MODELING UNCERTAINTY IN DEEP LEARNING MODELS OF ELECTRONIC HEALTH RECORDS

by

Riyi Qiu

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Software and Information Systems

Charlotte

2020

Approved by:

_____
Dr. Mirsad Hadzikadic

_____
Dr. Michael Dulin

_____
Dr. Yaorong Ge

_____
Dr. Xi Niu

_____
Dr. Jeffery Kimble

ABSTRACT

RIYI QIU. Modeling uncertainty in deep learning models of Electronic Health Records. (Under the direction of DR. MIRSAD HADZIKADIC)

Recent research development has demonstrated the advantages of deep learning models in prediction tasks on electronic health records (EHR) in the medical domain. However, the prediction results tend to be difficult to explain due to the complex neuron structures. Without the explainability and transparency, deep learning models are not trustworthy or reliable for making real world decisions, especially the high-stakes ones in the healthcare domain. To improve the trustworthiness of the deep learning model, quantifying the uncertainty is crucial.

In this dissertation work, we proposed several Bayesian Neural Network (BNN) structures to estimate the data uncertainty and model uncertainty associated with the EHR data and deep learning models, respectively. We also proposed Variational Neural Network (VNN) algorithms to estimate the uncertainty of the variables to investigate the medical and temporal features that contribute the most to the patient-level uncertainty. In order to verify the validity of the uncertainty estimations, we designed a series of experiments to examine the computational results against widely accepted facts about uncertainty. We also conducted post-hoc analysis to evaluate whether the proposed models tend to specialize in one or more patient subgroups, at the cost of model performance on others, as well as whether the treatment (improving uncertainty in one subgroup) will mitigate such performance cost. The experiment results have confirmed the validity of our computational approaches. Finally, we conducted a user study to understand the clinicians' perception of the proposed uncertainty models.

ACKNOWLEDGEMENTS

This dissertation would not be possible without the help of many people. First of all, I would like to express my deepest gratitude to my advisor Dr. Mirsad Hadzikadic for his consistent support in this long journey. I am grateful for his patience, guidance, inspiration, and encouragement. He taught me how to perform academic research studies and understand the importance of big pictures. I also want to thank my co-advisor Dr. Michael Dulin, for his valuable instructions and feedback in healthcare applications. I would like to thank my dissertation committee members, Dr. Yaorong Ge, Dr. Xi Niu, and Dr. Jeff Kimble, for their inspiring guidance and advice on my course works and research studies. Without their kindness and help, I would not be able to complete this dissertation work.

Furthermore, I would like to thank Dr. Lixia Yao and Dr. Yugang Jia for their tremendous help and collaborations. The approaches they took to solve problems were always insightful and have inspired me to think more.

Also, I thank Ms. Julie Fulton and Ms. Sandra Krause for helping me and caring for me throughout my Ph.D. I also gratefully acknowledge the support I received from the Graduate School and the Department of Software and Information Systems.

Last but not least, I would like to thank my family and friends. My parents and my wife have been supporting me consistently. My baby girl Stella always fills our home with lovely smiles. My friends from childhood, high school, undergraduate, and UNCC, all have been supporting me in different ways throughout this long journey. I will always be grateful for having them in my life and being a part of their life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

The expeditious growth of Electronic Health Records (EHR) is motivating a large number of predictive models such as logistic regression and random forest to enhance healthcare quality [2, 3, 4]. However, EHR data are usually incomplete, noisy, heterogeneous, and sparse. To tackle these challenges and build predictive models with conventional machine learning techniques, solid feature selection and data representation are necessary. Deep learning [5] is well known for the end-to-end learning capabilities so feature engineering is automatically performed. It is also able to extract the temporal information from the time series data in EHR. Therefore, we have seen the deep learning models largely outperformed conventional machine learning techniques in the EHR-based predictive tasks over the past few years. Although the performance is significantly improved, the black-box mechanism of the deep neural network makes it difficult to explain the model output in the context of clinical use.

The research community was aware of the situation and has been developing techniques to distill explainable insights [6]. Most of the existing attempts, however, provided model explanations that are only insightful to the deep learning expert and did not take the expertise of the end-users into account [7]. As indicated by clinicians who are familiar with machine learning [8], the explanation given by the model should enhance the end-users' trust of the predictions and allow them to validate model outputs with domain knowledge.

Furthermore, clinicians identified several classes of explanations that will enhance their trust, including feature importance, individual-level explanations, uncertainty, temporal explanations, and transparent design [8]. Among these classes, uncertainty is important because it (1) provides explainability in the form of complementing

the output results and increases end-users' confidence and (2) is possible to identify the significant individual-level error and propagate it to the clinicians even when the overall model performance is good. In the high-stake acute care specialties such as the Intensive Care Unit (ICU), it is crucial to point out the possible mistake and let the users understand what the predictive model does not know. As a fundamental part of every machine learning phase [9, 10], uncertainty can be caused by the noisy data, model structure, or model parameters. In this dissertation, we propose to (1) build deep learning models with EHR data for the risk prediction of healthcare events such as disease onset, mortality, and hospital length of stay, (2) use deep BNN techniques to estimate the uncertainty of each patient, (3) investigate the relationships between uncertainty and the model performance at the population level, (4) estimate the uncertainty of temporal and medical features to make the model more explainable, (5) conduct a series of experiments for the uncertainty verification, and (6) perform a user study to evaluate the clinicians' perception of the uncertainty model. Our model will (1) improve the end-users' efficiency by complementing and screening the predictions that are correct, (2) identify the individuals that the model is unsure of the prediction, (3) identify the patient groups that can benefit the most from uncertainty mitigation, and (4) trace back the source of uncertainty to the feature level hence helping the clinicians understand how the prediction is made and why is it uncertain.

1.1 Background

1.1.1 Problem Statement

With the rapidly growing computing power and data volume, deep learning techniques have exhibited superior performance in various applications, including clinical predictive tasks. Despite their promising performance, deep learning models have some widely agreed limitations. The neuron structure and the high dependence on mathematical approximation results in a black box of the model learning process. With the lack of transparency, it is difficult for end-users, including health profes-

sionals, to understand the models' behaviors. To provide reasonable explanations and increase end-users' confidence in the results, it is crucial to identify when and what the trained model learns or does not learn, and how certain it is.

For safety-critical tasks such as mortality prediction in an Intensive Care Unit (ICU), the predictive model should be reliable in terms of knowing when it does not know. However, the deep learning models often assumed the prediction to be right and caused catastrophic consequences [1]. Quantifying the uncertainty is necessary to prevent such situations from happening: it acts as the confidence representations of the model and the end-users will not trust the prediction blindly when it is high. Knowing the importance of uncertainty and the fact that a normal deep learning model is not capable of capturing uncertainty, researchers in other domains such as computer vision managed to estimate it with approaches such as BNN [11, 1]. As for healthcare applications, the existing literature mostly focuses on capturing the uncertainty in the medical image processing and classification, which is quite similar to computer vision in terms of data format.

Moreover, the non-linear black-box structure of the model helps extract meaningful information from the data but makes the output difficult to understand as well. When the clinicians are making decisions and seeking the help of the deep learning model, for instance, simply providing the result of risk prediction to them is not enough. They need to understand how the model reaches the result and what are the factors that lead to it. Although the deep learning community has been developing methods to improve the model explainability, most of them are designed to be understandable only by data scientists or machine learning experts. Uncertainty, identified as one dimension of the model explainability [8], is possible to help the end-users understand what features have led to the prediction or what are the features that make the prediction uncertain.

To our best knowledge, there are only a few existing efforts studying uncertainty in

deep learning models for the EHR data. They are (1) Heo *et al.*'s study in 2018 [12] on the feature (variable) uncertainty in medical risk prediction tasks; (2) Dusenberry *et al.*'s research in 2019 [13] on Bayesian RNN with stochastic embedding to capture model uncertainty on the entire patient datasets and different patient subgroups; and (3) Tan *et al.*'s work in 2019 [14] on attention mechanisms to accommodate varying time intervals in time-series data, which they called "uncertainty".

Therefore, a comprehensive study and investigation of uncertainty is in need of the EHR-based deep learning risk prediction models.

1.1.2 Research Questions

Based on the literature, we proposed to comprehensively study this research question: given an EHR dataset and the predictive task, how do we account for the uncertainty in deep learning models? It was addressed by answering the following questions:

RQ1: In the context of deep learning model with EHR data, is it possible to estimate two major types of uncertainty, and how to understand their relationships with the deep learning model performance? By proposing this research question, we wanted to learn from the existing methods for the uncertainty estimation in other domains, especially in the computer vision applications [1, 11, 15]. We investigated how to adapt these models to make predictions on EHR data and estimate the uncertainty associated with each prediction simultaneously. Several models were proposed for the simultaneous estimation of data uncertainty and model uncertainty. We also evaluated whether estimating the uncertainty compromised the model performance with some key metrics such as the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), Area Under the Precision-Recall Curve (AUC-PR), and Expected Calibration Error (ECE). Furthermore, the relationships between uncertainty and model predictions were explored, aiming to find out the best way to benefit the real-world clinical decision-making process.

RQ2: What temporal or medical features are most responsible for the high uncertainty? This question was to address a research gap for the uncertainty in deep learning models. Researchers have emphasized the importance of modeling uncertainty and proposed methods to estimate it. However, these algorithms either only accounted for the uncertainty at the individual level or estimated the uncertainty at the pixel (feature) level with pre-assigned labels (such as semantic segmentation labeling in the computer vision applications). For the clinical risk prediction, there was no such label for each temporal or medical feature. Therefore with existing models, the doctors were given no explanations of the time and patient conditions that contributed the most to the uncertainty. To address this issue and give the clinicians more information, we proposed several variational architectures for the feature uncertainty estimation with the BNN model.

RQ3: How to make sure that all the uncertainty estimations are correct? How will the proposed models bring benefits to clinical decision making? Without the ground truth for patient and feature uncertainty, it was difficult to evaluate the estimated uncertainty. Inspired by the natures of the uncertainty sources, we performed a series of experiments to verify that the estimated uncertainty successfully captured the noises from data or models. Furthermore, we needed to validate that the estimated uncertainty provided better and more meaningful information to the clinicians. we distilled insights from several post-hoc patient sub-group analyses and feature-level uncertainty estimations. Part of these finds were embedded into a user survey we designed in order to obtain clinicians' feedback on the proposed deep learning models and uncertainty estimations.

1.1.3 Contribution to the Knowledge

This dissertation comprehensively investigated the uncertainty associated with the deep learning models for EHR-based risk prediction tasks. The outcome of the study is important for both clinical risk prediction tasks and corresponding deep learning

models. For the high-stake clinical risk prediction, while the accuracy of the model is crucial, it is more important that the model can provide an interval/distribution estimation or propagate the uncertainty to the end-users. Our contributions are: (1) The first study that applies the BNN algorithms to the EHR-based tasks and estimates both data uncertainty and model uncertainty simultaneously in one model. (2) Exploring the major types of uncertainty and their relationships to the EHR-based deep learning model predictions hence improving the model trustworthiness for clinical applications; (3) Estimating the uncertainty at the (medical/temporal) feature level and make the uncertainty model more explainable; (4) Designing a series of verification experiments and a user study for validating the uncertainty estimations.

1.2 Related Works

In this section, I will discuss the related works from the following aspects: (1) EHR datasets, specifically the "tabular" EHR datasets that will be used for analysis in this dissertation, (2) EHR-based predictive modeling, including conventional regression and machine learning models and deep learning models, (3) a summary of uncertainty modeling in other domain, especially in the computer vision domain, and (4) existing uncertainty modeling in the context of EHR-based deep learning.

1.2.1 EHR Datasets

EHR is an effective tool for managing patients' medical history, communicating with patients and providers, and maintaining good patient-physician relationships [16]. In US hospitals, the adoption rate of a basic EHR system has increased from 9.4% to 83.8% during 2008-2015 [17]. The growth of the EHR has enabled enormous research and studies on clinical risk prediction. Major EHR data formats include tabular records, images, and clinical notes (natural language). Among these data, tabular records can include admission records, demographics, diagnosis, procedures, medications, lab test results, billing information, and healthcare provider information.

Most of these data are well defined with coding systems such as International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM) and Clinical Classifications Software (CCS), hence are ready for predictive analysis. Following are the EHR database/datasets that will be used to develop predictive models in this dissertation work.

The Medical Information Mart for Intensive Care (MIMIC-III) [18] is a large database consisting of de-identified information related to patients admitted to the intensive care unit (ICU) at the Beth Israel Deaconess Medical Center over 11 years. It contains 38,597 adult patients with 49,785 hospital admissions. The database contains 26 tables, including information such as patient demographics, lab test results, diagnoses, procedures, medications, and doctor notes. Since the database enables various types of research topics and is free to the public, many studies have been done with its subsets. Therefore, it is viable to develop new techniques or algorithms on the MIMIC-III and compare the performance with the existing benchmark models.

The Research Resource for Complex Physiologic Signals, well-known as PhysioNet, is offering free access to large collections of physiological and clinical data and related open-source software. PhysioNet holds data challenges annually for researchers and students to address an unsolved clinical problem, which includes risk assessments of specific events or diseases such as in-hospital mortality [19] and early sepsis [20]. These datasets are well organized and only needs a few pre-processing steps before applying the machine learning or deep learning algorithms. It is also easy to find the benchmark models to compare with.

The IBM MarketScan®[1] Research Database is a series of databases that "fully integrate de-identified patient-level health data, workplace productivity, laboratory results, health risk assessments (HRAs), hospital discharges and electronic medical records (EMRs) into data sets available for healthcare research" Among the databases,

[1]

MarketScan commercial claims and encounters database contains claims data of employees and their dependents who are less than 65 years old. The database contains diverse longitudinal claim information such as patient demographics, diagnoses, procedures, medications, revenue codes, and healthcare service provider information. The volume of the database is much higher than the MIMIC-III: it covers over 122 million patients and over 28 billion records. Therefore, the model developed based on it will be more generalizable. However, access to the database is restricted and it is very costly to subscribe.

1.2.2 Predictive Modeling with Electronic Health Records

Compared to the traditional cohort studies designed for specific tasks, EHR data is messier and noisier because it collects data for all patients and only collects medical features that are considered necessary by the physicians, but it is not as time-consuming as the survey-type studies that followed patients for years. The EHR-based risk assessment usually covers many more features, more patients, and more time points than the cohort-study-based algorithms. It also enables multiple tasks within the same dataset and can be easily implemented. The common path to building an EHR-based model is defining the task, generating the cohort or case-control set, pre-processing the data, training the model with one of the popular algorithms such as generalized/regularized linear regression or random forest, and validating the results [21].

1.2.3 Conventional Models for EHR-based Risk Prediction

The most common models for the EHR-based risk prediction are the regression models, including generalized linear regression models and regularized regression models [21]. The studies that used regression models usually performed feature selection to reduce the number of input variables. On the other hand, machine learning algorithms such as random forest are also frequently applied and have achieved good

performance. Although studies using machine learning models were less likely to perform feature selection than the ones using regression models, most of them only used 20 or fewer variables.

In the EHR data, the patients' data are longitudinal. However, existing regression or machine learning models did not make the most of the temporal information. Most studies did not consider longitudinal data at all. The rest only utilized the temporal information partially such as taking the maximum, mean, median, or count of a variable along the time dimension [21]. With EHR, the development of a patient's health conditions is observed and this is an important strength compared to the cohort studies. It was proved that predictive models ignored temporal information performed much worse than the ones that can make full use of it [22].

1.2.4 Deep learning with Electronic Health Records

Deep learning has been applied to process EHR data including both structured (e.g. diagnosis, medications, laboratory tests) and unstructured (e.g. free-text clinical notes) data. Here we discuss the structured ones. As described in the previous section, conventional regression and machine learning algorithms ignore part of or all of the longitudinal data in the EHR. This is one of the major reasons that deep learning models have been outperforming these conventional algorithms. Deep learning is also well known for its capability of end-to-end study. The model can take as many variables as possible (restrained by the computing power) without feature selection and distill useful information.

There are two major deep neural network structures and both of them are proved with excellent performance in EHR-based tasks.

- **Convolutional Neural Network (CNN).** CNN is a class of neural networks that is commonly used to analyze image data. As a variant of the fully connected network (multi-layer perceptrons), CNN applies the 'convolutions' to enable the regularization. These convolution kernels are normally small blocks that absorb

local information and pass it down to the next layer of the network. Unlike the multi-dimension convolution kernels (e.g. 2x2, 3x3, 2x2x2, etc.) for analyzing the image or video data, analyzing time-series data such as the EHR only needs a 1-dimension kernel to capture the longitudinal information. Razavian and Sontag [23] proposed a 1-D CNN for the diagnosis prediction from lab tests, since applying CNN in tabular EHR data only requires the convolutions over the temporal dimension, and convolving over the medical feature dimension does not provide any meaningful information. In their later study [24], another 1-D CNN was used for the disease onset prediction to obtain a better performance than linear regression and conventional machine learning models. Che *et al.* [25] utilized the 1-D CNN architecture and medical feature embedding techniques to predict heart failures and diabetes with high accuracy. Similar works for other clinical risk prediction tasks also confirmed the better performance of 1-D CNN [26].

- **Recurrent Neural Network (RNN).** RNN is a class of neural networks that are commonly applied to analyze time-series data and sequence data. The nodes in the network are connected to form a directed graph that cycles the information for arbitrarily long time [27]. Among the many variants of RNN, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are most frequently seen in the EHR context. Both network structures are composed of "gates" that memorize numbers and signals over time. In one of the earliest works, Lipton *et al.* applied LSTM to classify 128 phenotypes. Nickerson *et al.* [28] managed to forecast analgesic response with the LSTM. Choi *et al.* first used GRU to detect heart failure onset [22], then in their later studies designed a "doctor AI" system that was able to provide diagnosis based on patients' medical history [29]. For better RNN model interpretation, Choi *et al.* [30] designed an attention-based architecture that can provide interpretations

for both temporal dimension and the medical feature dimension. Jagannatha and Hu [31] used the reversed-order output of RNN layers by using both Bidirectional LSTM and Bidirectional GRU to predict medical events. Compared with LSTM, GRU is faster to train since it has fewer gates; it is also proved to have better/comparable performance in most tasks, as claimed by Jozefowicz *et al.* [32]. Similar results were presented by Esteban *et al.* [33] in their work for clinical event prediction. In GRU/LSTM model, the output only depends on the hidden state of the last time step. Although it contains information from all previous time steps, a strong signal from early time steps may have weakened or vanished. Proposed by Howard and Ruder [34], pooling operations can improve the model performance by taking average/maximum/minimum of all hidden states. These operations were proved to be able to increase the model performance by 1%-2% [35].

1.2.5 Modeling Uncertainty in Deep Learning Models

Despite the promising performance, deep learning has limitations. The black-box structure makes it difficult for end-users to understand and trust the model's predicting behaviors. To increase the trustworthiness of the deep learning algorithms, modeling the uncertainty is crucial. Uncertainty is a fundamental part of every machine learning phase [9, 10]. Modeling uncertainty is critical in the cases of 'AI Failure' [36]: the self-driving vehicle can kill pedestrians or the Amazon recruiting tool can be gender or race biased. Similarly, a patient can be falsely recognized as 'low-risk' in the hospital. In these cases, If the deep learning model can yield high uncertainty along with the wrong predictions, such 'failures' could have been avoided. As concluded by Gal [11], there are several situations that can lead to uncertainty: out of distribution test data, noisy data, model structure, and model parameters. Based on these situations, two major types of uncertainty can be concluded [37, 11]: (1) aleatoric uncertainty, which is caused by the noisy data; (2) the epistemic uncertainty, which

includes the uncertainty from both model structure and model parameters. Furthermore, aleatoric uncertainty can be divided into homoscedastic uncertainty and heteroscedastic uncertainty. Homoscedastic uncertainty is captured independently of input data, while the heteroscedastic uncertainty is instance-dependent.

Bayesian Neural Network (BNN) methods have been adapted to capture the uncertainty in domains such as computer vision and natural language processing, e.g., [1, 11, 38]. It is robust to over-fitting, enables uncertainty estimation, provides more calibrated models, and can easily learn from small datasets [39]. The key idea of Bayesian modeling is to represent the model weights with some predefined prior distributions and to train the model to learn the probability density of the posteriors. Estimating the posterior requires calculating integration over the model parameters (also known as the process of inference). The process can be intractable to compute analytically. Various methods have been proposed to approximate the inference back in the 1990s. Some major works include Laplace Approximation [40], Minimal Description Length (Variational Inference) [41], Hamiltonian Monte Carlo [42], and Ensemble Learning [43]. However, for the massive datasets nowadays, these methods were not scalable [11]. To adapt BNN to modern applications, sampling-based algorithms or variational inference methods have been proposed, with good scalability. Graves *et al.* [44] applied data sub-sampling techniques to estimate the weights of Bayesian layers, but did not perform well in practice [45]. Blundell *et al.* [46] proposed Bayes by Backpropagation (BBB) that used a mixture of two Gaussian priors for Bayesian learning and largely improved the performance in practice. However, doubling the number of distributions increased the computational cost and made the model difficult to adapt to complex models. Hernandez-Lobato and Adams [45] proposed Probabilistic Back Propagation (PBP) that computed a forward propagation of probabilities followed by backward computation of gradients and outperformed state-of-the-art models in ten tasks. In his dissertation work, Gal [11] developed

an approximate inference technique that performs several stochastic forward passes through the model, and estimated the uncertainty by capturing sample mean and variance. The model scaled well to large data and could be adapted to different deep learning models without changing network structures. Gal [11]'s idea has inspired this study's four designs of the neural networks.

