.

[118] S. Walczak and V. Velanovich, *Prediction of perioperative transfusions using an artificial neural network*, *PloS one* **15** (2020), no. 2 e0229450.

[119] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau, *Benchmarking batch deep reinforcement learning algorithms*, *arXiv preprint arXiv:1910.01708* (2019).

[120] L. Li, M. Komorowski, and A. A. Faisal, *Optimizing sequential medical treatments with auto-encoding heuristic search in pomdps*, *arXiv preprint arXiv:1905.07465* (2019).

[121] A. Pape, P. Stein, O. Horn, and O. Habler, *Clinical evidence of blood transfusion effectiveness*, *Blood Transfusion* **7** (2009), no. 4 250.

[122] G. Lample and D. S. Chaplot, *Playing fps games with deep reinforcement learning*, in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[123] J. Kober, J. A. Bagnell, and J. Peters, *Reinforcement learning in robotics: A survey*, *The International Journal of Robotics Research* **32** (2013), no. 11 1238–1274.

[124] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, *Applications of deep learning and reinforcement learning to biological data*, *IEEE transactions on neural networks and learning systems* **29** (2018), no. 6 2063–2079.

[125] Y. Goldberg and M. R. Kosorok, *Q-learning with censored data*, *Annals of statistics* **40** (2012), no. 1 529.

[126] D. Ernst, G.-B. Stan, J. Goncalves, and L. Wehenkel, *Clinical data based optimal sti strategies for hiv: a reinforcement learning approach*, in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 667–672, IEEE, 2006.

[127] A. Raghu, M. Komorowski, I. Ahmed, L. Celi, P. Szolovits, and M. Ghassemi, *Deep reinforcement learning for sepsis treatment*, *arXiv preprint arXiv:1711.09602* (2017).

[128] N. Prasad, L.-F. Cheng, C. Chivers, M. Draugelis, and B. E. Engelhardt, *A reinforcement learning approach to weaning of mechanical ventilation in intensive care units*, *arXiv preprint arXiv:1704.06300* (2017).

[129] C. Yu, J. Liu, S. Nemati, and G. Yin, *Reinforcement learning in healthcare: A survey*, *ACM Computing Surveys (CSUR)* **55** (2021), no. 1 1–36.

[130] A. Y. Ng, D. Harada, and S. Russell, *Policy invariance under reward transformations: Theory and application to reward shaping*, in *Icml*, vol. 99, pp. 278–287, 1999.

[131] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et. al.*, *Deep q-learning from demonstrations*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[132] F. Fernández and M. Veloso, *Probabilistic policy reuse in a reinforcement learning agent*, in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 720–727, 2006.

[133] M. E. Taylor, P. Stone, and Y. Liu, *Transfer learning via inter-task mappings for temporal difference learning.*, *Journal of Machine Learning Research* **8** (2007), no. 9.

[134] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, *Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection*, *The International journal of robotics research* **37** (2018), no. 4-5 421–436.

[135] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, *The arcade learning environment: An evaluation platform for general agents*, *Journal of Artificial Intelligence Research* **47** (2013) 253–279.

[136] M. R. Kosorok and E. E. Moodie, *Adaptive treatment strategies in practice: planning trials and analyzing data for personalized medicine*. SIAM, 2015.

[137] H. Wei, G. Zheng, H. Yao, and Z. Li, *Intellilight: A reinforcement learning approach for intelligent traffic light control*, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2496–2505, 2018.

[138] A. Kadar, O. Chechik, E. Steinberg, E. Reider, and A. Sternheim, *Predicting the need for blood transfusion in patients with hip fractures*, *International orthopaedics* **37** (2013), no. 4 693–700.

[139] M. Ghassemi, M. Wu, M. C. Hughes, P. Szolovits, and F. Doshi-Velez, *Predicting intervention onset in the icu with switching state space models*, *AMIA Summits on Translational Science Proceedings* **2017** (2017) 82.

[140] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et. al.*, *Human-level control through deep reinforcement learning*, *nature* **518** (2015), no. 7540 529–533.

[141] C. J. C. H. Watkins, *Learning from delayed rewards*, .

[142] P. J. Huber, *Robust estimation of a location parameter*, in *Breakthroughs in statistics*, pp. 492–518. Springer, 1992.

[143] L.-J. Lin, *Self-improving reactive agents based on reinforcement learning, planning and teaching*, *Machine learning* **8** (1992), no. 3 293–321.

[144] D. Shung, J. Huang, E. Castro, J. K. Tay, M. Simonov, L. Laine, R. Batra, and S. Krishnaswamy, *Neural network predicts need for red blood cell transfusion for patients with acute gastrointestinal bleeding admitted to the intensive care unit*, *Scientific Reports* **11** (2021), no. 1 1–12.