1.2.6 Modeling Uncertainty in EHR-based Deep Learning Models

In recent three years, there are a few existing studies that estimated data uncertainty or model uncertainty in the EHR data using deep learning models. Based on the well-known RETAIN model [30], Heo *et al.* [12] introduced the notion of input-dependent uncertainty to an attention mechanism, to generate an attention weight for each feature with different degrees of noise, to learn larger variance on instances the model is uncertain about. Their study was the first to investigate the feature-level (variable-level) uncertainty, which has great potential for richer interpretations of deep learning model results to assist clinicians. Compared to this study, their study's focus is feature-level uncertainty and attention weights in order to improve model prediction performance. Instead, this study focuses on different uncertainty: data uncertainty and model uncertainty as well as their relationships. Dusenberry *et al.* [13] used different approaches to capture the notion of model uncertainty, and found that a Bayesian RNN with stochastic embedding parameters is a more efficient way to capture model uncertainty compared to ensembles of a large number of deep learning models. They also analyzed how model uncertainty is impacted by patient subgroups by age and gender. Although this study dives deep on the subject of model uncertainty in the medical domain, they did not consider another major type of uncertainty in their analysis: data uncertainty. Tan *et al.* [14] proposed a novel Uncertainty-Aware Convolutional Recurrent Neural Network (UA-CRNN) that is able to accommodate varying time intervals in time series data in EHR records. They called this irregular varying time intervals as "uncertainty" in time series data.

Their definition of uncertainty is completely different from the data and model uncertainty in this chapter. In contrast to Dusenberry *et al.* [13]'s study, their study only focuses on data uncertainty. Without understanding model uncertainty, the other major type of uncertainty, the work did not paint a full picture of uncertainty in patients' EHR data.

Their work, although different definitions and approaches, all suggests the significance of studying EHR uncertainty. This study will continue to address the open research questions on computational approaches to capture both EHR aleatoric uncertainty and epistemic uncertainty at the same time, how to validate the results without any ground truth on uncertainty, uncertainty's impacts on model performance, uncertainty's effects on different patient demographic groups, as well as the implications for doctors.

1.3 Overview of the Dissertation

This dissertation is organized into five parts. In Chapter 2, we will explore the performance of two common deep learning models (CNN and GRU) for clinical risk prediction tasks using a huge commercial EHR database; In Chapter 3, we will build several BNN models to capture the patient level uncertainty, investigate the relationships between uncertainty estimations and model performance, conduct uncertainty verification experiments, and distill actionable insights for clinicians by several post-hoc patient sub-group analyses; In Chapter 4, we will construct variational layers for the estimations of temporal feature uncertainty, investigate the relationships between patient-level uncertainty and feature-level uncertainty, and verify the calculated uncertainty; In Chapter 5, we will design a user study that investigates the clinicians' perception of the predictive model with extra uncertainty information; finally, we will wrap up this dissertation with conclusions and future directions in Chapter 6.

CHAPTER 2: DEEP LEARNING MODELS FOR EHR-BASED PREDICTIVE TASKS

In this chapter, we applied CNN and GRU to a clinical predictive task: Total joint replacement (TJR). The deep learning models outperformed conventional machine learning models. These CNN and GRU models will serve as the baseline models for the BNN

2.1 Background

TJR is one of the most commonly performed elective surgical procedures in the United States, with over 1 million total hip and total knee replacement procedures performed each year [47]. The volume of primary and revision TJR procedures has risen continuously in recent decades. By 2030, primary total hip replacement (THR) is projected to grow 171% and primary total knee replacement (TKR) is projected to grow by up to 1 89%, for a projected 635,000 and 1.28 million procedures, respectively[48].

Given its volume and growth rate, the total cost of TJR has been scrutinized for opportunities to improve the margin of providers or reduce the healthcare burden of payers. One important finding is that there is a significant cost variation of TJR procedures. Based on a report published by Health Care Cost Institute (HCCI) in 2016[49], inpatient facility service of TJR is a top shoppable service in the United States for employer-sponsored insurance (ESI) population with age younger than 65, which accounts for 1.3% of total ESI spending in 2011. Another report published by BlueCross BlueShield (BCBSA) and Blue Health Intelligence (BHI) in 2015 showed that identical TJR procedures can quadruple in cost depending on which hospital is selected within a market. A more recent study[50] showed that the average cost of

care for total knee arthroplasty across the hospitals varied by a factor of about 2 to 1, despite having similar patient demographics and readmission and complication rates.

Based on those findings, various cost transparency tools have been developed to enable patients (consumers) to consume value through shopping, chosen the lower-priced higher-quality providers. Employers usually offer those tools to their employees through third parties or carriers for free. To maximize the return of investment on such tools, it is crucial to identify those who might benefit from such tools (e.g. people who need TJR surgery in the future) and engage them in time.

In view of this, we proposed to leverage claims data to identify the patients who might need a TJR surgery in the future. Compared with clinical data, the claims data are easy to obtain and deploy on a large scale, especially for non-clinical settings. However, the claims data are usually noisy, high-dimensional, sparse, incomplete, and heterogeneous [51, 22, 26, 25, 30, 52]. To tackle such challenges, researchers have been applying deep neural networks models such as Convolutional Neural Networks (CNN) [25, 26, 24, 23] and Recurrent Neural Networks (RNN) [51, 22, 30, 52, 53, 31, 24, 54, 55, 33, 28, 29] to predict the events.

In this chapter, we investigated the performance of various CNN and RNN algorithms to predict TJR on a large scale commercial claim dataset. More specifically, we are interested in the following aspects:

- Compared with baseline algorithms (LASSO and random forest), how much performance gain can we achieve by using the complex deep learning approach? The baseline algorithms aggregate the medical events along the time dimension hence losing the temporal and contextual information, while the deep learning based approach should be able to capture more complex structure of data at the expense of computational complexity.

- Which deep learning model is better for elective surgery prediction? It is well known that the RNN algorithm can do a better job in capturing longtime de-

pendency than that of CNN algorithm. However, the results from the literature were data dependent[56].

- How will data representation methods impact the performance of the deep learning model? We implemented two data representation methods in this chapter: multi-hot coding and embedding. Previous studies have shown mixed results for acute cases and we want to investigate its role for elective surgery[51, 22, 31].

- Will the hidden state information help our prediction task? In traditional RNN algorithm, only the last hidden state information will be used for prediction. Given that TJR is an elective procedure, it is possible that the patient may delay the procedure even if he has met the criteria. From this perspective, we believed the intermediate state information could also be useful.

2.2 Methodology

2.2.1 Data Description

Data were extracted from MarketScan[1] commercial claims and encounters database. It covers employees and their dependents with age less than 65 years old. The cohort is defined as follows:

- Fully enrolled in years 2014, 2015, and 2016.

- Diagnosed with Rheumatoid Arthritis/ Osteoarthritis based on CMS-CCW Chronic Condition Algorithms[2].

- No TJR surgery[3] in 2014 and 2015.

- Age over 45 in 2014.

The cohort of 540,000 patients were selected with around 3.5% positive cases (have a TJR surgery in 2016). The basic statistics of the dataset are listed in Table 2.1.

[1] ©2017 Truven Health Analytics LLC, All rights reserved
[2] https://www.ccwdata.org/documents/10280/19139608/ccw-cond-algo-arthritis.pdf
[3] TJR surgery is identified by DRG = 269 or DRG=270

Table 2.1: Basic statistics of the TJR dataset by year.

year	2014	2015
# patients with records	535,499	537,205
# days with events	18,300,352	19,863,997
# medical codes	134,071,176	146,802,340
Avg. days with events per patient	34	37
Avg. codes per patient	250	273

The following data elements from MarketScan database were selected as features for modeling purpose:

- **Demographic variables**: Age and gender. For deep learning models, the Demographic variables were concatenated with other variables in the last layer.
- **Diagnosis codes**: 283 distinct CCS diagnosis codes[4], mapped from both ICD-9-CM[5] and ICD-10-CM[6] codes in the MarketScan database.
- **Procedure codes**: 240 CCS procedure codes[7], mapped from both ICD-10-PCS codes[8], Current Procedural Terminology (CPT) and Healthcare Common Procedure Coding System (HCPCS) in the database.
- **Therapeutic classes**: 222 therapeutic classes which are derived from drug information by data vendor.
- **Revenue codes**: 651 standard revenue codes defined by the Health Care Finance Administration (HCFA).
- **Place of service**: 45 codes, such as pharmacy, home, ambulance, hospital, or other facilities.

[4]clinical classification software (CCS) provided by ARHQ https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

[5]International Classification of Disease Ninth Revision, Clinical Modification (ICD-9-CM)

[6]International Classification of Disease Tenth Revision, Clinical Modification (ICD-10-CM)

[7]clinical classification software (CCS) provided by ARHQ https://www.hcup-us.ahrq.gov/toolssoftware/ccs10/ccs10.jsp

[8]International Classification of Disease Tenth Revision, Procedure Coding System (ICD-10-PCS)

- **Provider types**. 131 provider types such as birthing center, radiology, or dentist.

- **Service sub-category codes**: 498 sub-service types such as mammograms, MRIs, or PET Scans etc.

2.2.2 EHR Data Representation

Since many EHR data are noisy, high-dimensional, and sparse, we first needed to explore the literature and find out efficient representation methods for the subsequent predictive tasks. Three data representation methods will be used in this dissertation. An aggregated occurrence vector will be applied on the conventional machine learning models; For the deep neural networks, we will evaluate the performance and efficiency of two representation methods: simple multi-hot encoding and medical feature embedding.

- **Aggregated occurrence vector**. We created a binary vector for each patient with length equals to the number of unique codes for each year. If a code appeared in that year, the corresponding elements in that vector will be set as 1 for this patient. The final feature vector for each patient was the concatenation of binary vectors for all years in the observation window. As mentioned in the works using conventional machine learning methods [57, 51], this data representation method did not take or takes little advantages of the longitudinal data.

- **Multi-hot encoding**. Every patient record was formed as a temporary-code binary matrix. The *(i, j)*th element of the matrix was 1 if *i*-th code appeared on the *j*-th day for a specific patient. The detailed explanation can be found in the works of Cheng *et al.*[26] and Che *et al.*[25].

- **Embedding**. To reduce the dimension of the feature matrix, we used Skip-Gram [58] method for code embedding[58, 59]. More specifically, we used a sliding time window of 14 days (can be any other number of days that make sense in the medical setting) to collect unique codes and reshuffled it as a 'sentence'. Detailed explanation was described by Choi *et al.*[60] and Farhan *et al.*[61]. The output embedding dimension was set to 100.

2.2.3 Deep Learning Model Structures

We used two aforementioned deep learning models for the experiments: CNN and RNN.

- **CNN** with only 1-d convolutions are commonly used to analyze time-series data such as the EHR and capture the longitudinal information [25, 26, 24, 23].

- **RNN** is commonly applied to analyze time-series data and sequence data. The nodes in the network are connected to form a directed graph that cycles the information for arbitrarily long time [27]. As mentioned in the previous chapter, we use GRU as the RNN model in this dissertation.

2.3 Experiments and Implementations

The script language used for the experiments was Python. The medical feature embedding was trained with Gensim. The neural network models were implemented in Keras and Tensorflow. For each implemented method, the result was provided by the mean and 95% confidence interval of a 5-fold cross validation.

2.3.1 Benchmark Model Implementations

The alpha of LASSO was set to 0.001 after performing a grid search. The detailed parameter for RF was as follows: the number of trees was 100; the maximum depth of the tree was 100; the minimum number of samples required to split an internal node was 10; the minimum number of samples required to be at a leaf node was 10; the

number of features to consider when looking for the best split was set to the square root of the number of total features.

2.3.2 Model Implementation

2.3.2.1 Implementation of CNN

The CNN model we used was similar to the work of Cheng *et al.* [26]. The network had 4 layers: The size of the input layer was the same as the number of features (codes); the second layer was a one dimensional convolutional layer, where convolutions of different sizes slided along the time axis and obtained features; the third layer performed dropout, pooling, and normalization operations, in order to fasten the computation and control over-fitting; the last layer was a fully connected (dense) output layer with a logistic regression to make the prediction. After tuning, we used 3 type of filters with filter length equaling 3, 4 and 5 and set the number of filters to 100. We used 'adam' as the optimizer and the learning rate was set to 0.001. The batch size was set as 250.

2.3.2.2 Implementation of RNN

We selected the gated recurrent unit (GRU) to implement the RNN for simplicity. As shown in Figure 2.1, given an input sequence x_t and the last hidden state h_{t-1} at each time step, GRU updates the hidden states h_t. GRU cell is built with sophisticated gating mechanism. It contains a reset gate r_t and an update gate z_t. The computations inside the solid line box of Figure 2.1 are as follows:

$$z_t = \sigma(U_z x_t + W_z h_{t-1} + b_z)$$

$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r)$$

$$\tilde{h}_t = \tanh(U_h x_t + r_t \odot W_h h_{t-1} + b_h)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

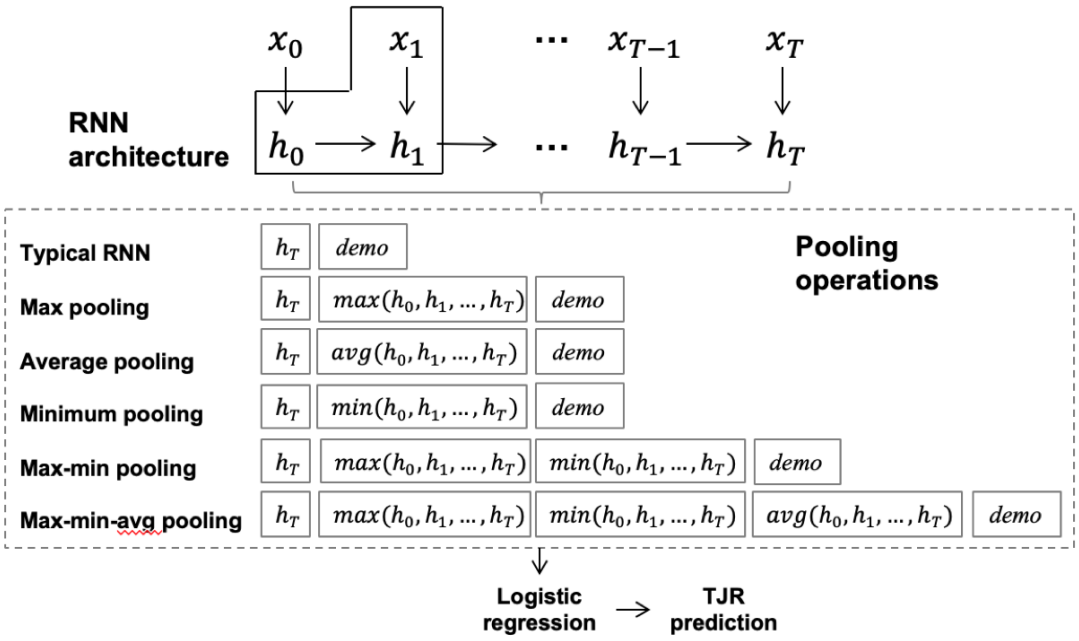

Figure 2.1: The RNN architecture and pooling operations.

where $\sigma()$ denotes the sigmoid function and $\odot$ is the operator for Hadamard product (*i.e.* element-wise multiplication); At each time step, the visit information x_t and hidden state of the last time step h_{t-1} are the inputs; three sets of U, W, b are the weights and biases to calculate two gates and the intermediate memory unit $\tilde{h}_t$. The sigmoid function makes the values of both gates between 0 and 1. The reset gate retains the useful information and drops the rest and the update gate decides how much of last hidden state to be passed onto the next state. We employ dropout on the final hidden state h_T for regularization, concatenate it with the patient's demographic data, and make the TJR prediction with logistic regression (Listed as the "Typical RNN" in Figure 2.1).

After tuning, we set the hyper-parameters as follows: the hidden layer dimension was set to 200; the dropout rate was 0.2; the optimizer, learning rate, and batch size were 'adam', 0.001, and 250, respectively.

2.3.2.3 Implementation of Pooling with RNN

In a typical GRU model, the output only depends on the logistic regression of the last hidden state h_T. Recently, researchers started to explore how to leverage other hidden state information to boost the performance [34, 35]. We also applied different pooling operations to all hidden state and concatenated it with the final state for final prediction. The rationale for using these sequences was that the positive or negative signal was often included in just a few visits at any point of the whole time period. The signal might have been ignored or weakened while it was passed to the last hidden state. As an elective surgery, the decision of TJR could possibly be made early but postponed due to other more urgent health problems. Three kinds of pooling operations are as follows:

- Max pooling layer. This layer outputs the maximum values of each dimension over the whole time period. By sending the strongest signal directly to the final state, the network becomes more sensitive to the important events.

- Average pooling layer. The idea of average pooling is to make the loss function consider all intermediate states, leading to better convergence and generalization.

- Minimum pooling layer. This layer is equivalent to *-maxpool(-x)*. Proposed by Skinner[35], min-pooling is supposed to enable the model to pass the other end of the activations in addition to the max-pooling. The network will become more “balanced" and “expressive".

It was possible to use one or multiple pooling strategies and concatenate them together to test its performance improvement, as shown in Figure 2.1.

Table 2.2: Performance comparison between baseline models and deep learning models.

Model	Trained with 2015 data		Trained with both 2014 and 2015 data	
	AUC	Precision@Recall=0.9	AUC	Precision@Recall=0.9
LASSO	0.7616 ± 0.0048	0.0527 ± 0.0003	0.7682 ± 0.0046	0.0532 ± 0.0013
RF	0.7853 ± 0.0050	0.0533 ± 0.0015	0.7887 ± 0.0040	0.0541 ± 0.0007
CNN-MH	0.8086 ± 0.0036	0.0572 ± 0.0012	0.8218 ± 0.0053	0.0645 ± 0.0015
RNN-MH	$\mathbf{0.8200 \pm 0.0073}$	$\mathbf{0.0577 \pm 0.0029}$	$\mathbf{0.8339 \pm 0.0024}$	$\mathbf{0.0662 \pm 0.0008}$

Table 2.3: Running time comparison between RNN-MH and RNN-EMB.

	Measurement	RNN-MH	RNN-EMB
Trained with 2015 data	Avg. training time per epoch	2589s	1028s
	No. of epoch to converge	3	7
	Total training time	7767s	7196s
Trained with both 2014 and 2015 data	Avg. training time per epoch	3450s	1246s
	No. of epoch to converge	4	11
	Total training time	13800s	13706s

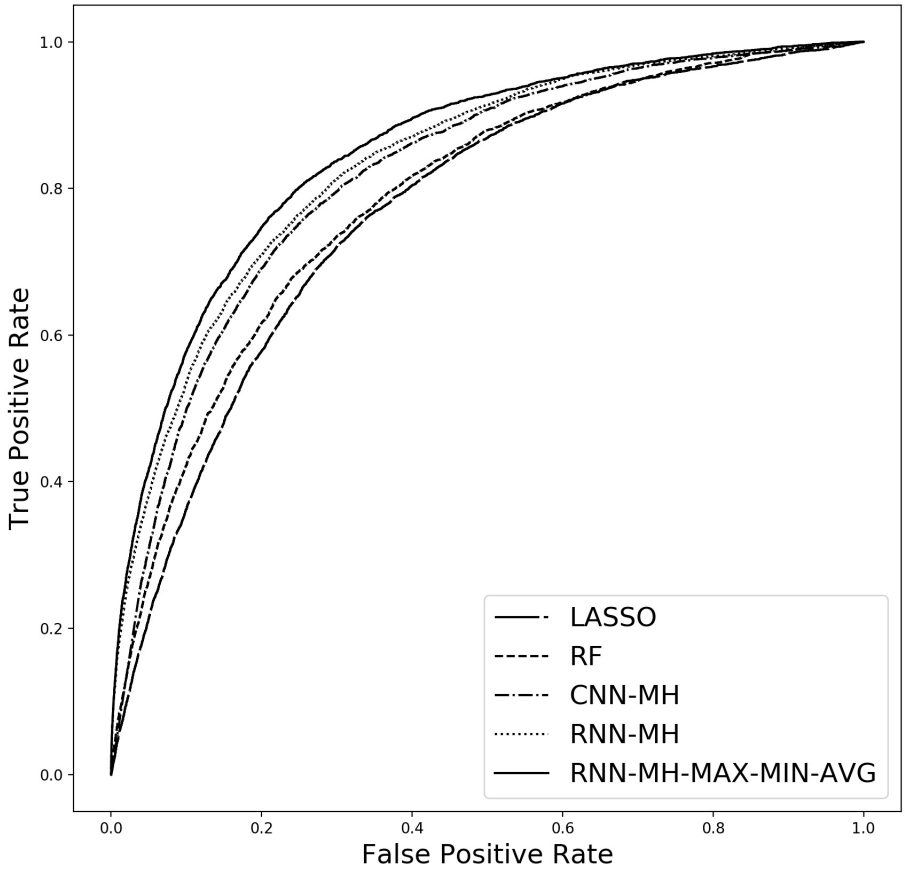

Figure 2.2: Comparison of ROC for different models trained with 2014 and 2015 data.

2.4 Results and Discussions

We perform experiments with two different observation windows. For the first setting, we only use 2015 data to predict TJR event in 2016. For the second setting, we use both 2014 and 2015 data to predict TJR event in 2016.

Two metrics are used for performance comparison. The first one is the area under the curve (AUC) which measures the overall performance of the model. The second one is precision with recall set to 0.9, which measures the real world performance of the model after consulting with business partners.

- The deep learning approach performs much better than traditional algorithms such as logistic regression and random forest (RF).

  Table 2.2 shows the performance of RF, LASSO, CNN with multi-hot coding (CNN-MH) and GRU with multi-hot coding (RNN-MH). The pair-wise t-test shows that the deep learning methods (CNN-MH and RNN-MH) perform significantly better than RF and LASSO in all scenarios with p=0.001.

  Another interesting observation is that the performance of RF and LASSO does not increase when more data (2014 data) was included. However, the performance of deep learning methods will increase significantly with more data. This indicates that the deep learning method is more capable of exploring complex relationships in time series data.

- The RNN based algorithm outperforms CNN based approach regardless of different representation (multi-hot or embedding).

  From Table 2.2, it is clear that RNN is much better than CNN in all scenarios, especially when 2014 data is included in the observation window. Pair-wise t-test shows that the difference is significant with p=0.001. In view of this, we will only investigate RNN algorithms from now on.

- The data representation has a limited effect on the performance and training efficiency of RNN algorithms.

  As shown in Table 2.3, the difference between RNN with two different data representation methods (RNN-EMB: RNN with embedding) are not significant in all scenarios. The training time per epoch for RNN-EMB is almost 2-3 times of that for RNN-MH. However, the RNN-MH converges much faster than RNN-EMB. As a result, the total training time of RNN-MH is similar to that of RNN-EMB in our experiments. As we formatted our input data as a 3-D matrix (the 3 dimensions are feature, time and patients respectively), the training time for both of models increases linearly with the number of patients, which is different compared with what is reported by Choi *et al.*[22].

- Additional maximum and minimum pooling mechanism can improve the performance of the RNN baseline algorithm. The best performance is achieved when we add pooling mechanism to RNN-MH algorithm.

  Table 2.4 shows the performance of RNN-MH and RNN-EMB with pooling methods. Pair-wise t-test demonstrates that the performance of RNN-EMB will be better if the maximum or minimum pooling are included with $p = 0.005$. However, there is no significant difference when the average pooling is used with RNN-EMB. When we use all three pooling methods, the performance of both RNN-MH and RNN-EMB are improved significantly.

In Figure 2.2, we plot the ROC curves of baseline models and three deep learning models with multi-hot encoding trained with 2014 and 2015 data. The figures demonstrate that deep learning methods are much better than the traditional methods and adding the pooling mechanisms further improve the performance of RNN.

Table 2.4: Performance of RNN with different pooling methods.

Model	Trained with 2015 data		Trained with both 2014 and 2015 data	
	AUC	Precision@Recall=0.9	AUC	Precision@Recall=0.9
RNN-MH	0.8200 ± 0.0073	0.0577 ± 0.0029	0.8339 ± 0.0024	0.0662 ± 0.0008
RNN-MH-MAX-MIN-AVG	$\mathbf{0.8289 \pm 0.0043}$	$\mathbf{0.0601 \pm 0.0015}$	$\mathbf{0.8423 \pm 0.0042}$	$\mathbf{0.0693 \pm 0.0025}$
RNN-EMB	0.8154 ± 0.0060	0.0574 ± 0.0016	0.8349 ± 0.0053	0.0668 ± 0.0020
RNN-EMB-MAX-MIN-AVG	0.8234 ± 0.0056	0.0591 ± 0.0019	0.8402 ± 0.0051	0.0685 ± 0.0024
RNN-EMB-MAX	0.8222 ± 0.0060	0.0594 ± 0.0018	0.8379 ± 0.0032	0.0681 ± 0.0017
RNN-EMB-MIN	0.8241 ± 0.0043	0.0598 ± 0.0018	0.8387 ± 0.0055	0.0675 ± 0.0020
RNN-EMB-AVG	0.8173 ± 0.0061	0.0575 ± 0.0017	0.8334 ± 0.0065	0.0667 ± 0.0032
RNN-EMB-MAX-MIN	0.8218 ± 0.0044	0.0591 ± 0.0014	0.8405 ± 0.0045	0.0687 ± 0.0020

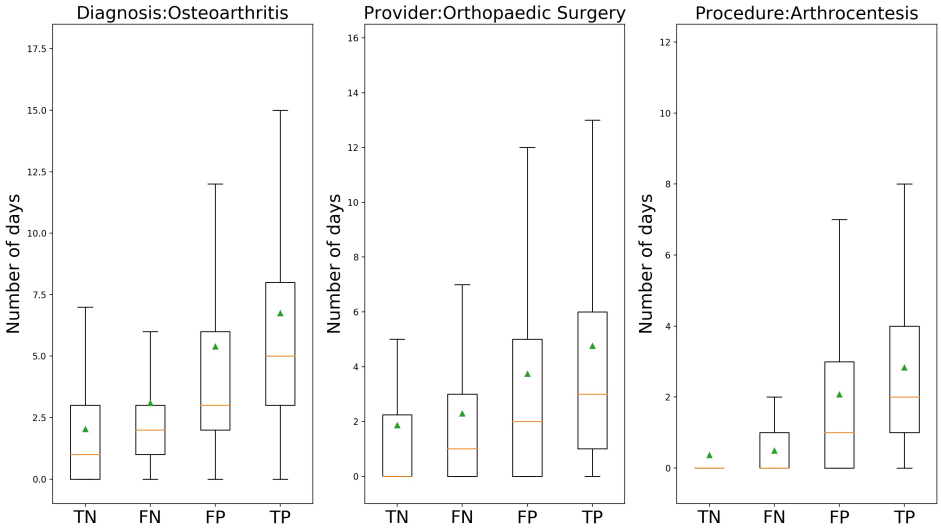

Figure 2.3: Explanation of model behavior with 3 examples. Outliers are not shown and the green triangle indicates the the mean of each group.

2.5 Model Explanation

We analyze the predictions made by the best model (RNN-MH-MAX-MIN-AVG) trained with both 2014 and 2015 data. Similar to what is described by Choi *et al.*[22], we sample 200 patients each from true positive (TP), false positive (FP), true negative (TN) and false negative (FN). This procedure is performed for all 5-folds, which give us 1000 patients in total. Then we calculate the number of days with those codes per patient. In the end, we compare their distribution across 4 categories (TP, FP, TN, and FN).

Figure 2.3 showed 3 representative codes that can explain the behavior of our prediction. The box-plot in each sub-figure showed the distribution of the number of days with that specific code per 4 categories (TP, FP, TN, and FN). The model assign more weights to the patients who have more interactions with health care provider, as shown in left sub-figure. More specifically, our model prefers those who got service from orthopedic surgery more often and have undergone the procedure of arthrocentesis more frequently.

2.6 Conclusions and Future Works

In this chapter, we investigated several deep learning methods to predict the TJR surgery based on a large commercial claims dataset with more than 2,000 variables and 540,000 patients. Without surprise, the performance of deep learning based approach is much better than traditional methods (e.g. random forest and LASSO). Among the investigated deep learning methods, the RNN with pooling mechanism worked the best for the use case. We tested two different data representation methods and discovered that the embedding techniques do not improve either the performance or the training efficiency of RNN in all scenarios. Our experiments also suggested that pooling mechanisms are able to discover the additional signals from the intermediate hidden states hence improving the performance of the baseline RNN algorithm.

In the rest chapters of this dissertation, we will develop the deep learning models from this chapter and estimate the uncertainty. Some of the settings that have been proved to be viable in this work will continue to be used. Due to the restricted access to the large Truven datasets, we will use the open-source MIMIC-III and PhysioNet-2012 datasets. Given that these datasets both have low numbers of medical features and medical embedding did not outperform multi-hot encoding in this chapter, we will only use multi-hot encoding for the following chapters. We will also use GRU with pooling mechanisms as the benchmark RNN model since pooling models will always outperform models without pooling.

CHAPTER 3: MODELING UNCERTAINTY OF EHR-BASED DEEP LEARNING MODELS

3.1 Background

Other than healthcare [26, 62], deep learning has a profound impact on various data-driven applications such as computer vision, natural language processing, and robotics [1, 63, 64]. It is well known for learning predictive artificial features from raw input, which largely reduces feature engineering efforts [5] and meanwhile distills meaningful information from complicated input data.

As mentioned in Chapter 1, deep learning models have some widely agreed limitations despite their promising performance. The neuron structure and the high dependence on mathematical approximation results in a black box of the model learning process. With the lack of transparency, it is difficult for end-users, including health professionals, to understand the models' behaviors. To provide reasonable explanations and increase end-users' confidence in the results, it is crucial to identify when and what the trained model learns or does not learn, and how certain it is. That is a concept we will study in this chapter: uncertainty in deep learning. Uncertainty plays a fundamental part in every phase of deep learning or machine learning in general [9, 10]. A machine learning algorithm with high uncertainty may cause negative or even catastrophic consequences: a self-driving vehicle could not classify a pedestrian correctly unless she is near the sidewalk; the Amazon AI recruiting tool showed bias against women, etc. Similarly in the context of healthcare, a predictive model could falsely label a patient as "low-risk" at the hospital admission. For these cases that predictive models made mistakes and the AI "failed", if we were able to derive an assessment of uncertainty on the results beforehand and communicate it to

the end-users, more attention could have been allocated and the undesired outcomes could have been avoided.

According to [37, 11], there are two major sources of uncertainty in machine learning. Uncertainty can be caused by the noise from data, such as a case that is out of distribution, a wrong observed label, an erroneous data, or an imputed missing patient record. While making predictions on new data, it is unknown on what characteristics or which part of the data may lead to better model performance. In addition, uncertainty can also be introduced by the model structure (such as the selection between linear model, tree-based model, or deep neural networks) and model parameters. Gal in his dissertation [11] concluded these two sources of uncertainty as (1) aleatoric uncertainty, which is caused by noisy data; and (2) epistemic uncertainty, which includes the uncertainty from both model structure and model parameters. The former one, aleatoric uncertainty can further be divided into homoscedastic uncertainty and heteroscedastic uncertainty. Homoscedastic uncertainty is captured independently of input data, while heteroscedastic uncertainty is instance-dependent.

For the healthcare domain where the risk and cost associated with a decision are high, it is not sufficient for a machine learning model to just deliver a prediction result. To gain the trust from end-users, the models should go beyond to reliably know when they are confident and when they are likely to make a mistake. Quantifying uncertainty is a way to represent such a confidence level. Knowing the importance of uncertainty and the fact that normal deep learning model is not capable of capturing uncertainty, researchers in the computer vision domain first estimated it with approaches such as Bayesian deep learning, which replaces deterministic model weights with prior distributions and use the learned posteriors to represent the uncertainty. A representative example of such a stream of research is Gal's dissertation research [11]. As for the healthcare domain, most of the existing literature focuses on capturing the uncertainty in the medical image processing and classification, which is essentially

the same with the problem in computer vision. To our best knowledge, there are only a few existing efforts studying uncertainty in deep learning models for the EHR data. They are 1) Heo *et al.*'s study in 2018 [12] on feature (variable) uncertainty in medical risk prediction tasks; 2) Dusenberry *et al.*'s research in 2019 [13] on Bayesian RNN with stochastic embedding to capture model uncertainty on the entire patient datasets and different patient subgroups; 3) Tan *et al.*'s work in 2019 [14] on attention mechanisms to accommodate varying time intervals in time series data, which they called "uncertainty". Their work collectively suggests the value of understanding deep learning uncertainty in the EHR data. However, there are still open research questions on computational approaches to capture both EHR aleatoric uncertainty and epistemic uncertainty efficiently and simultaneously, how to validate the results, uncertainty's relationship with model performance, uncertainty's effects on different patient demographic groups, as well as the implications that can assist the clinicians in allocating their attention and making decisions.

In this chapter, we proposed four neural network structures to capture both EHR aleatoric uncertainty and epistemic uncertainty in one Bayesian deep learning model. Specifically, the designed structure is from adapting and combining various Bayesian learning methods and sampling methods, to accommodate EHR data. The methods include the combination of Heteroscedastic Neural Networks (HNN) [1] with Deep Ensemble (DE) [65] and Dropout [15] respectively. The four models are applied into two published EHR datasets for a series of clinical prediction tasks (such as in-hospital mortality). Because there is no ground truth about either type of uncertainty, we verified the validity of both aleatoric and epistemic uncertainty through a series of experiments by intentionally introducing noise and randomness to the original dataset. We also examined the interaction effects between data uncertainty and model uncertainty on model performance, as well as the effects, broke down by patient demographic variables.

The major contributions of this chapter can be summarized in following aspects:

- The four deep learning neural network structures to capture both EHR data uncertainty and model uncertainty simultaneously in one model.
- A series of experiment design by intentionally introducing noise and randomness to the original dataset to characterize and validate the nature of our captured uncertainty, and their relationship with model performance.
- Patient subgroup analysis on the effects of uncertainty on different patient demographic groups in order to derive practical implications for doctors.

3.2 Methodology

In a normal deep learning classification model that does not estimate the uncertainty, the network takes all input variables into the trained black-box and only yield some probabilities at the output layer. To make the output layer return uncertainty together with the predicted probabilities, we present several methods for estimating heteroscedastic aleatoric uncertainty and epistemic uncertainty.

3.2.1 Estimating Heteroscedastic Aleatoric Uncertainty

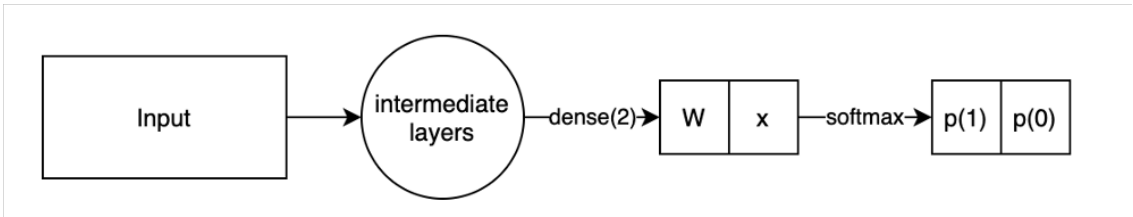

Figure 3.1: Normal deep learning output layer for categorical (binary) prediction.

To capture heteroscedastic aleatoric uncertainty in a classification model with EHR, we need to estimate the observation noise σ. In contrast to the homoscedastic uncertainty which assumes constant σ, the heteroscedastic uncertainty assumes that σ is input-dependent [66]. In a normal deep learning model (as shown in Figure 3.1), the

network output $\mathbf{x}$ is passed into a dense layer with weight $\mathbf{W}$ and a Softmax function to predict the probability vector $\hat{p}$:

$$\hat{p} = Softmax(\mathbf{W}\mathbf{x})$$

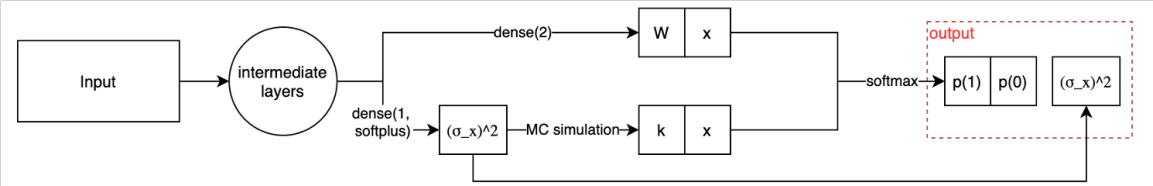

Figure 3.2: BNN output layer for categorical (binary) prediction.

According to the Heteroscedastic Neural Networks (HNN) proposed by Kendall and Gal in their study in 2017 [1], we could add a noise term k to the weight $\mathbf{W}$ and place a Gaussian distribution over the $k\mathbf{x}$. As shown in Figure 3.2, the data uncertainty is the extra term in the model output (red dotted box). It was represented by the variance σ_x^2 of the Gaussian distribution and calculated from $\mathbf{x}$ by another dense layer:

$$\hat{\mathbf{x}} = (\mathbf{W} + \mathbf{k})\mathbf{x} = \mathbf{W}\mathbf{x} + \mathbf{k}\mathbf{x},$$
$$k\mathbf{x} \sim \mathbf{N}(\mathbf{0}, \sigma_{\mathbf{x}}^{\mathbf{2}}\mathbf{I})$$

where $\mathbf{I}$ stands for an identity matrix. Then, we predict the probability $\hat{p}$ using the "corrupted" output $\hat{\mathbf{x}}$, as in the study of Kendall and Gal [1]:

$$\hat{p} = Softmax(\hat{\mathbf{x}})$$

Since there is no analytical solution to integrate out the Gaussian distribution of introduced error $\mathbf{kx}$ for a normally used cross entropy loss function for classification tasks, Monte Carlo (MC) simulation is used in Kendall and Gal's study in 2017 [1] to approximate the objective. We will briefly introduce it here. The simulation is

performed after the calculation of the network output $\mathbf{x}$, so it only increases a fraction of the model computing time. Assume that T times Monte Carlo is simulated, the loss function for this part is:

$$\hat{x}_t = \mathbf{W}\mathbf{x} + \sigma_{\mathbf{x}}\epsilon_{\mathbf{t}}, \epsilon_{\mathbf{t}} \sim \mathbf{N}(\mathbf{0}, \mathbf{I})$$

$$L_{Bayes} = \log \frac{1}{T} \sum_{t=1}^{T} \exp\left(\hat{x}_t - \log \sum_{c} \exp \hat{x}_{t,c}\right)$$

where t represents one MC simulation, c is every element in $\hat{x}_t$, and L_{Bayes} stands for Bayesian categorical cross entropy. The BNN model will be optimized towards the weighted average of a regularizer on estimated σ_x, and the categorical cross entropy as commonly used in normal deep learning classification models. Note that only the classification task is supervised. The aleatoric uncertainty as the variance term σ_x^2 is learned as we minimize the loss function.

The HNN only performs Bayesian learning at the output layer, so it can be readily applied to any built models with appropriate changes on the input layer, output layer, and loss function. In this chapter, we will experiment with the idea of placing HNN [1] on tops of some EHR friendly deep learning models, such as CNN and GRU.

Except for the HNN models, it was claimed that for the binary tasks, the single probability predicted by a normal deep learning model can be viewed as the uncertainty from data [67, 13]. While is true that the probability is dependent on the input, it reduces the model output from three to two and loses that advantages brought by the HNN.

3.2.2 Estimating Epistemic Uncertainty

We used two different approaches to capture epistemic uncertainty, as described in the following subsections.

3.2.2.1 Deep Ensemble

Deep Ensemble (DE) [65] is a simple ensemble method that estimates the uncertainty from the model without requiring the model to be "Bayesian". We need to train an ensemble of neural networks using the same network structure, hyper-parameters, and input data. These networks differ from each other only by the randomness in the model weight initialization. Consider that n models are trained, for one patient in the test dataset, the predicted probabilities of this patient $\{\lambda_1, \lambda_2, ...\lambda_n\}$ can be viewed as a distribution $p(\lambda|x, w)$, where x represents the input data and w represents the model weights. The variance of this distribution represents the epistemic uncertainty:

$$un_{epi} = var(\lambda_1, \lambda_2, ...\lambda_n)$$

3.2.2.2 Dropout

Dropout [68] is a regularization method that has been commonly used in deep neural networks to reduce the problem of over-fitting. It was accomplished by randomly dropping a certain portion of nodes/units in a neural network layer. The dropout layers in the deep learning model are usually turned on while training for better regularization and then turned off while making predictions on the testing phase. As theoretically proved by Gal and Ghahramani [15], dropout can be used as the approximation of Bayesian inference in deep learning models. The epistemic uncertainty can be estimated by (1) enabling the dropout layers in the trained model, (2) making multiple times of predictions on the test data to form a distribution, and (3) calculating the epistemic uncertainty from the distribution. Consider n times of predictions are made for each patient, the probabilities $\{\lambda_1, \lambda_2, ...\lambda_n\}$ can be viewed as a distribution $p(\lambda|x, w)$, where x represents the input and w represents the model weights. The

epistemic uncertainty is calculated as the prediction entropy:

$$un_{epi} = -\sum_{i=1}^{n} log(\lambda_i) * \lambda_i$$

Having the approaches to calculating data uncertainty and model uncertainty, we will be able to apply the approaches to two published EHR datasets to test their validity.

3.3 Experiments

In this section, we will talk about the two EHR datasets for our experiments as well as the experiment set-up, such as baseline models, methods to verify uncertainty, etc.

3.3.1 Datasets and Our Prediction Tasks

We use two datasets and five clinical tasks for the experiments in this chapter.

- **MIMIC-III dataset and the task of predicting mortality.** We follow the data preprocessing and feature extraction steps described in [69] to prepare the input data for the MIMIC-III mortality task. The dataset contains a total of 21,139 records, among which 2,797 are positive cases. The observation window is the first 48 hours after admission. Seventeen features are collected, including heart rate, temperature, weight, pH, Glascow coma scales, and patient monitor records, such as systolic blood pressure and respiratory rate. The categorical features are one-hot encoded and others are normalized. Due to the sparsity of the EHR data, there is a large number of missing values. Each value is imputed and followed by an indicator specifying its status (true value or missing value). After encoding, normalizing, and imputation, each patient's record is in the shape of 48 hours with 76 generated features.

- **PhysioNet 2012 Data Challenge.** The 2012 PhysioNet Challenge dataset

[19] stores time series data from 12,000 ICU records, each contains 37 variables such as heart rate, serum glucose, and Glasgow coma score, in 48 hours (155 time steps) after hospital admission. We used the training set A (4000 records) in the experiments. We conducted four binary predictive tasks: in-hospital mortality, length-of-stay less than 3, having a cardiac condition, and recovering from surgery.

3.3.2 Baselines

Since our approaches to capturing data uncertainty and model uncertainty need baseline deep learning models to work on. Below is a list of baselines in this chapter. They represent state-of-the-art deep learning models used in the health context.