[145] S. Van Buuren and K. Groothuis-Oudshoorn, *mice: Multivariate imputation by chained equations in r*, *Journal of statistical software* **45** (2011) 1–67.

[146] J.-L. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. Reinhart, P. Suter, and L. G. Thijs, *The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure*, 1996.

[147] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, *arXiv preprint arXiv:1412.3555* (2014).

[148] Y. Rubanova, R. T. Chen, and D. K. Duvenaud, *Latent ordinary differential equations for irregularly-sampled time series*, *Advances in neural information processing systems* **32** (2019).

[149] P. Kidger, J. Morrill, J. Foster, and T. Lyons, *Neural controlled differential equations for irregular time series*, *Advances in Neural Information Processing Systems* **33** (2020) 6696–6707.

[150] A. Lederman, R. Lederman, and K. Verspoor, *Tasks as needs: reframing the paradigm of clinical natural language processing research for real-world decision support*, *Journal of the American Medical Informatics Association* **29** (2022), no. 10 1810–1817.

[151] C. Zuheros, E. Martínez-Cámara, E. Herrera-Viedma, and F. Herrera, *Sentiment analysis based multi-person multi-criteria decision making methodology using natural language processing and deep learning for smarter decision aid. case study of restaurant choice using tripadvisor reviews*, *Information Fusion* **68** (2021) 22–36.

[152] Y.-H. Wang and G.-Y. Lin, *Exploring ai-healthcare innovation: Natural language processing-based patents analysis for technology-driven roadmapping*, *Kybernetes* (2022).

[153] L. Christensen, P. Haug, and M. Fiszman, *Mplus: a probabilistic medical language understanding system*, in *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain*, pp. 29–36, 2002.

[154] R. Pivovarov and N. Elhadad, *Automated methods for the summarization of electronic health records*, *Journal of the American Medical Informatics Association* **22** (2015), no. 5 938–947.

[155] J. Zeng, I. Banerjee, A. S. Henry, D. J. Wood, R. D. Shachter, M. F. Gensheimer, and D. L. Rubin, *Natural language processing to identify cancer treatments with electronic medical records*, *JCO Clinical Cancer Informatics* **5** (2021) 379–393.

[156] OpenAI, *Gpt-4 technical report*, 2023.

[157] J. Elias, *Google is asking employees to test potential chatgpt competitors, including a chatbot called 'apprentice bard'*, *CNBC. Archived from the original on February* **2** (2023) 2023.

[158] Q. Zhong, L. Ding, J. Liu, B. Du, and D. Tao, *Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert*, *arXiv preprint arXiv:2302.10198* (2023).

[159] W. Jiao, W. Wang, J.-t. Huang, X. Wang, and Z. Tu, *Is chatgpt a good translator? a preliminary study*, *arXiv preprint arXiv:2301.08745* (2023).

[160] J. Wang, Y. Liang, F. Meng, H. Shi, Z. Li, J. Xu, J. Qu, and J. Zhou, *Is chatgpt a good nlg evaluator? a preliminary study*, *arXiv preprint arXiv:2303.04048* (2023).

[161] S. Vilar, C. Friedman, and G. Hripcsak, *Detection of drug–drug interactions through data mining studies using clinical sources, scientific literature and social media*, *Briefings in bioinformatics* **19** (2018), no. 5 863–877.

[162] S. Gehrmann, F. Dernoncourt, Y. Li, E. T. Carlson, J. T. Wu, J. Welt, J. Foote Jr, E. T. Moseley, D. W. Grant, P. D. Tyler, *et. al.*, *Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives*, *PloS one* **13** (2018), no. 2 e0192360.

[163] M. Afshar, A. Phillips, N. Karnik, J. Mueller, D. To, R. Gonzalez, R. Price, R. Cooper, C. Joyce, and D. Dligach, *Natural language processing and machine learning to identify alcohol misuse from the electronic health record in trauma patients: development and internal validation*, *Journal of the American Medical Informatics Association* **26** (2019), no. 3 254–261.

[164] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, *arXiv preprint arXiv:1907.11692* (2019).

[165] S. Biswas, *Chatgpt and the future of medical writing*, 2023.

[166] T. H. Kung, M. Cheatham, A. Medenilla, C. Sillos, L. De Leon, C. Elepaño, M. Madriaga, R. Aggabao, G. Diaz-Candido, J. Maningo, *et. al.*, *Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models*, *PLoS digital health* **2** (2023), no. 2 e0000198.

[167] H. Nori, N. King, S. M. McKinney, D. Carignan, and E. Horvitz, *Capabilities of gpt-4 on medical challenge problems*, *arXiv preprint arXiv:2303.13375* (2023).