- **CNN**: 1-dimensional temporal CNN that was widely used in the medical domain (examples are [25, 23, 70]). The convolutions are capable of capturing local temporal relations between medical features, hence outperforming conventional methods such as logistic regression and random forest.

- **GRU**: In addition to the RNN model with GRU layers [22], the pooling mechanisms [70, 35] are added to detect strong signals in the early stage of the time series data.

- **RETAIN**: The RNN model with both temporal and feature attentions proposed by [30].

- **UA-Attention**: The uncertainty-aware RNN model [12] that is based on RETAIN.

3.3.3 Experiment Settings

We will introduce the four proposed variants of deep learning models for capturing both types of uncertainty. They are:

- **HCNN-DE**: Based on 1-D CNN, HNN [1], and DE [65].
- **HCNN-DR**: 1-D CNN model that uses HNN and Dropout [15] to capture data uncertainty and model uncertainty, respectively.
- **HGRU-DE**: GRU model that combines HNN at the output layer and uses DE to quantify model uncertainty.
- **HGRU-DR**: HGRU model with Dropout layer turned on in the testing phase.

The structure and configurations of HNN are shown in Fig. 3.3. The model takes input (N, T, F), which corresponds to the number of cases, time duration, and the number of medical features respectively. The dotted box illustrates the two possible structures with HGRU or HCNN. One model will take only one possibility of the two. On the GRU side, the GRU layer was configured with 100 hidden units, then the output is concatenated with ts maximum, minimum, and average, forming the output of GRU in the shape of 400 units. On the CNN side, we used 64 filters with kernel size 3 for two 1-D convolutional layers, each of which is followed by a 'Batch Normalization" layer and Dropout layer. Following the dotted box structure is a Dense layer with 100 units/nodes with another Dropout layer. For all Dropout layers, the dropout rate is set to 0.5. For the output, the dense layers with 'softplus' and 'softmax' are used to generate the aleatoric uncertainty and the predicted probability respectively. The aleatoric uncertainty is learned by optimizing the Bayesian categorical cross entropy (aleatoric uncertainty loss) and the predicted probability is learned by optimizing the normal categorical cross entropy. We use 'Adam' as the optimizer with the learning rate and decay both set to be 0.001. The number of MC simulation for updating the Bayesian categorical cross entropy is set to 100. The weights for adding up the Bayesian categorical cross entropy and the categorical cross entropy are set to be 0.5 and 1 respectively. When training DE (Deep Ensemble) models for epistemic uncertainty, 'training' option is set to be 'False' for the Dropout

layers; when training Dropout models for epistemic uncertainty, 'training' is set to be 'True'. For the calculations of epistemic uncertainty, we used 100 outputs to form the distribution, meaning that 100 models trained for the Deep Ensemble approach and 100 predictions made by the Dropout model.

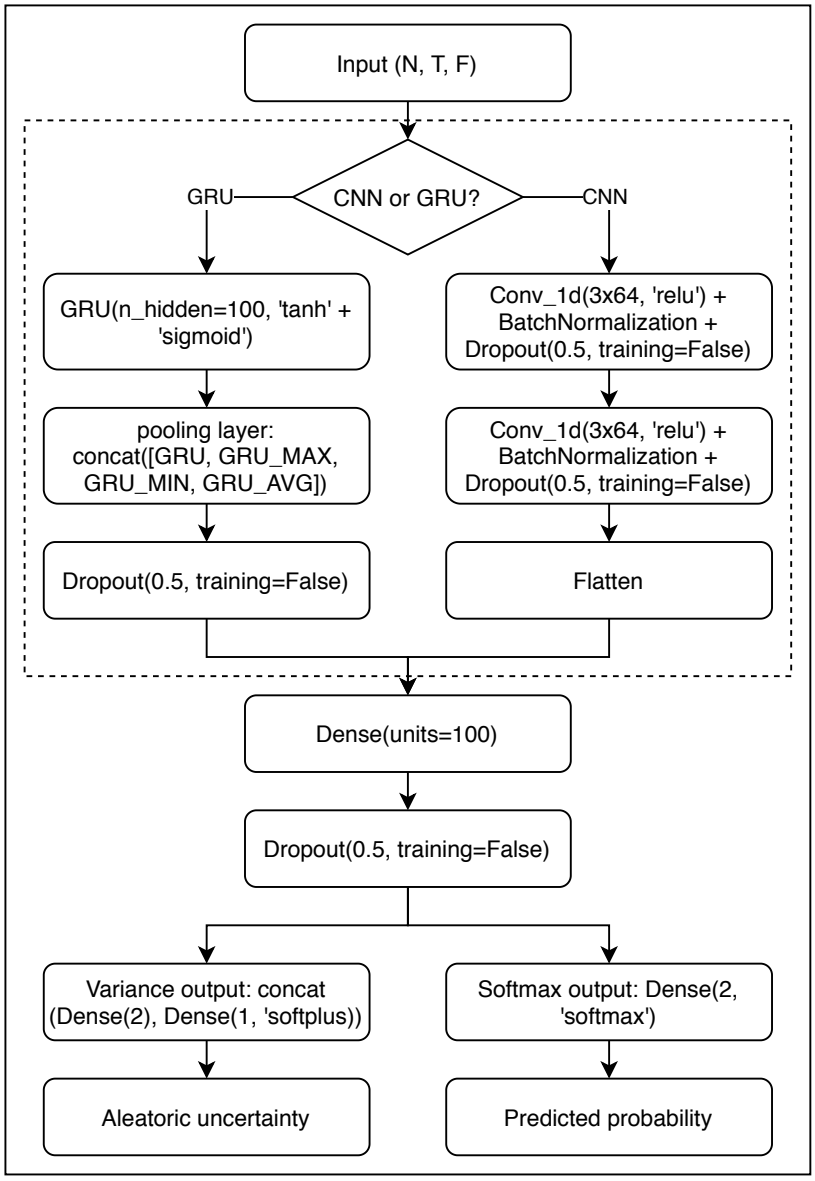

Figure 3.3: The network structure and configurations of proposed HNN models.

3.3.4 Uncertainty Verification

Through a series of experiments, we will test the captured uncertainty (both data and model) against several commonly accepted facts about uncertainty, in order to verify that the estimated uncertainty does capture what we intended to capture for both data and model. Below we will introduce the set-up of the experiments.

Aleatoric uncertainty verification. It is by definition that when the noises from data increase, aleatoric uncertainty will increase. Therefore we will test whether as the data gets noisier, our model will output higher estimated aleatoric uncertainty. Specifically, we randomly selected some portion of combinations of (time, feature) from the patient records in the original data, and removed them to generate new datasets. As the result, we created 5 new datasets with 0% (all data removed), 25%, 50%, 75%, and 100% (all data retained). We then evaluated how the estimated aleatoric uncertainty change, and whether the change was consistent with the definition of aleatoric uncertainty as expected. In addition, we also investigated how model performance will change as the result.

Epistemic uncertainty verification. Since the epistemic uncertainty can be reduced or explained away with sufficient training data [1], we are interested in this test: will adding more training data of specific patient group decrease the model uncertainty of the patients in that group? If so, will the change affect the model performance? We will conduct this experiment using the PhysioNet dataset and the mortality prediction task. In the dataset, each patient was labeled with one of the four ICU types at admission: Coronary Care Unit, Cardiac Surgery Recovery Unit, Medical ICU, and Surgical ICU. We will test against this known fact (epistemic uncertainty will be reduced with sufficient training data) by reducing the number of patients of a certain ICU type in the training set, and then evaluate how the estimated epistemic uncertainty as well as model performance will change. Similarly, five new datasets were created: 0% (no train data in this type), 25%, 50%, 75%, and 100% (all training

data in this type retained) respectively.

Interaction effects of aleatoric uncertainty and epistemic uncertainty on the model performance. With the capability of estimating both types of uncertainty in one model, we also want to investigate how do the two types of uncertainty interact with the model performance. Will the population with both low aleatoric uncertainty and low epistemic uncertainty always perform the best? What if the aleatoric uncertainty is high while the epistemic uncertainty is low? We divided the dataset into four parts by the medians of aleatoric uncertainty and epistemic uncertainty, then evaluated the performance of each part to learn the interaction effects.

3.4 Results and Discussion

3.4.1 Model Performance

We used the area under the receiver operator characteristic curves (AUC) and 5-fold cross-validation to report the model performance. Table 3.1 shows the AUC scores with 95% confidence interval of all the five tasks. From these results, we can confidently claim that capturing the uncertainty does not compromise the model performance: The AUC scores of proposed models are comparable, if not better than, to the baseline models in all five tasks. The HGRU models achieve the highest average AUC scores and improve the performance by 1%-4% compared with the baselines. This performance improvement implies that the process of learning aleatoric uncertainty is in fact helping mitigate the negative effect of noisy data on model performance. The process of estimating epistemic uncertainty, on the other hand, does not have this mitigating effect because the process does not involve any model learning process.

It is always the case that the GRU models always outperform the CNN models. This could be due to the fact that GRU captures more long-term information and CNN focus more on local (short-term) context. For the PhysioNet dataset, the performance differences between CNN and GRU are larger due to the varying time intervals

of the patient records. Although we have 155 time points for all the patients, some of them only have shorter length of records with many missing values. Under this situation of data irregularity and sparsity, GRUs' long-term memory and tolerance on missing value have advantages compared to CNN. Therefore, we will focus on the HGRU models in the rest of the result analysis to demonstrate our idea. Specifically, we will only use the HGRU-DR model for the demonstration purpose.

Table 3.1: Model performance (AUC scores) comparison for 5 binary tasks

	MIMIC	**PhysioNet**			
	Mortality	**Mortality**	**Stay < 3**	**Cardiac**	**Recovery**
Baseline Models					
CNN	0.8439 ± 0.0149	0.7518 ± 0.0116	0.8389 ± 0.0091	0.9183 ± 0.0218	0.8587 ± 0.0184
GRU	0.8589 ± 0.0163	0.7881 ± 0.0193	0.8615 ± 0.0077	0.9602 ± 0.0173	0.9032 ± 0.0142
RETAIN	0.8242 ± 0.0178	0.7652 ± 0.0203	0.8515 ± 0.0185	0.9485 ± 0.0138	0.8830 ± 0.0095
UA-RETAIN	0.8296 ± 0.0263	0.7737 ± 0.0234	0.8595 ± 0.0163	0.9574 ± 0.0274	0.8895 ± 0.0153
Our Proposed Models					
HCNN-DE	0.8502 ± 0.0128	0.7532 ± 0.0196	0.8392 ± 0.0116	0.9053 ± 0.0186	0.8630 ± 0.0263
HCNN-DR	0.8483 ± 0.0187	0.7549 ± 0.0209	0.8280 ± 0.0232	0.8975 ± 0.0198	0.8554 ± 0.0210
HGRU-DE	$\mathbf{0.8659 \pm 0.0172}$	0.7973 ± 0.0183	$\mathbf{0.8646 \pm 0.0219}$	$\mathbf{0.9629 \pm 0.0120}$	$\mathbf{0.9102 \pm 0.0217}$
HGRU-DR	0.8618 ± 0.0216	$\mathbf{0.7989 \pm 0.0190}$	0.8565 ± 0.0241	0.9580 ± 0.0085	0.9002 ± 0.0297

The reported numbers are the mean AUC and standard errors for 95% confidence interval over 5-fold cross validation. The bold numbers indicate the best performance in that (column) group.

3.4.2 Comparing the Uncertainties and Model Performance

Next we will explore how aleatoric uncertainty and epistemic uncertainty affect the model performance. We divided the test dataset into two segments by the median of aleatoric uncertainty and epistemic uncertainty respectively, and compare the model performance for each segment. As shown in Table 3.2, the segments with lower aleatoric uncertainty have seen a better performance (higher AUC scores) by 0.06 - 0.12 than the segments with higher aleatoric uncertainty. These results confirmed that high data uncertainty does harm the model performance from the population level, although it may not always the case at the individual level.

We also compare the performance between segments with high and low epistemic uncertainty. The difference is not as large as that between the two aleatoric uncertainty groups. For the two mortality prediction tasks, the low-epistemic-uncertainty group performs significantly better, while for the other tasks the differences are gener-

ally very small. It is possible that with the model being sufficiently trained on similar cases, the effect of epistemic uncertainty on the performance is largely weakened.

Table 3.2: Model performance comparison between uncertainty groups separated by medians

	MIMIC	**PhysioNet**			
	Mortality	**Mortality**	**Stay < 3**	**Cardiac**	**Recovery**
Aleatoric Uncertainty - High	0.8017 ± 0.0216	0.7504 ± 0.0018	0.8104 ± 0.0289	0.9204 ± 0.0183	0.8304 ± 0.0238
Aleatoric Uncertainty - Low	0.9030 ± 0.0129	0.8335 ± 0.0072	0.9352 ± 0.0352	0.9835 ± 0.0215	0.9559 ± 0.0294
Epistemic Uncertainty - High	0.7503 ± 0.0284	0.7045 ± 0.0064	0.7548 ± 0.0254	0.8365 ± 0.0194	0.7943 ± 0.0124
Epistemic Uncertainty - Low	0.7993 ± 0.0163	0.7261 ± 0.0082	0.7792 ± 0.0265	0.8461 ± 0.0122	0.8024 ± 0.0182

The reported numbers are the mean AUC and standard errors for 95% confidence interval over 5-fold cross validation.

3.4.3 Verification of Aleatoric Uncertainty

To verify whether the estimated aleatoric uncertainty does capture the noise from the data, we manually created several datasets by introducing different levels of missing values. As mentioned in Section 3.3.4, removing some portions of the non-missing values is practically equivalent to increasing data noise level. Based on this idea, we created 5 datasets with 0% (all data removed), 25%, 50%, 75%, and 100% (all data retained) to evaluate the change of the estimated aleatoric uncertainty and corresponding AUC scores. We conducted this experiment on the two datasets for the same mortality prediction task. The results are displayed in Fig. 3.4. For both datasets, the aleatoric uncertainty (green bars) captured by HGRU-DR model decreases as the percentage of retained data increases, which is as expected by the definition of aleatoric uncertainty. In addition, as the aleatoric uncertainty is decreasing, the AUC scores (blue bars) are increasing, which confirms the relationship between aleatoric uncertainty and model performance documented in Table 3.2. Therefore, we believe that HGRU-DR is able to capture aleatoric uncertainty as it intended to.

3.4.4 Verification of Epistemic Uncertainty

Similarly, we want to verify whether the estimated epistemic uncertainty is able to reflect the model noises. We used the PhysioNet dataset to demonstrate our idea to conduct this verification. In the PhysioNet dataset, patients are grouped by the

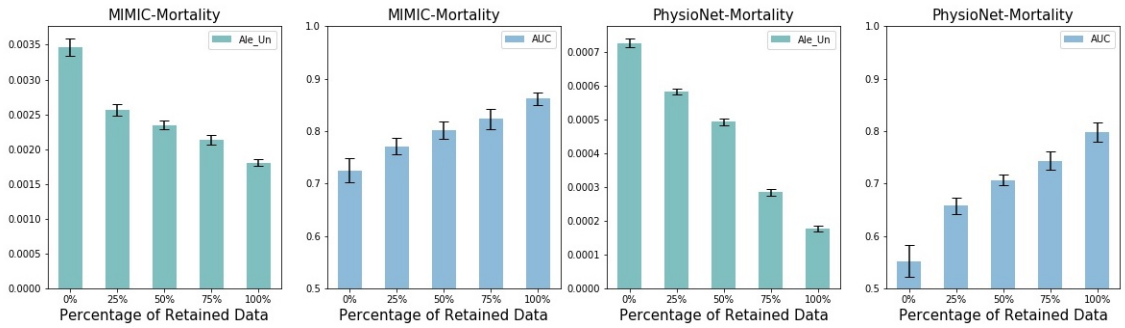

Figure 3.4: Aleatoric uncertainty verification using MIMIC-Mortality and PhysioNet-Mortality datasets

ICU type at the time of admission. As mentioned in Section IV.D, the ICU type may take one of the four values: Coronary Care Unit, Cardiac Surgery Recovery Unit, Medical ICU, and Surgical ICU. If the amount of training samples of Coronary-ICU patients is decreased, the model will see fewer patients of this type, and its performance is supposed to be damaged [1]. To test this, we created five new datasets, each with 0% (no train data), 25%, 50%, 75%, and 100% (all data retained for training), and evaluated the change of the estimated epistemic uncertainty and the corresponding AUCs. For demonstration purposes, we displayed the results for only two ICU types out of the four: the Coronary ICU and the Cardiac Surgery Recovery ICU as in Fig. 3.5. As illustrated by the green bars, our model successfully captures the decreasing trend of estimated epistemic uncertainty as the amount of training samples increases. The AUC scores (blue bars) increase as expected too.

We also notice that the AUC of “Coronary ICU" group is only around 0.74 at its highest, compared with the highest AUC of 0.86 for the “Cardiac Surgery Recovery ICU" group. As shown in Fig. 3.5, the epistemic uncertainties of these two groups are roughly at the same level. We further compared the average aleatoric uncertainty of these two patient groups: 0.0015 and 0.0005 respectively. This finding indicates that the differences in the performance could possibly be caused by either aleatoric uncertainty or epistemic uncertainty. That is the grounding that we investigated both

data and model uncertainty simultaneously.

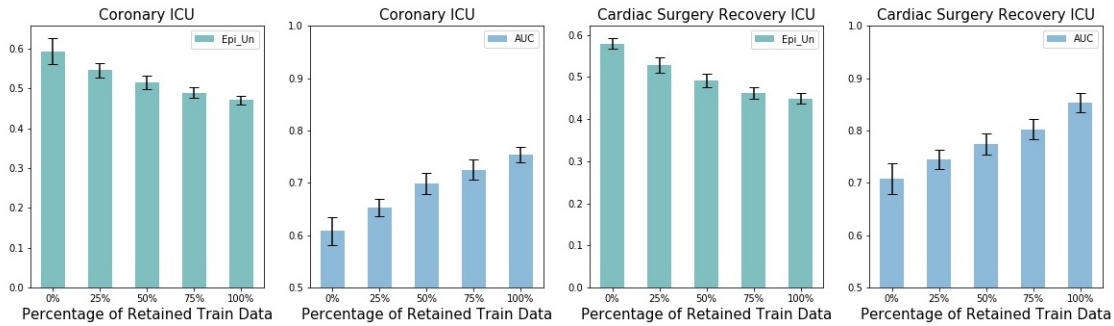

Figure 3.5: Epistemic uncertainty verification using “Coronary ICU" patients and “Cardiac Surgery Recovery ICU" patients

3.4.5 Interaction Effects of Aleatoric Uncertainty and Epistemic Uncertainty on the Model Performance

With aleatoric uncertainty and epistemic uncertainty both confirmed to be negatively correlated with the model performance at the population level, we are interested in their interaction effects. The test data is divided into four groups by the medians of data and model uncertainty to calculate the AUC scores. As reported in Table 3.3, AUC is highest for the group with both low data and low model uncertainty, indicating that both have a negative impact on the performance. However, the AUC is not the lowest in the group that both uncertainties are high. When the epistemic uncertainty is high, the effect of aleatoric uncertainty on the model performance is not deterministic: higher aleatoric uncertainty does not necessarily lead to lower performance. Imagine when a model makes predictions on cases that it was never trained on, data noises are not as important as they are for the familiar cases. An example from the image classification can explain the situation: when using a model trained on images of dogs to make prediction on a photo of a cat, poor image quality or missing pixels does not matter much anymore.

Table 3.3: AUC scores of four different patient groups, divided at medians of both aleatoric uncertainty and epistemic uncertainty

MIMIC-Mortality		Epistemic Uncertainty	
		High	Low
Aleatoric Uncertainty	High	0.6741 ± 0.0123	0.7063 ± 0.0203
	Low	0.6716 ± 0.0164	0.7331 ± 0.0168
PhysioNet-Mortality		Epistemic Uncertainty	
		High	Low
Aleatoric Uncertainty	High	0.7561 ± 0.0231	0.7800 ± 0.0205
	Low	0.6876 ± 0.0193	0.8009 ± 0.0210

3.4.6 Patient Subgroup Analysis

We will also analyze how uncertainties and model performance are impacted across different patient subgroups. We split. patients into subgroups by demographic variables: gender (female vs. male) and age (under 65 vs. above 65). We re-run the HGRU-DR model on each subgroup and then examined the effects of uncertainties and model performance.

Starting with the gender variable, we only used randomly 50% of the training data for female and male patients respectively, and also randomly removed 50% of the features in test data. This setting corresponds to the “50%" bin in Figs. 3.4 and 3.5. This way made room to improve the aleatoric uncertainty and epistemic uncertainty later on. As presented in Table 3.4, the female group has a lower AUC than the male group. By comparing the uncertainties, we found that the two groups' epistemic uncertainties are roughly the same, while the aleatoric uncertainty's difference is large. This observation suggests that aleatoric uncertainty plays a role in the female group's lower performance. And more importantly, the observation suggests a mitigation action: we could improve female group's model performance by improving its aleatoric uncertainty through collect more feature values for this group. To perform this action, we added the removed features back to the female group and keep the other settings

unchanged. Now in the lower half of Table 3.4, the AUC of the female group is increased by 0.06 and even outperforms the male group (half data though). This example demonstrates the viability and benefits of inspecting uncertainties for certain patient subgroups, identifying possible causes, then taking the actions to improve the performance.

Table 3.4: Improving the estimated aleatoric uncertainty to improve the model performance for the female group

Gender	**Female**	**Male**
AUC	0.7301 ± 0.0164	0.7683 ± 0.0144
Epistemic Uncertainty	0.4823 ± 0.0272	0.4992 ± 0.0289
Aleatoric Uncertainty	0.2242 ± 0.0190	0.1223 ± 0.0284
	Increasing Data Points	**-**
AUC	0.7909 ± 0.0183	-
Epistemic Uncertainty	0.4732 ± 0.0122	-
Aleatoric Uncertainty	0.0942 ± 0.0110	-

Likewise, we divided the patients by age. The result is presented in Table 3.5. The HGRU-DR model performs significantly better in the non-senior group (under 65) than in the senior group (65 or older). The average aleatoric uncertainty level of these two groups are roughly the same, both around 0.37. However, the obvious difference of the epistemic uncertainty (0.3640 versus 0.4803) suggests that epistemic uncertainty plays a role in the lower model performance for the senior patients. In this case, the model hasn't seen enough senior cases, and therefore collecting more data points for each senior patient does not help. This observation suggests our corresponding action: we could improve the model uncertainty for the senior group by adding more training samples in this senior group. Therefore we added the 50% samples back to the training dataset and saw the AUC score increase by 0.03, while the epistemic uncertainty decrease from 0.4803 to 0.2983.

Table 3.5: Improving the estimated model uncertainty to improve the model performance for the senior patients

Age	**Under 65**	**65 or older**
AUC	0.8060 ± 0.0198	0.7292 ± 0.0156
Epistemic Uncertainty	0.3640 ± 0.0238	0.4803 ± 0.0442
Aleatoric Uncertainty	0.3709 ± 0.0187	0.3773 ± 0.0215
	-	**Increasing Training Samples**
AUC	-	0.7573 ± 0.0103
Epistemic Uncertainty	-	0.2983 ± 0.0247
Aleatoric Uncertainty	-	0.3732 ± 0.0154

3.5 Conclusion and Future Works

In this chapter, we proposed and implemented four neural network structures to capture both EHR aleatoric uncertainty and epistemic uncertainty in one Bayesian deep learning model. The four models were applied to two published EHR datasets with a series of clinical prediction tasks (such as in-hospital mortality). By manually introducing varying levels of noises and randomness into datasets, or removing a varying number of data points and training samples, we verified the validity of our computational approach to both aleatoric and epistemic uncertainty. We also found a negative correlation between both types of uncertainty and model performance. There also existed interesting interaction effects between both uncertainties on model performance. At last, we conducted patient subgroup analysis to find actionable treatments for those subgroups for which the model tends to under-performs. The results have enriched the model output from the deep learning models, which will help doctors and health professionals allocate their attention to those in need, as well as offer action suggestions to improve model confidence in individual patients. In the next chapter, we will continue to develop deep learning approaches to estimate the feature-level uncertainties and perform targeted post-hoc analysis on those features with larger estimated feature uncertainty. For better evaluation of the uncertainty

estimations, we will also use a user study to understand the real end-users' feedback.

CHAPTER 4: MODELING FEATURE-LEVEL UNCERTAINTY OF EHR-BASED DEEP LEARNING MODELS

4.1 Background

In Chapter 3, we proposed HNN frameworks for the estimations of patient-level uncertainty, hence providing extra confidence representations and distribution estimations to increase the clinicians' trust. However, it is still difficult for the clinicians to explain the uncertainty score to the patients. In addition, they need to understand why a model is not confident and what could be the possible causes - either a certain time period or the happening of certain medical events. Therefore, the explanations on uncertainty, or feature-level uncertainty, is important.

In the literature of EHR-based uncertainty estimation, we only found one work from Heo *et al.* [12], in which they introduced the notion of input-dependent uncertainty to an attention mechanism, to generate an attention weight for each feature with different degrees of noise, and learn larger variance on instances the model is uncertain about. Their study was the first to investigate the feature-level (variable-level) uncertainty, which has great potential for richer interpretations of deep learning model results to assist clinicians. Compared to this dissertation study, their work focused on the feature-level uncertainty and attention weights in order to improve model prediction performance, while we tried to utilize the feature uncertainty to improve model trustworthiness and explainability.

Similar to the patient-level uncertainty, the evaluation and validation of the feature-level uncertainty estimations are difficult. For the computer vision tasks such as semantic segmentation, labeled data is easy to obtain and each feature (pixel) can get a ground truth (e.g. a pixel belongs to side-walk, drive-way, or traffic lights).

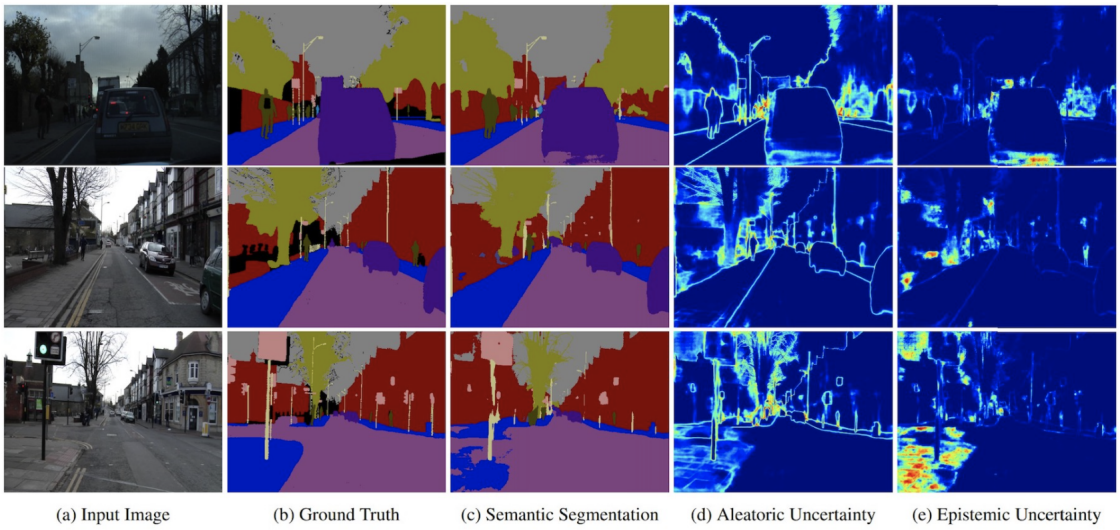

Figure 4.1: Example of pixel-level uncertainty validation in computer vision domain. Source: Kendall and Gal, 2017 [1].

As presented in [1], these data are good resources for validating the uncertainty estimations. In Figure 4.1, the input image (a) was used to make prediction of semantic segmentation. With the help of ground truth (b), we can see how the model performed in the outputs (c). We can also visually evaluate how could the data and model uncertainty highlight the uncertain pixels. Without such labeled ground truth and visualizations in EHR-based tasks, we propose to use uncertainty verification experiments to make sure that the estimated feature uncertainty can correctly capture the noises in the features.

In this chapter, we propose two Variational Neural Networks (VNN) frameworks, namely Variational Convolutional Neural Networks (VCNN) and Variational Gated Recurrent Unit (VGRU). The models were applied to estimate the temporal feature uncertainty - time window uncertainty with VCNN and time point uncertainty with VGRU. Similar to the patient-level uncertainty estimations, we design and conduct experiments to verify that the uncertainty estimations could correctly capture the manually-added noises. We also present a patient's record and uncertainty estimations (both patient-level and feature-level) identified as 'high-uncertainty' to a

clinician for verification.

4.2 Methodology

In HNN models, we placed the prior distributions on the output layer and learned the posteriors in the training process to estimate the data uncertainty. We estimated the model uncertainty by approximating the BNN models with DE or DN models that introduced randomness to the output. Inspired by and based on these methods, we proposed VNN models, which applied Bayesian learning to the first neural network layers so that the temporal feature uncertainty can be estimated by inferring the posteriors. In this section, I will describe two variational layers for VCNN and VGRU, respectively.

- **Variational Convolutional Layer.** The first 1-d convolutional layer of the network was directly connected to the input data and used the temporal convolutions/kernels to extract meaningful local information. We made this layer 'variational' by replacing the weights of these convolutions with prior Gaussian distributions, so that the output of this layer can be used to estimate the uncertainty of corresponding time span. As shown in Figure 4.2, the variational convolution distilled local information from the dark grey area in the input and stored the information in the feature maps (also in the format of a vector of distributions). Similar to a normal convolutional layer, the length of the feature maps depend on the length of the input and the size of the convolutions/kernels. Then with the generated feature maps, we used sampling to get (1) the means of the distributions that would be passed to further layers of the network to make predictions and (2) the max-pooled variances of the distributions that would be used as the uncertainty estimations of corresponding temporal features (time spans). The size of these temporal features were decided by the size of the convolutions, so they had to be clinically meaningful to the doctors (e.g. 3 days, 7 days, 14 days, etc.).

- **Variational Recurrent Unit.** The construction of VGRU layer was similar to that of VCNN's. The weights and biases in the recurrent unit were all replaced with Gaussian priors. Therefore, the output of the first GRU layer, the intermediate temporal states, were represented by a group of learned posteriors (Figure 4.3). These distributions can be used to sample a set of weights for the further layers of the network: using the last intermediate state directly for the final prediction (as the red vector in Figure 4.3), or being stacked to a normal GRU layer, or even being attached to another VGRU layer. The variances of the distributions would be treated as the corresponding temporal feature (time point) uncertainty estimations. The captured uncertainty can be represented either by a max-pooled vector (like in the VCNN) or a heat map.

4.3 Experiments

In this chapter, we describe the experiment set-up, including baseline models and methods for uncertainty verification. The EHR datasets and predictive tasks that we used are the same as the ones in Chapter 3.

4.3.1 Baselines

For the comparison of model performance, we used the HNN models with Dropout or Deep Ensemble from the last chapter.

- **HCNN-DE**: Based on 1-D CNN, HNN [1], and DE.
- **HCNN-DR**: 1-D CNN model that uses HNN and Dropout [15] to capture data uncertainty and model uncertainty, respectively.
- **HGRU-DE**: GRU model that combines HNN at the output layer and uses DE to quantify model uncertainty.
- **HGRU-DR**: HGRU model with Dropout layer turned on in the testing phase.

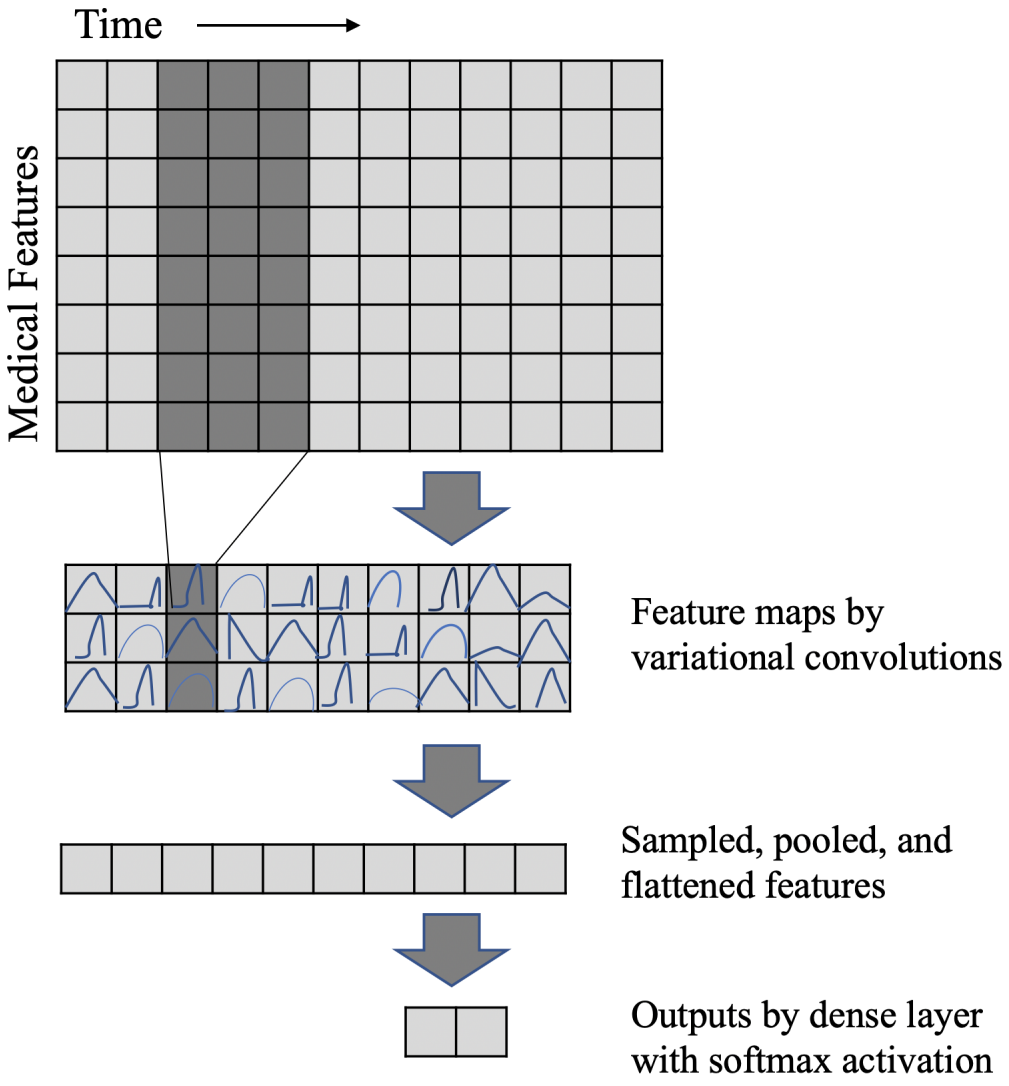

Figure 4.2: Variational Convolutional Layer for estimating the temporal feature uncertainty.

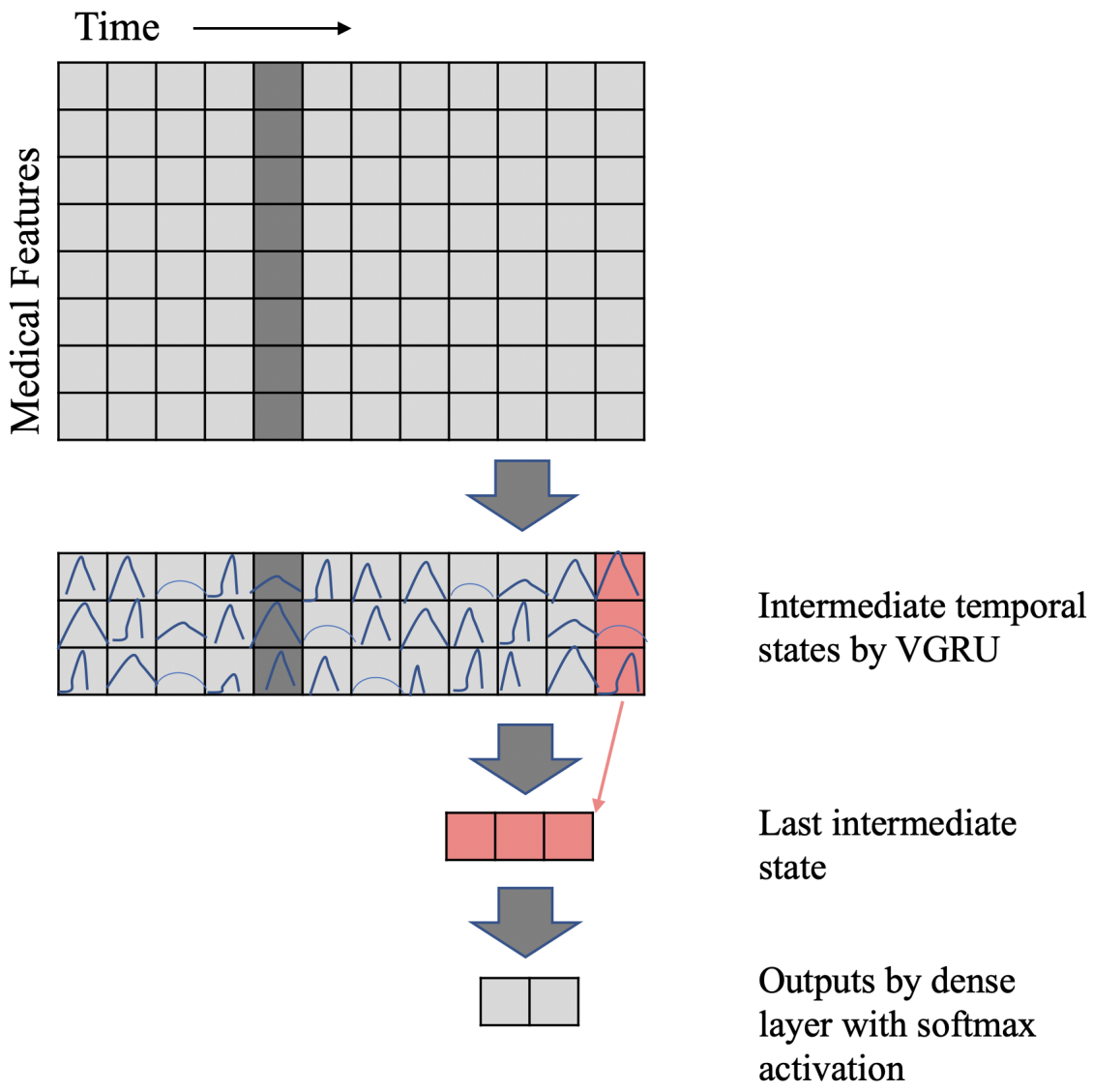

Figure 4.3: Variational Recurrent Unit for estimating the temporal feature uncertainty.

4.3.2 Experiment Settings

Based on the baseline models, we proposed four variants of VNN models for capturing temporal feature uncertainty.

- **VCNN-DE**: Based on HCNN-DE, the first 1-d convolutional layer was replaced by a 1-d variational convolutional layer.
- **VCNN-DR**: 1-D VCNN model that uses Dropout to capture model uncertainty.
- **VGRU-DE**: VGRU model that combines variational recurrent unit at the first layer, HNN at the output layer, and Deep Ensemble.
- **VGRU-DR**: VGRU model with Dropout layer turned on in the testing phase.

The structures and configurations of the VNN models are shown in Figure 4.4 and Figure 4.5. The input shape (N, T, F) corresponds to the number of cases, time duration, and the number of medical features, respectively. For VCNN, the first component was a 1-d variational convolutional layer with 64 kernels of size 3. 100 MC simulations were performed to generate distributions for the weights and biases, hence creating distributions for the feature maps. The means of these feature maps were passed down the network for Batch Normalization, Dropout, and further network components. The rest of the network had same settings with the HCNN models. The variances of the feature maps were used as the estimations of the temporal feature uncertainty with a max-pooling operation (taking the maximum of each temporal feature). For the VGRU model, the first layer was a variational GRU layer with 100 hidden units. 100 MC simulations were also performed to generate distributions for the GRU weights and biases, then formatting distributions for the GRU intermediate hidden states. Similarly, the means of the these distributions were passed down to the rest of the components, which are identical to the ones in HGRU. The variances were

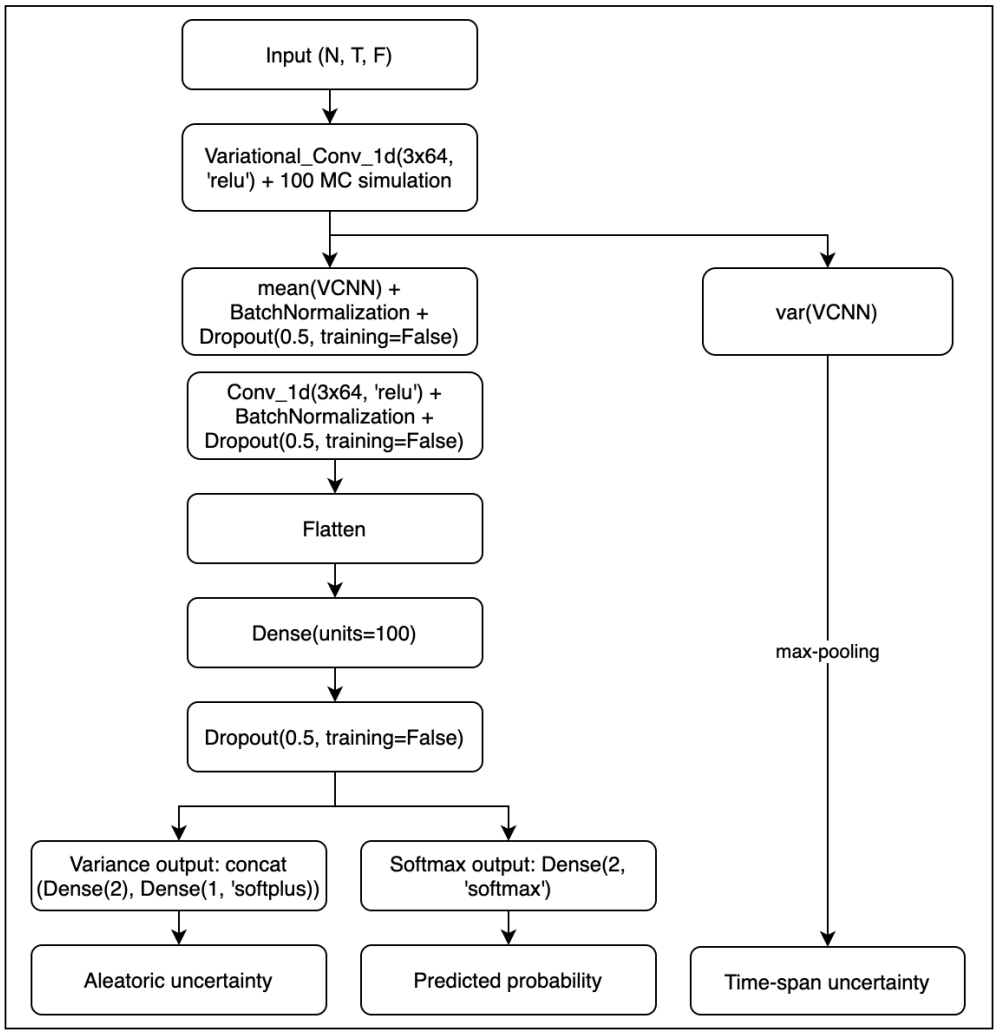

Figure 4.4: The network structure and configurations of the proposed VCNN model.

treated as the corresponding time point uncertainty after running through a max-pooling operation. The optimizer was Adam with learning rate of 0.001 and decay rate of 0.0001. When training DE (Deep Ensemble) models for epistemic uncertainty, 'training' option was set to 'False' for the Dropout layers; when training Dropout models for epistemic uncertainty, 'training' was set to 'True'. For the calculations of epistemic uncertainty, we used 100 outputs to form the distribution, meaning that 100 models trained for the Deep Ensemble approach and 100 predictions made by the Dropout model.