[168] S. B. Patel and K. Lam, *Chatgpt: the future of discharge summaries?*, *The Lancet Digital Health* **5** (2023), no. 3 e107–e108.

[169] Y. Juhn and H. Liu, *Artificial intelligence approaches using natural language processing to advance ehr-based clinical research*, *Journal of Allergy and Clinical Immunology* **145** (2020), no. 2 463–469.

[170] N. Grabar, C. Grouin, *et. al.*, *Year 2020 (with covid): Observation of scientific literature on clinical natural language processing*, *Yearbook of Medical Informatics* **30** (2021), no. 01 257–263.

[171] S. K. Mukhiya, U. Ahmed, F. Rabbi, K. I. Pun, and Y. Lamo, *Adaptation of idpt system based on patient-authored text data using nlp*, in *2020 IEEE 33rd international symposium on computer-based medical systems (CBMS)*, pp. 226–232, IEEE, 2020.

[172] H. Nayel and H. Shashirekha, *Improving ner for clinical texts by ensemble approach using segment representations*, in *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pp. 197–204, 2017.

[173] X. Lv, Y. Guan, J. Yang, and J. Wu, *Clinical relation extraction with deep learning*, *International Journal of Hybrid Information Technology* **9** (2016), no. 7 237–248.

[174] A. Romanov and C. Shivade, *Lessons from natural language inference in the clinical domain*, *arXiv preprint arXiv:1808.06752* (2018).

[175] Y. Wang, K. Verspoor, and T. Baldwin, *Learning from unlabelled data for clinical semantic textual similarity*, in *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pp. 227–233, 2020.

[176] H. Hassanzadeh, A. Nguyen, S. Karimi, and K. Chu, *Transferability of artificial neural networks for clinical document classification across hospitals: a case study on abnormality detection from radiology reports*, *Journal of biomedical informatics* **85** (2018) 68–79.

[177] S. Soni and K. Roberts, *Evaluation of dataset selection for pre-training and fine-tuning transformer language models for clinical question answering*, in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 5532–5538, 2020.

[178] Y. Peng, S. Yan, and Z. Lu, *Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets*, *arXiv preprint arXiv:1906.05474* (2019).

[179] A. Mascio, Z. Kraljevic, D. Bean, R. Dobson, R. Stewart, R. Bendayan, and A. Roberts, *Comparative analysis of text classification approaches in electronic health records*, *arXiv preprint arXiv:2005.06624* (2020).

[180] W. Digan, A. Névéol, A. Neuraz, M. Wack, D. Baudoin, A. Burgun, and B. Rance, *Can reproducibility be improved in clinical natural language processing? a study of 7 clinical nlp suites*, *Journal of the American Medical Informatics Association* **28** (2021), no. 3 504–515.

[181] Y. Shen, L. Heacock, J. Elias, K. D. Hentel, B. Reig, G. Shih, and L. Moy, *Chatgpt and other large language models are double-edged swords*, 2023.

[182] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et. al.*, *Language models are few-shot learners*, *Advances in neural information processing systems* **33** (2020) 1877–1901.

[183] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, *Large language models are zero-shot reasoners*, *arXiv preprint arXiv:2205.11916* (2022).

[184] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, *et. al.*, *Chain-of-thought prompting elicits reasoning in large language models*, in *Advances in Neural Information Processing Systems*, 2022.

[185] B. Wang, X. Deng, and H. Sun, *Iteratively prompt pre-trained language models for chain of thought*, in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2714–2730, 2022.

[186] R. I. Doğan, R. Leaman, and Z. Lu, *Ncbi disease corpus: a resource for disease name recognition and concept normalization*, *Journal of biomedical informatics* **47** (2014) 1–10.

[187] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu, *Biocreative v cdr task corpus: a resource for chemical disease relation extraction*, *Database* **2016** (2016).

[188] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, *2010 i2b2/va challenge on concepts, assertions, and relations in clinical text*, *Journal of the American Medical Informatics Association* **18** (2011), no. 5 552–556.

[189] I. Segura-Bedmar, P. Martínez Fernández, and M. Herrero Zazo, *Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013)*, Association for Computational Linguistics, 2013.

[190] G. Soğancıoğlu, H. Öztürk, and A. Özgür, *Biosses: a semantic sentence similarity estimation system for the biomedical domain*, *Bioinformatics* **33** (2017), no. 14 i49–i58.

[191] O. Uzuner, P. Szolovits, and I. Kohane, *i2b2 workshop on natural language processing challenges for clinical records*, in *Proceedings of the Fall Symposium of the American Medical Informatics Association*, Citeseer, 2006.

[192] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, *et. al.*, *An overview of the bioasq large-scale biomedical semantic indexing and question answering competition*, *BMC bioinformatics* **16** (2015), no. 1 1–28.