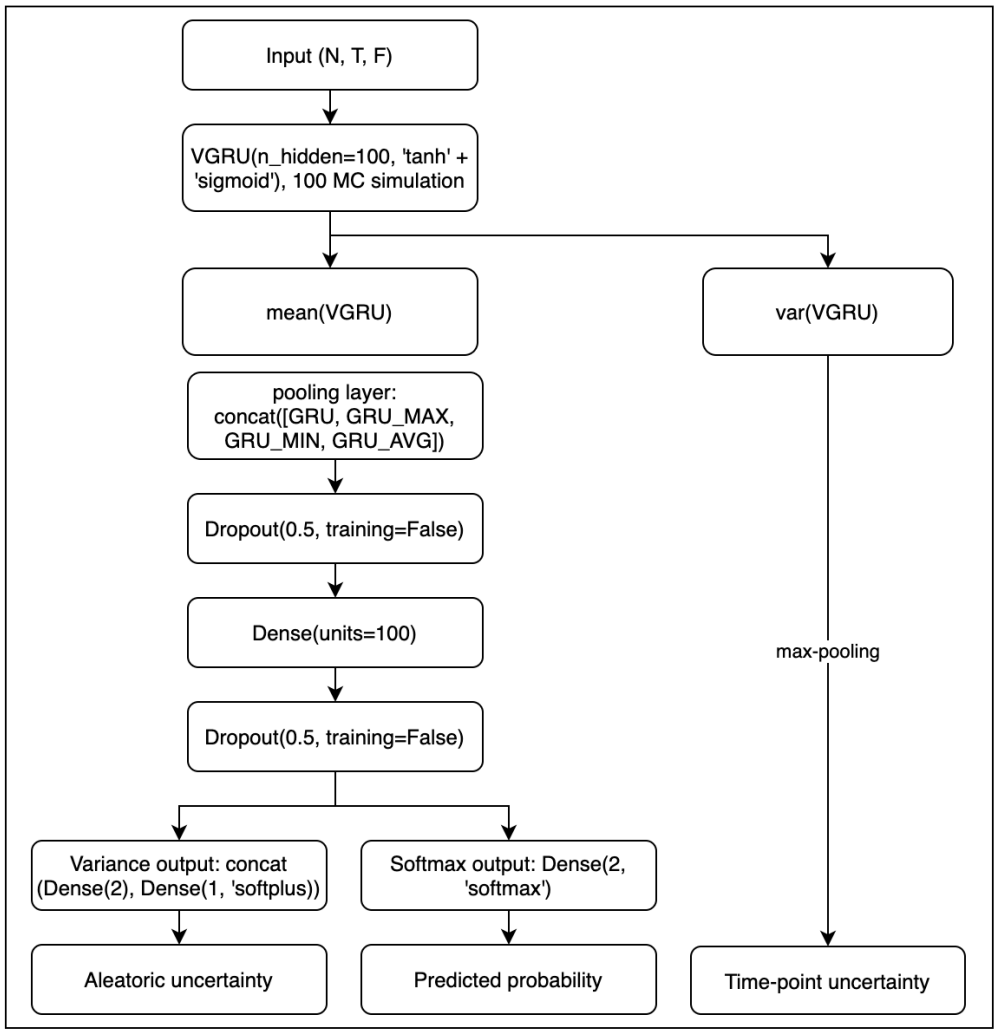

Figure 4.5: The network structure and configurations of the proposed VGRU model.

4.3.3 Uncertainty Verification

For the temporal feature uncertainty, we are still facing a problem: the lack of labels like the image pixels in the semantic segmentation. Therefore, verification experiments similar to what we did in Chapter 3 can to some extent serve as a model validation method. Below we will describe the experiments setup.

The temporal feature uncertainty verification relies on manually adding noises to the input features. There are three reason: (1) the estimations of temporal feature uncertainty are at the first layer of the network and directly connect to the input data, (2) experiments reveal that temporal feature uncertainty is related to the aleatoric uncertainty, but not epistemic uncertainty (will be investigated in Section 4.4.2), and (3) a temporal feature uncertainty verification experiment like what we did in the model uncertainty verification is not realistic (imagine assigning different temporal features into groups). Therefore, we take the following steps to verify the temporal feature uncertainty: using a trained VNN model, we calculate the average uncertainty of a temporal feature - time span for VCNN and time point for VGRU. Then we randomly reduce the data points for this time span and create 5 groups with 0% (all data removed), 25%, 50%, 75%, and 100% (all data retained). We then evaluate how the estimated temporal feature uncertainty change, and whether the change was consistent with the patient-level aleatoric uncertainty as expected. Since the volume of data for a single time point is not sufficient, we perform the experiment for VGRU by selecting the time span of length three with largest sum of point uncertainty.

4.4 Results and Discussions

4.4.1 Model Performance Evaluation

We continue using the AUC score as the evaluation metric and 5-fold cross-validation to report the model performance. Table 4.1 shows the AUC with 95% confidence interval of all the five tasks. Although the VNN model performance drops for all the

tasks compared to corresponding HNN models, the differences are not statistically significant. The mean AUC of a VNN model is around 0.01-0.02 smaller than that of a HNN model, while the variances are between 0.008 and 0.03. We can confidently claim that for the estimation of temporal feature uncertainty, this level of performance loss is trivial (even not a loss according to the statistical significance).

Table 4.1: Model performance (AUC scores) comparison for 5 binary tasks

	MIMIC	**PhysioNet**			
	Mortality	**Mortality**	**Stay < 3**	**Cardiac**	**Recovery**
Baseline Models					
HCNN-DE	0.8502 ± 0.0128	0.7532 ± 0.0196	0.8392 ± 0.0116	0.9053 ± 0.0186	0.8630 ± 0.0263
HCNN-DR	0.8483 ± 0.0187	0.7549 ± 0.0209	0.8280 ± 0.0232	0.8975 ± 0.0198	0.8554 ± 0.0210
HGRU-DE	$\mathbf{0.8659 \pm 0.0172}$	0.7973 ± 0.0183	$\mathbf{0.8646 \pm 0.0219}$	$\mathbf{0.9629 \pm 0.0120}$	$\mathbf{0.9102 \pm 0.0217}$
HGRU-DR	0.8618 ± 0.0216	$\mathbf{0.7989 \pm 0.0190}$	0.8565 ± 0.0241	0.9580 ± 0.0085	0.9002 ± 0.0297
Proposed VNN Models					
VCNN-DE	0.8432 ± 0.0218	0.7382 ± 0.0143	0.8149 ± 0.0116	0.8833 ± 0.0156	0.8482 ± 0.0313
VCNN-DR	0.8378 ± 0.0167	0.7369 ± 0.0234	0.8130 ± 0.0192	0.8795 ± 0.0238	0.8384 ± 0.0199
VGRU-DE	0.8489 ± 0.0189	0.7825 ± 0.0206	0.8489 ± 0.0192	0.9459 ± 0.0228	0.8990 ± 0.0263
VGRU-DR	0.8509 ± 0.0224	0.7837 ± 0.0213	0.8389 ± 0.0161	0.9428 ± 0.0155	0.8892 ± 0.0189

The reported numbers are the mean AUC and standard errors for 95% confidence interval over 5-fold cross validation. The bold numbers indicate the best performance in that (column) group.

4.4.2 Comparing the Patient Uncertainty and Temporal Feature Uncertainty

Next we will explore how the temporal feature uncertainty differs between different patient-level uncertainty groups. We divide the data into two groups by the median of the data uncertainty or model uncertainty, then compare the average temporal feature uncertainty of each group. Two mortality datasets are used and the 5-fold cross-validation results are listed in Table 4.2. For both VCNN and VGRU models, the average feature uncertainty is high when the model uncertainty is high. This difference is significant for most of the experiments except for the VGRU model using MIMIC-mortality data. The results indicate that when the patient-level data uncertainty is high, it can be reflected in the temporal feature uncertainty. On the other hand, there are no significant differences between the groups with high or low model uncertainty: the averages temporal feature uncertainty are all at a similar level for each group of experiments, which indicates that the temporal feature uncertainty tends to capture the noises in the data. Therefore in the next section, we will perform

the feature uncertainty verification experiments with the approach similar to data uncertainty verification experiments.

Table 4.2: Average temporal feature uncertainty of four different patient groups, divided at medians of both aleatoric uncertainty and epistemic uncertainty

MIMIC-Mortality		VCNN	VGRU
Aleatoric Uncertainty	High	0.1502 ± 0.0128	0.2336 ± 0.0183
	Low	0.1127 ± 0.0181	0.2038 ± 0.0208
Epistemic Uncertainty	High	0.1420 ± 0.0182	0.2102 ± 0.0264
	Low	0.1449 ± 0.0129	0.2219 ± 0.0197
PhysioNet-Mortality		VCNN	VGRU
Aleatoric Uncertainty	High	0.1713 ± 0.0134	0.2478 ± 0.0177
	Low	0.1235 ± 0.0099	0.2092 ± 0.0143
Epistemic Uncertainty	High	0.1602 ± 0.0235	0.2291 ± 0.0287
	Low	0.1384 ± 0.0277	0.2205 ± 0.0307

4.4.3 Verification of Temporal Feature Uncertainty

To verify whether the estimated temporal feature uncertainty does capture noise from the corresponding time spans/points, we manually create several datasets with different levels of noise. Unlike Section 3.4.3 in which we removed portions of data points from all features, we only introduce noises for specific time spans, so that the corresponding estimated feature uncertainty can be verified. With the original data (100% retained), we identify a time span with highest uncertainty for each task and model combination, then create 5 datasets with 0% (all data removed), 25%, 50%, 75%, and 100% (all data retained) of data points in the time span to evaluate the change of the estimated temporal feature uncertainty and corresponding patient-level data uncertainty. We conduct this experiment on two mortality datasets and two VNN models (VCNN-DP and VGRU-DP). The results are displayed in Figure 4.6 & 4.7. Using MIMIC-mortality dataset and VCNN as an example, the model assigns the highest average uncertainty to the time span at 30-33 hours after ICU admission. As shown in the first and second bar charts of Figure 4.6, with more and more data

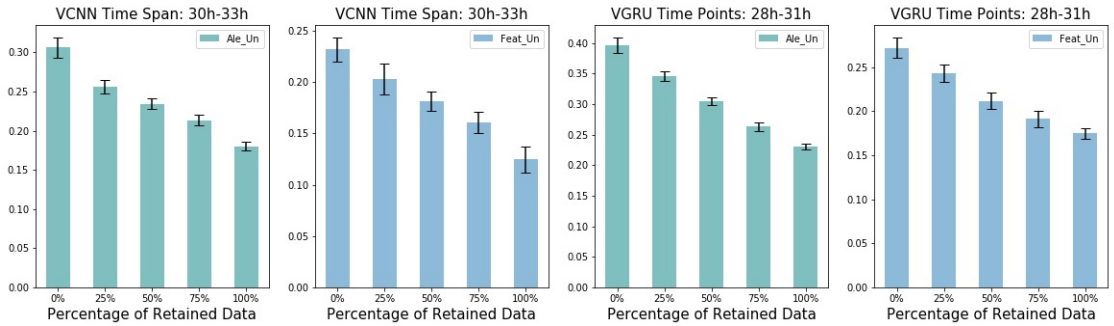

Figure 4.6: Temporal feature uncertainty and aleatoric uncertainty verification of VCNN and VGRU using MIMIC-Mortality dataset.

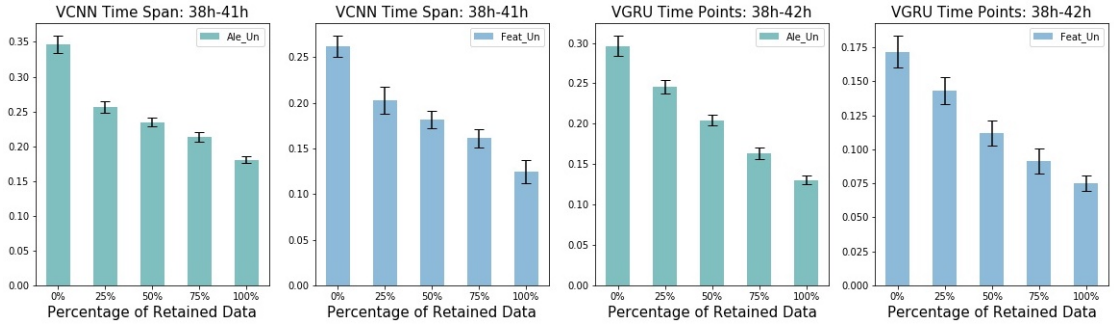

Figure 4.7: Temporal feature uncertainty and aleatoric uncertainty verification of VCNN and VGRU using PhysioNet-Mortality dataset.

points randomly removed from this time period (until no data left), the captured average temporal feature uncertainty (blue bars) increases from around 0.12 to almost 0.23 and the average data uncertainty also increases from 0.18 to 0.30 correspondingly. Similarly for the VGRU model that assigned highest uncertainty to the 28h-31h time span (third and fourth images of Figure 4.6), both temporal feature uncertainty and data uncertainty increases as the retained data decreased gradually from 100% to none. The trends are all the same for the two models trained with PhysioNet-mortality dataset (Figure 4.7). These exhibited trends indicate that our models are capable of capturing the data noises in the temporal features and more importantly, the changes are also reflected in the estimated patient-level data uncertainty.

To further analyze how does temporal feature uncertainty relates to data uncertainty, we sort the data uncertainty estimations, pick a case with high data uncertainty

and high loss (categorical cross-entropy), and present the patient's records within the highest-uncertainty time span to a doctor. More specifically, our VGRU-DP model assigns low probability of mortality (<0.2) to the patient, while the ground truth is 1 (patient died in the ICU). The model also estimates a high data uncertainty (top-5 among over 400 patients), identifies that 38-43 hours after the admission is the most 'uncertain' time period. We convert the raw data of these 5 hours into an Excel sheet (as shown in Figure 4.8) and present it to the doctor. Several features are noticeable: the 'GCS' value (Glasgow Coma Scale) decreased from 14 to 3 (from fully awake to deep coma), the 'MechVent' (Mechanical Ventilation) was used three times, and the 'HR' (heart rate) dropped drastically then kept at a low level. These important risk factors were somehow 'blurred' by other features (noises) hence ignored by the model. A possible cause is the frequency of the feature record. While other features were recorded frequently, GCS and MechVent were recorded twice and three times, respectively. These strong signals separated by long time steps can easily be ignored by the normal deep learning model due to different reasons (lack of long-term memory or imputation noise). This case indicates that the presence of uncertainty estimations can help prevent the wrong decisions and save the clinicians' time when checking for the causes when uncertainty alerts them.

Time (hh:mm:ss)	DiasABP	FiO2	GCS	HR	MAP	MechVent	SysABP	Urine	pH	PaO2
38:11:00	62		14		91					
38:19:00					72					
38:26:00	33			88	72		148			
38:27:00									7.24	54
38:41:00	54			95	92		171			
38:56:00	55			94	100		194			
39:11:00	53	1		101	91	1	178	40		
39:26:00	40			82	53		140			
39:41:00	34			71	58		126			
39:59:00									7.28	110
40:11:00	33			64	48	1	92	10		
40:41:00				65	47		88			
40:55:00	32									
41:11:00		0.8		65	69		124	10		
41:41:00	45	0.8				1				
42:11:00	42		3	71	65		121	7		
42:16:00									7.35	62
42:41:00	41			68	58		107			

Figure 4.8: Patient records presented to a doctor: the time span (38h-43h) with highest uncertainty.

4.5 Conclusion and Future Works

In this chapter, we proposed novel variational layers based on the HNN model. The VNN frameworks were capable of estimating the uncertainty for temporal feature like time spans or points. We evaluated the model performance with AUC scores, making sure that the proposed models preserved good predictive power as the HNNs. By comparing the temporal feature uncertainty with population level data/model uncertainty, we claimed that the feature-level uncertainty is a reflection of patients' data uncertainty. Based on this finding, we performed uncertainty verification experiments that were similar to the data uncertainty verification in Chapter 3. Experiment results verified that the VNN models were capable of capturing the increased noises in the temporal features. We also presented a patient's case to a doctor for the user verification of the data.

There are several limitations or future directions for the works in this chapter. First, the model requires more systematic user-level evaluation and validation; therefore, we will discuss a user study in the next chapter. Second, a medical feature uncertainty is desired; with the proposed model, we can only estimate uncertainty for the temporal features; if there are an enormous number of medical features within a certain period, the doctors' work will be very difficult; while it still a challenging task, being able to identify the medical features with high uncertainty is extremely important.

CHAPTER 5: A USER STUDY FOR UNDERSTANDING CLINICIANS' PERCEPTION OF OUR UNCERTAINTY MODEL

5.1 Background

In the previous chapters, we were able to estimate the data uncertainty and model uncertainty in the EHR-based deep learning models. We also proposed several methods for the estimation of temporal feature level uncertainty. Our experiments verified that the uncertainty estimations were able to react to the change their sources (towards the correct directions). However, there still existed a lack of solid evaluations for the captured uncertainty. Unlike the model predictions, uncertainty estimations at the patient or feature level did not have ground truth or labels that they could be evaluated upon. Since the purpose of estimating the uncertainty was to improve the clinicians' trust in the deep learning models, we proposed to present our model outcome and get it evaluated by these real end-users.

With the prosperity of machine learning models in healthcare research and applications, researchers started to pay closer attention to the 'user experience'. A predictive model with high accuracy may not necessarily be accepted by the clinicians. Although the sophisticated machine learning models were able to automatically take all complex information into consideration and provide individualized decisions, clinicians and physicians still need to understand it and explain the process of decision making to the patients [8, 71]. Therefore, meaningful and explainable output is an extremely important step for the practical adoption of machine learning model in clinical decision making. Great efforts have been made by the research community to address the issue. Examples include intrinsic explainable models, post-hoc global explanations, and post-hoc local explanations. However, most of these explanations were extracted

based on the intuition of researchers instead of the end-users [6].

In recent years, the healthcare informatics research community started to pay more attention to the doctors' need when developing the machine learning model explainability. In the survey study conducted by Tonekaboni *et al.* [8], clinicians from the ICU or Emergency Department (ED) with previous machine learning experience were interviewed to understand 'what makes a model explainable' for these end-users. Based on the feedback, reliable machine learning model explanations were divided into several aspects, including feature importance, instance level explanation, uncertainty, temporal explanations, and transparent design. They also identified three metrics for explainability evaluations: domain appropriate representation, potential actionability, and consistency. We also followed these metrics when designing the experiments in this thesis. Another notable survey study investigated the clinicians' understanding, explainability, and trust in the machine learning risk prediction models. In this work, Diprose *et al.* [71] designed a set of survey instruments to systematically compare the clinicians' feedback on different levels and methods of model explanations. The ability of explaining a machine learning output to a patient was proved to be significantly related to the clinicians' trust in the model, although the differences between explanation methods they evaluated were not significant. This survey set the questions and answers as simple as possible and only focused on the problem/hypothesis they wanted to test. For the design of our survey instruments, we adopt a similar strategy to retain the participants with simple multiple-choice questions and compared these answers to derive a trend or statistic significance.

5.2 Survey Design

5.2.1 Survey Instrument

In this survey, we wanted to evaluate whether the uncertainty modeling improved the clinicians' trust in the deep learning model outcome. Based on the experiment results from the previous chapters, we designed a online survey that provide the

survey context to the participants and ask for their feedback on each integration of the predictive model. The context included a de-identified patient's ICU record within the first 48 hours of admission, the deep learning model estimation of the mortality risk, and the representations of both patient-level and feature-level uncertainty. For the selection of the patient we considered the ones with (1) high difference between model prediction and ground truth, (2) high patient-level uncertainty, and (3) a time period in the record with high feature-uncertainty.

The patient's records were presented in two parts.

- **Data at admission.** As shown in Table 5.1, basic demographic information and ICU types were recorded at the time of admission.

- **Records of first 48 hours.** The presentation of the patient records is difficult for the survey purpose. The data composed of 35 medical features and 155 different time points. A sample of the data can be found in Figure 5.2 - these are just first 10 of thousands of lines in the patient's record. Presenting them all together in text format would easily consume over 20 minutes of the survey time and could not help the participants understand the patient's situation. To create a better presentation, we (1) categorized the features into groups including [blood oxygenation level] (Figure 5.1), [kidney (comprehensive metabolic penal)] (Figure 5.2), [blood count] (Figure 5.3), [vital signs] (Figure 5.4), [blood pressures] (Figure 5.5), and [others] (Figure 5.6), (2) abandon some useless features (verified by a doctor) that only showed once in the record, and (3) drawn line plots of each group in Tableau software and uploaded to the cloud for participants reference. In addition to the lines, we also used color bands to indicate the reference values for the features that apply.

We presented the visualization to doctors, public health practitioners, healthcare informatics researchers, and machine learning researchers for feedback (not the survey

Table 5.1: Patient's data at admission

Age	70
Gender	Male
Height	173 cm
ICU Type	Cardiac Surgery Recovery Unit

Table 5.2: Sample of patient's data in the text (Excel) format

Time(hh:mm)	**Parameter**	**Value**
00:00	RecordID	13xxxx
00:00	Age	70
00:00	Gender	1 (stands for male)
00:00	Height	173
00:00	ICU Type	2 (Cardiac Surgery Recovery Unit)
00:42	pH	7.45
00:42	PaCO2	34
00:42	PaO2	344
01:11	DiasABP	67
01:11	FiO2	1

participants). This visualization method is proved to be much more informative and efficient than simply presenting the text. It only took an average of about 5-10 minutes for the user to look through, compared to over 20 minutes

Then we split our BNN model outputs into three parts, presented each part to the participants sequentially, and asked for their feedback correspondingly. The setup of the survey questions was inspired by the work of Diprose *et al.* [71]. We tried to make the questions as simple as possible for two reasons. First, simple choices would lead to more responses: in Diprose *et al.*'s work, they only asked for *Yes/No* responses for most of the questions and got 13% response rate; if too many choices were given, the participants were more likely to quit half-way; the second reason was the association between answer variances and statistical significance: for the questions to be tested for significant difference (e.g. Student's t-test), the answers scaled from 1-10 were less likely to get the job done than the answers with only *Yes/No/Maybe* choices. Following are the three stages of model output presentations

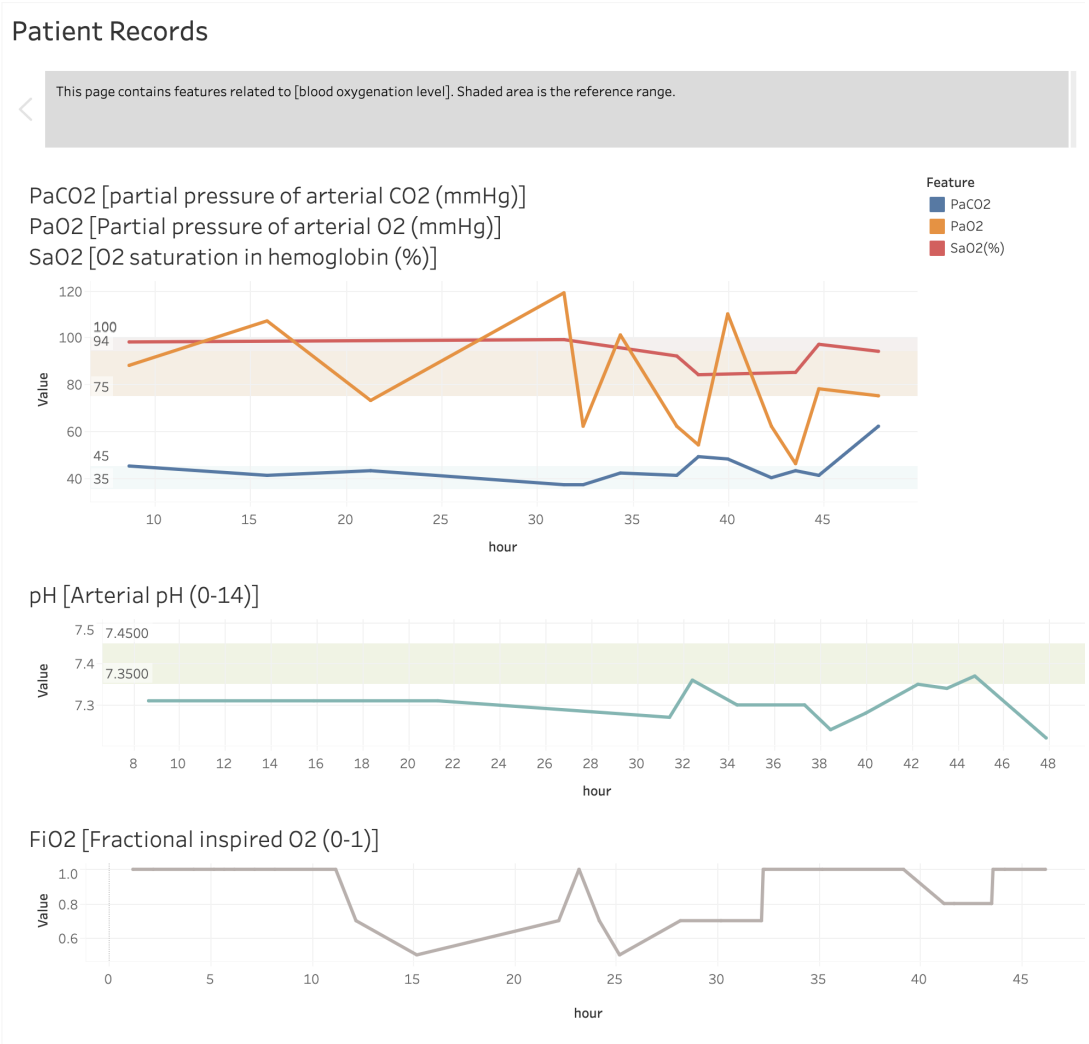

Figure 5.1: Tableau visualization of the patient's records related to [blood oxygenation level].

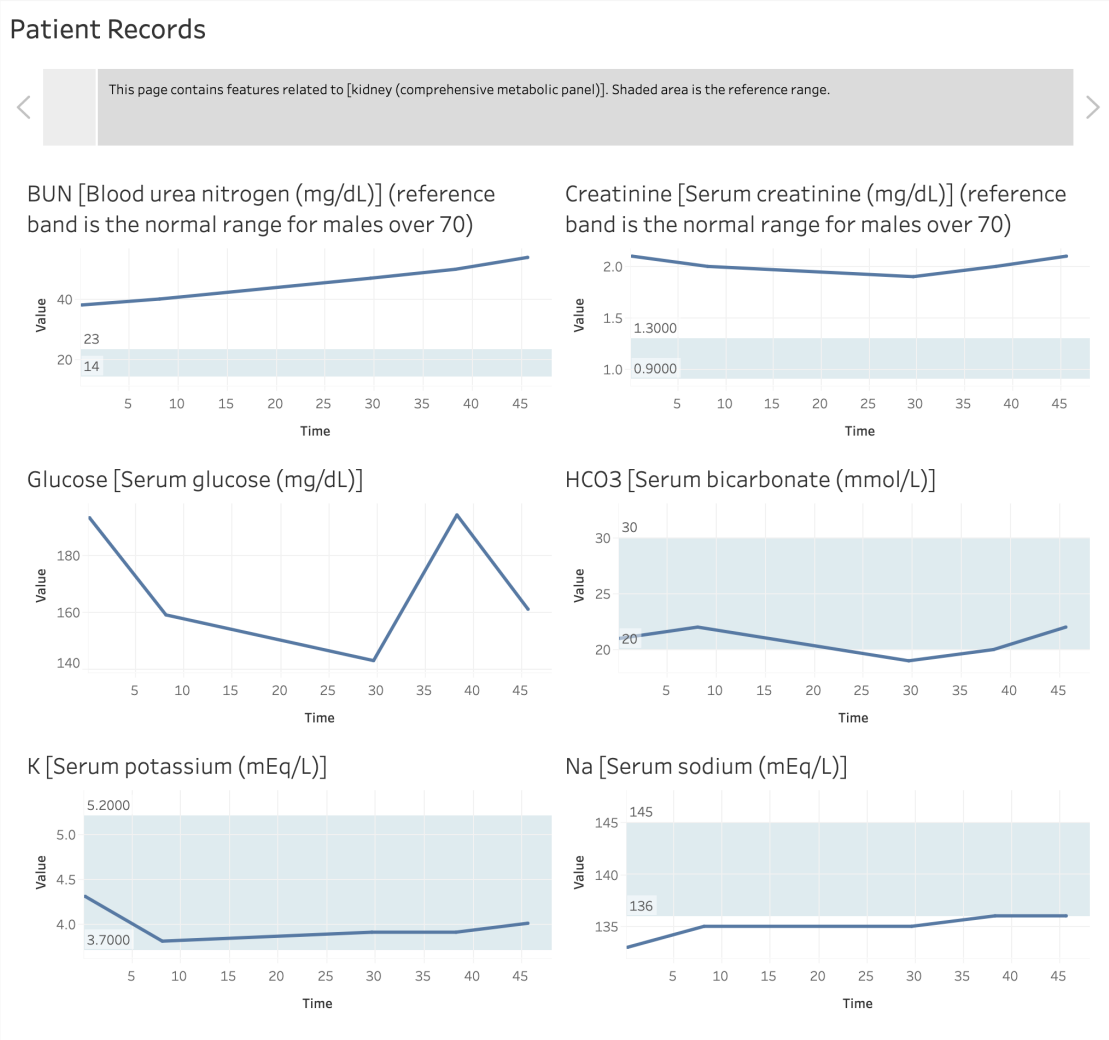

Figure 5.2: Tableau visualization of the patient's records related to [kidney (comprehensive metabolic penal)].

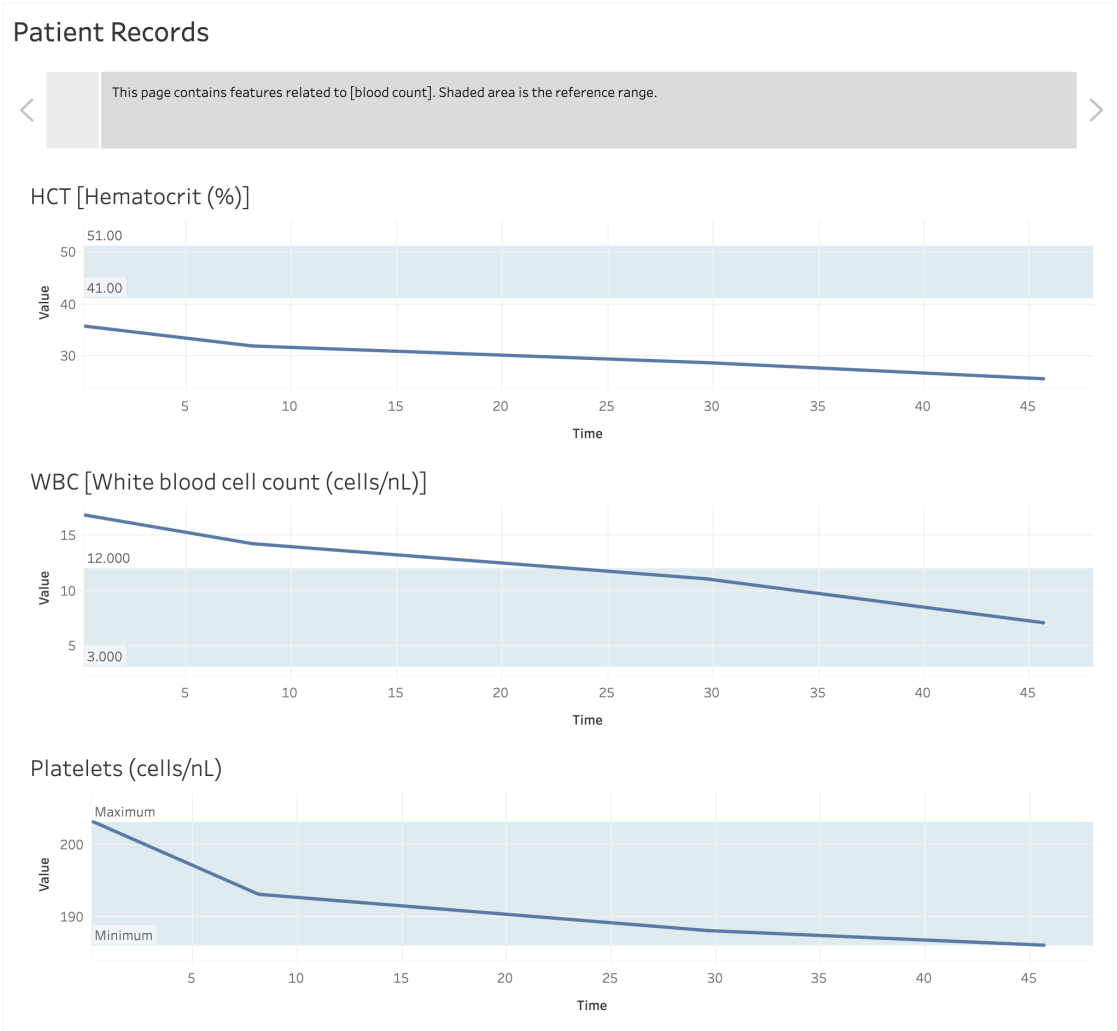

Figure 5.3: Tableau visualization of the patient's records related to [blood count].

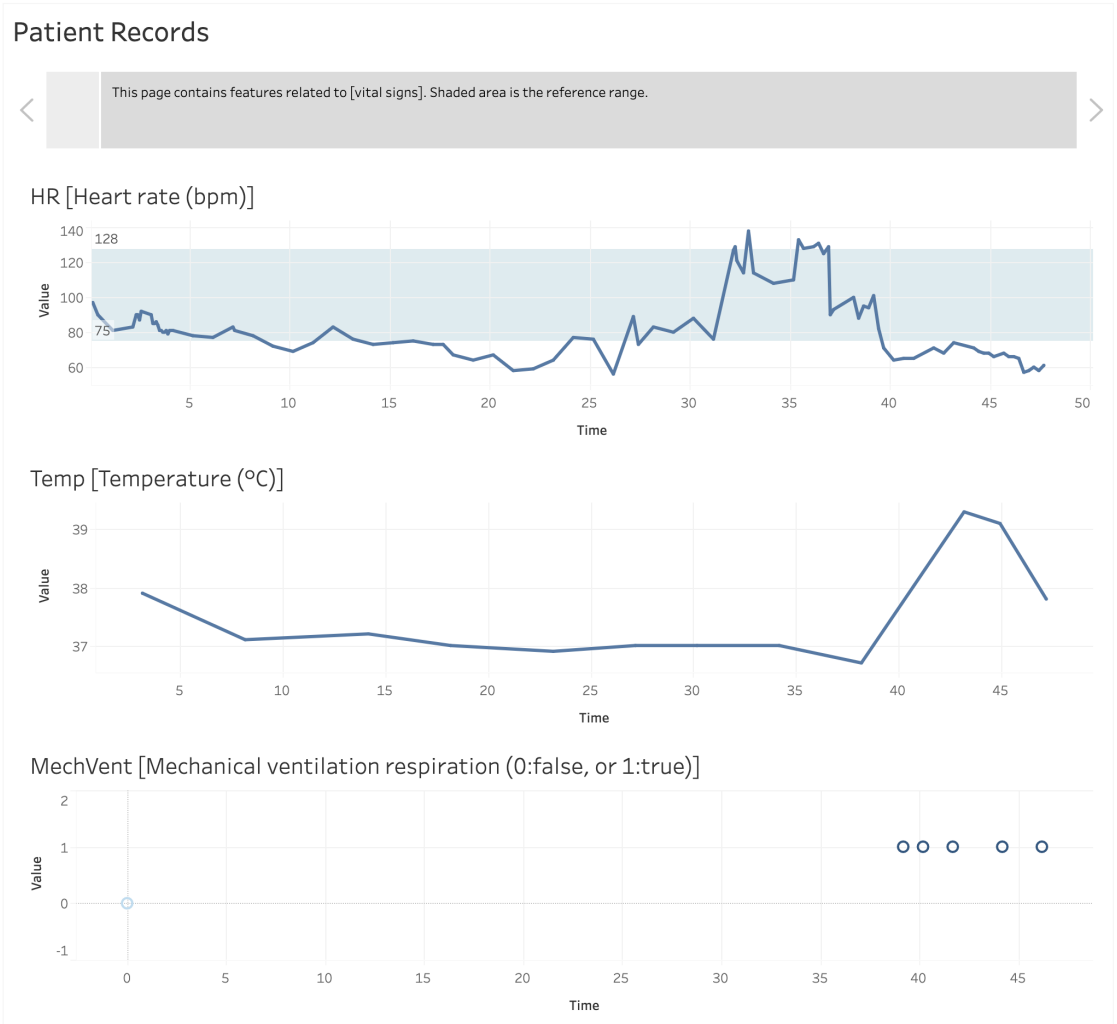

Figure 5.4: Tableau visualization of the patient's records related to [vital signs].

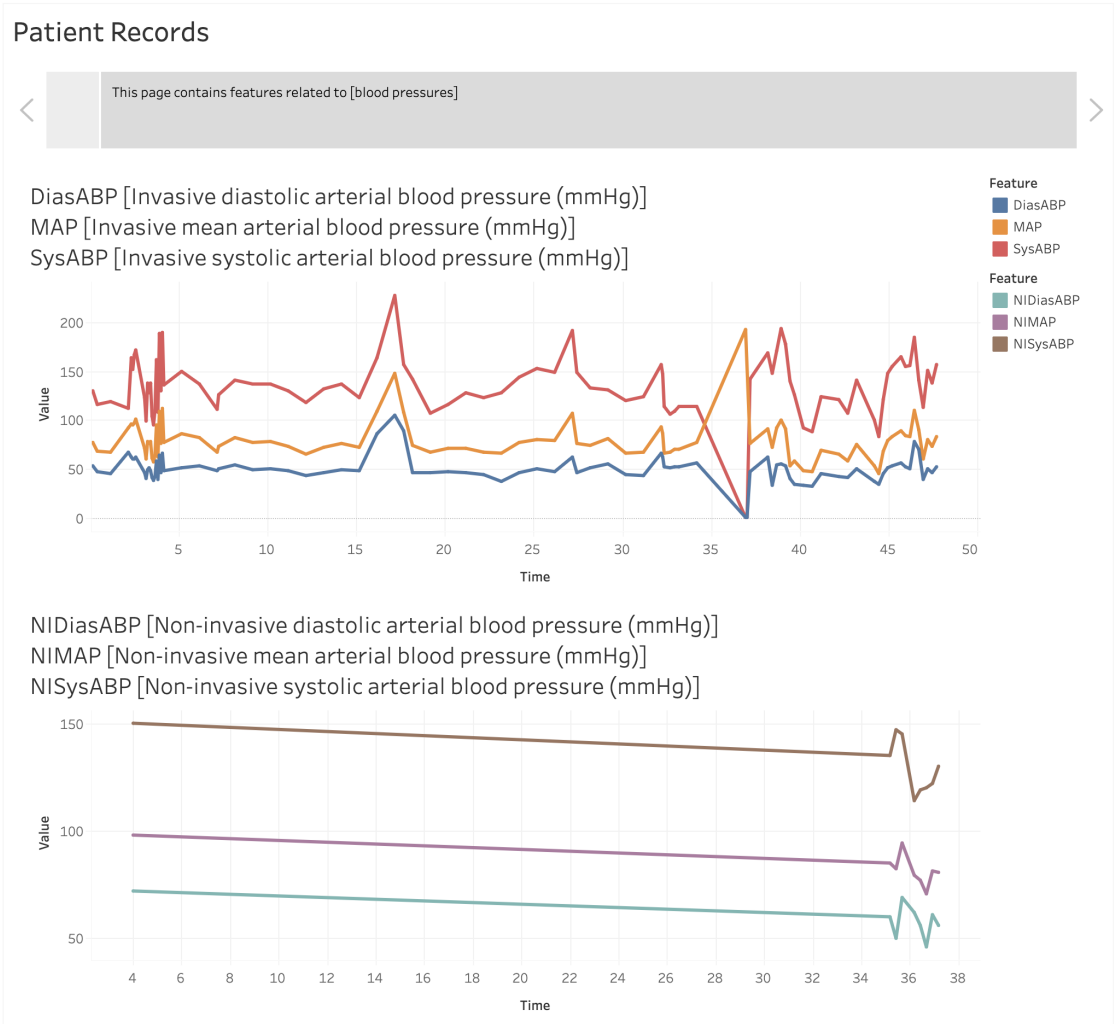

Figure 5.5: Tableau visualization of the patient's records related to [blood pressures].

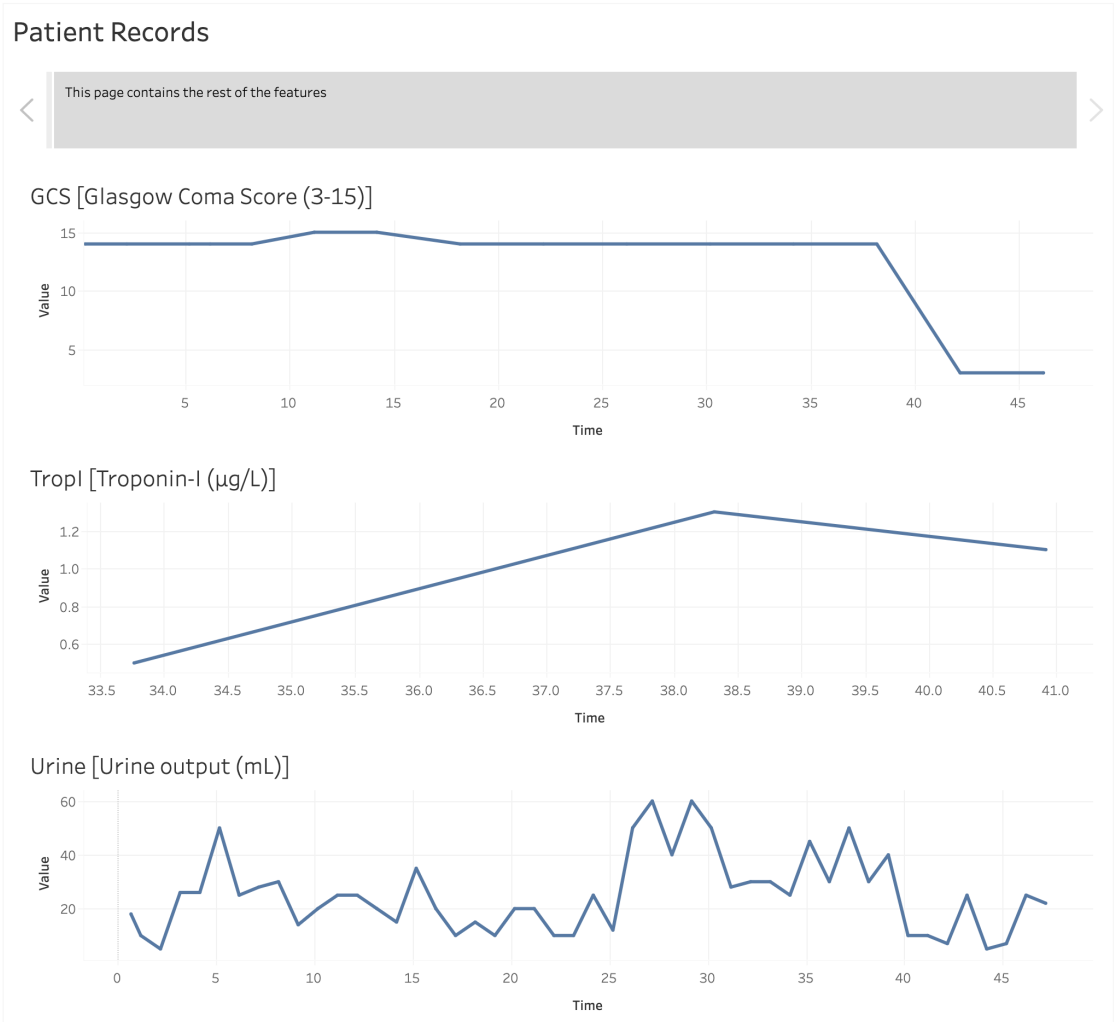

Figure 5.6: Tableau visualization of the patient's records related to [others].

and corresponding multiple-choice questions. The last part is the open-ended question asking for the participants' most-wanted feature from the model.

- **Feedback on the point estimation of mortality risk.** If our model tells you the probability that this patient will die in the ICU is 40%, what do you think? Please make your prediction and select how confident you are in the following questions. *Question: Do you think the model output (probability) is useful or not? [Yes/No/Maybe]*

- **Feedback on the point estimation of mortality risk with patient-level uncertainty representation.** In addition to the probability, the model tells you that it is confidence level about its prediction is low, what do you think? *Question: Do you think the model output (probability and confidence level) is useful or not? [Yes/No/Maybe]*

- **Feedback on the point estimation of mortality risk with patient-level uncertainty and feature-level uncertainty representation.** The model tells you that these time/medical features are worth further looking: (1) the patient records between 38-42 hours after the admission; (2) medical features - Mechanical Ventilation Respiration (MechVent), Heart Rate (HR), Urine Output (Urine), and Glascow Coma Score (GCS). Would you take a closer look at the data and update your feedback? *Question#1: Do you think the model output (probability, confidence level, and the important times/features) is useful or not? [Yes/No/Maybe]. Question#2: Can you pick one or two features that are most useful to you? [time feature: 38-42 hours after the admission/Mechanical Ventilation Respiration (MechVent)/Heart Rate (HR)/Urine Output (Urine)/Glascow Coma Score (GCS)]*

- **General feedback.** Following the three layers of model output and questions, we finished the survey by asking an open question: *If we try to develop a*

computer model to assist the clinicians and physicians, what features/properties would you look for in this model that will encourage you to use it in your daily work?

5.2.2 Participants

This study was approved by the UNCC institutional review board (IRB). The link to the Google Form was then sent to three groups of healthcare practitioners: students enrolled in the Doctor of Nursing Practice (DNP) program of UNCC, doctors from Atrium Health, and some of the students enrolled in the Master of Publich Health program of Harvard University. Since the survey recruiting advertisement was sent to listserv email address, we were not able to estimate the number of potential participants therefore the response rate could not be estimated.

5.3 Results and Discussions

The survey was sent out with the expectation of getting about 25 responses. However, due to the outbreak of the COVID-19 pandemic (Coronavirus Disease 2019), most of our targeted participants were devoted to fighting against the disease. Therefore, we only received 9 responses, with 3 from the Harvard MPH program and 6 from the UNCC DNP program.

Although the sample size were not sufficient for the statistical test, we were still able to observe meaningful results from the survey responses. We draw the count of each answer (Yes/No/Maybe) and compared their trends as the output contents increased from only probability of death to a combination of probability of death and two levels of uncertainty (as shown in Figure 5.7). The same question was asked after presenting each layer of output to the participants - whether they think the model was useful or not.

The number of the answer 'Yes' was only 1 when a point estimation was presented (i.e. the output we usually got from a normal binary-task deep/machine learning

model); it increased to 4 when the patient-level uncertainty was 'translated' to a model confidence level; when we further notified the participants that the feature-level uncertainty indicated that a certain period of time or several medical features in the patient's record was worth further looking, the number of participants who answered 'Yes' increased to 7, accounting for almost 78% of the total number. This trend proved that solid and meaningful uncertainty propagation can notably improve the clinicians' trust in the output provided by the deep learning models. The patient-level uncertainty could be translated into confidence representation, hence confirming the doctors' decision or alerting them when there existed conflicts. The feature-level uncertainty could help them quickly locate the possible problem, as well as explain the output to the patients when necessary. Accordingly, the count of answers 'No' and 'Maybe' decreased respectively, as the output contents became richer. The results showed that with plain deep learning output, 2 participants found themselves not trusting in the model at all; The uncertainty estimations changed their minds and both selected 'Maybe' or 'Yes' for the remaining questions. This change is even more inspiring than the increasing numbers of 'Yes' - converting the users who did not believe in the model at all is the most important step of gaining their full trust.

In addition to the *Yes/No/Maybe* questions, we listed the high-uncertainty temporal/medical features to the participants and asked them to pick 1-2 features that were most useful to them. We first identified the time period by comparing the time-point uncertainty estimations given by the VGRU models, then picked up the medical features that showed in this time period. The results are presented in Figure 5.8. Not surprisingly, the temporal feature was most selected (6 votes), since it covered multiple medical features and successfully attended to the time period that contributed the most to the high patient-level uncertainty. Two important medical features got 3 votes: 'Mechanical Ventilation Respiration' was changed from 'off' to 'on' during the 38-42 hour period, which is an important indicator of the patient's situation; simi-

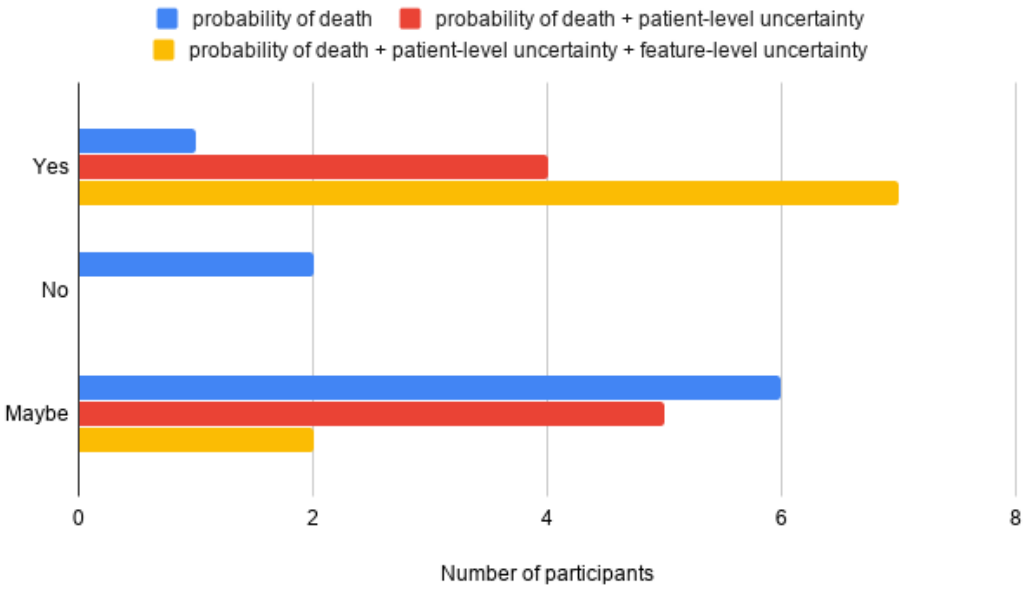

Figure 5.7: Change of participants' trust in the model by the contents of output.

larly, the patient's heart rate dropped drastically during the time period, indicating the possibility of worsened health condition.

The last question we asked in this survey was open-ended: we wanted to understand what was the most demanded feature from an AI model by the users and got 7 responses. The answers were used to generate a word cloud. From Figure 5.9 we were able to verify the importance of explainable machine learning models: the clinicians required better explanations to better understand the model results or trends; they also wanted to rely on the models to alert them when a potential problem existed.

5.4 Conclusions

In this chapter, we designed an online survey to evaluate the clinicians' feedback on our proposed uncertainty models. Although the number of responses was too small for statistical significance tests, we were able to observe some meaningful trends from the answers. The results verified that the uncertainty estimations had positive effects on improving clinicians' trust in the deep learning models. However, some limitations needed to be overcome if another survey was conducted in the future. First, to

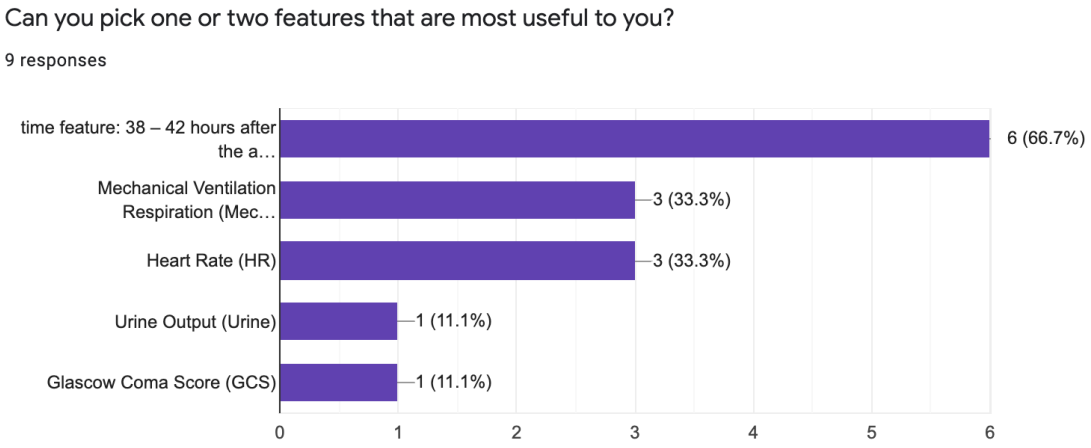

Figure 5.8: Feature importance voted by the participants.

Figure 5.9: Word cloud generated from the open-ended question: the most wanted AI model features.

confidently report the results from statistical tests, more potential participants should be reached out and more responses should be recorded. Second, due to the small number of responses, we only presented one patient's record in the scenario of wrong-prediction and high-uncertainty; if possible, the responses to patient scenarios of right-prediction and low-uncertainty should also be evaluated for a more comprehensive study. Another limitation is the lack of generalizability, caused by the very small sample size; the results may not hold for different groups of survey participants, clinical tasks, or patients.

CHAPTER 6: CONCLUSIONS AND FUTURE WORKS

In the era of big data, machine learning and deep learning models have been widely adopted in various domains including the healthcare industry. However, the lack of explainability and trustworthiness has prevented its application in high-stake clinical risk prediction tasks. Modeling uncertainty will provide an extra layer of confidence representations to the users, as well as alert the clinicians when the model is 'uncertain' of its predictions. Motivated by the importance of modeling uncertainty, we proposed to comprehensively study the uncertainty in the EHR-based deep learning models.

We claim five major contributions in this dissertation study:

- We reviewed the Bayesian learning algorithms for uncertainty estimation and their existing applications in the EHR-based clinical risk prediction. The importance of modeling uncertainty was emphasized and huge research gap was identified.

- We proposed a series of novel deep learning frameworks that can estimate the data and model uncertainty for a patient. The HNN models/modules were developed to estimate the heteroskedastic aleatoric uncertainty, which captured the patient-dependent noises in the EHR data. We applied Deep Ensemble and Dropout methods to capture the epistemic uncertainty that represented the noises from the model structure and parameters. We proposed to combine these methods with the normal deep learning models such as CNN and GRU to estimate both types of patient-level uncertainty in one model.

- We proposed novel deep learning frameworks that can estimate the uncertainty

of temporal features. We developed the VNN models by proposing variational CNN and GRU layers, and estimated the temporal feature uncertainty, which explained the time spans that contribute the most to a high patient-level uncertainty.

- We proposed a series of experiments that overcome the lack of uncertainty ground truth issue and were able to verify the validity of the estimated data uncertainty, model uncertainty, and temporal feature uncertainty.

- We conducted a clinician user study in hope of further validating the proposed models. The survey instrument including data representation and questions were well designed. Although sufficient responses were not obtained to draw statistical significance, the trends it exhibited still indicated that patient-level and feature-level uncertainty estimations were improving the clinicians' trust in the deep learning model outputs.

While the proposed models and uncertainty estimations can already improve the deep learning model trustworthiness and explainability, there are several major limitations. We explain them with corresponding future directions:

- The sizes of the datasets for modeling uncertainty were small. For the vanilla deep learning model implementation in Chapter 2, I was able to access a commercial employee claim database with 20 million patients and billions of records (during an industry internship). But for the uncertainty modeling, we only worked on the MIMIC-III and PhysioNet data, with around 34,000 and 12,000 patients, respectively. With large datasets, estimating data uncertainty is important since the model uncertainty can be explained away [1]. Therefore, experiments on verifying zero/small model uncertainty and investigating the data uncertainty can be future directions if we gain access to larger datasets.

- Since the major task of this dissertation is trustworthiness and explainability (but not performance), we only used simple CNN and GRU models for the implementation of Bayesian learning methods. However, these frameworks are applicable for more sophisticated deep learning models such as Convolutional RNN (CRNN), Temporal CNN (TCNN), and many other modular network structures. One future direction is to implement the proposed frameworks in these well-performed DNNs.

- In Chapter 4, we estimate the temporal feature uncertainty. However, estimating the medical feature uncertainty remains very difficult due to the structure of the networks. Attention models like RETAIN [30] were developed to train medical feature attention and gain some degree of interpretability, but these models rely on data perturbations and can be very difficult to perform inference. The future direction on estimating the medical feature uncertainty will lean towards the deep learning algorithm development.

- In our post-hoc analysis, the full time length of data (48 hours) were used for both training and validation, which is not applicable for the real-world scenarios. Ideally, we should perform the analysis on the first 24 hours and make some recommendations, then validate these decisions in the second 24 hours. However, the data volume was not sufficient. Therefore, a future direction is getting datasets that are richer in the temporal dimension, so they can fit better for the real-time clinical decision support.

- Due to the COVID-19 pandemic, the number of responses in the user study was only 9. We were not able to test statistical significance for the survey hypothesis. If possible, we will conduct a user study with more potential participants and try to improve the response rate. Furthermore, with enough participants, different survey groups should be created to compare the users' trust. For example, a

patient with high uncertainty v.s. a patient with low uncertainty should be presented to two groups of participants and evaluate the effect of uncertainty values.

REFERENCES

[1] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Advances in neural information processing systems*, pp. 5574–5584, 2017.

[2] P. B. Jensen, L. J. Jensen, and S. Brunak, "Mining electronic health records: towards better research applications and clinical care," *Nature Reviews Genetics*, vol. 13, no. 6, p. 395, 2012.

[3] C. M. DesRoches, E. G. Campbell, S. R. Rao, K. Donelan, T. G. Ferris, A. Jha, R. Kaushal, D. E. Levy, S. Rosenbaum, A. E. Shields, *et al.*, "Electronic health records in ambulatory careâa national survey of physicians," *New England Journal of Medicine*, vol. 359, no. 1, pp. 50–60, 2008.

[4] C. Shivade, P. Raghavan, E. Fosler-Lussier, P. J. Embi, N. Elhadad, S. B. Johnson, and A. M. Lai, "A review of approaches to identifying patient phenotype cohorts using electronic health records," *Journal of the American Medical Informatics Association*, vol. 21, no. 2, pp. 221–230, 2013.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[6] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *arXiv preprint arXiv:1808.00033*, 2018.

[7] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, p. 93, 2019.

[8] S. Tonekaboni, S. Joshi, M. D. McCradden, and A. Goldenberg, "What clinicians want: Contextualizing explainable machine learning for clinical end use," *arXiv preprint arXiv:1905.05134*, 2019.

[9] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, p. 452, 2015.

[10] M. Krzywinski and N. Altman, "Points of significance: Importance of being uncertain," 2013.

[11] Y. Gal, *Uncertainty in deep learning*. PhD thesis, PhD thesis, University of Cambridge, 2016.

[12] J. Heo, H. B. Lee, S. Kim, J. Lee, K. J. Kim, E. Yang, and S. J. Hwang, "Uncertainty-aware attention for reliable interpretation and prediction," in *Advances in Neural Information Processing Systems*, pp. 909–918, 2018.

[13] M. W. Dusenberry, D. Tran, E. Choi, J. Kemp, J. Nixon, G. Jerfel, K. Heller, and A. M. Dai, "Analyzing the role of model uncertainty for electronic health records," *arXiv preprint arXiv:1906.03842*, 2019.

[14] Q. Tan, A. J. Ma, M. Ye, B. Yang, H. Deng, V. W.-S. Wong, Y.-K. Tse, T. C.-F. Yip, G. L.-H. Wong, J. Y.-L. Ching, *et al.*, "Ua-crnn: Uncertainty-aware convolutional recurrent neural network for mortality risk prediction," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 109–118, 2019.

[15] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, 2016.

[16] K. Häyrinen, K. Saranto, and P. Nykänen, "Definition, structure, content, use and impacts of electronic health records: a review of the research literature," *International journal of medical informatics*, vol. 77, no. 5, pp. 291–304, 2008.

[17] J. Henry, Y. Pylypchuk, T. Searcy, and V. Patel, "Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2015," *ONC Data Brief*, vol. 35, pp. 1–9, 2016.

[18] 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*, vol. 3, p. 160035, 2016.

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

[20] M. Reyna, C. Josef, R. Jeter, S. Shashikumar, M. Westover, S. Nemati, G. Clifford, and A. Sharma, "Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019," *Critical Care Medicine*, 2019.

[21] B. A. Goldstein, A. M. Navar, M. J. Pencina, and J. Ioannidis, "Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review," *Journal of the American Medical Informatics Association*, vol. 24, no. 1, pp. 198–208, 2017.

[22] E. Choi, A. Schuetz, W. F. Stewart, and J. Sun, "Using recurrent neural network models for early detection of heart failure onset," *Journal of the American Medical Informatics Association*, vol. 24, no. 2, pp. 361–370, 2016.

[23] N. Razavian and D. Sontag, "Temporal convolutional neural networks for diagnosis from lab tests," *arXiv preprint arXiv:1511.07938*, 2015.

[24] N. Razavian, J. Marcus, and D. Sontag, "Multi-task prediction of disease onsets from longitudinal lab tests," *arXiv preprint arXiv:1608.00647*, 2016.

[25] Z. Che, Y. Cheng, Z. Sun, and Y. Liu, "Exploiting convolutional neural network for risk prediction with medical feature embedding," *arXiv preprint arXiv:1701.07474*, 2017.

[26] Y. Cheng, F. Wang, P. Zhang, and J. Hu, "Risk prediction with electronic health records: A deep learning approach," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 432–440, SIAM, 2016.

[27] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[28] P. Nickerson, P. Tighe, B. Shickel, and P. Rashidi, "Deep neural network architectures for forecasting analgesic response," in *Conference proceedings:... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, vol. 2016, p. 2966, NIH Public Access, 2016.

[29] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor ai: Predicting clinical events via recurrent neural networks," in *Machine Learning for Healthcare Conference*, pp. 301–318, 2016.

[30] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Advances in Neural Information Processing Systems*, pp. 3504–3512, 2016.

[31] A. N. Jagannatha and H. Yu, "Bidirectional rnn for medical event detection in electronic health records," in *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, vol. 2016, p. 473, NIH Public Access, 2016.

[32] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International Conference on Machine Learning*, pp. 2342–2350, 2015.

[33] C. Esteban, O. Staeck, S. Baier, Y. Yang, and V. Tresp, "Predicting clinical events by combining static and dynamic information using recurrent neural networks," in *Healthcare Informatics (ICHI), 2016 IEEE International Conference on*, pp. 93–101, Ieee, 2016.

[34] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 328–339, 2018.

[35] M. Skinner, "Product categorization with lstms and balanced pooling views," in *SIGIR 2018 Workshop on eCommerce (ECOM 18)*, 2018.

[36] Synced, "2018 in review: 10 ai failures," Dec. 2018.

[37] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? does it matter?," *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[38] A. Siddhant and Z. C. Lipton, "Deep bayesian active learning for natural language processing: Results of a large-scale empirical study," *arXiv preprint arXiv:1808.05697*, 2018.

[39] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, pp. 1019–1027, 2016.

[40] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.

[41] G. Hinton and D. Van Camp, "Keeping neural networks simple by minimizing the description length of the weights," in *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*, Citeseer, 1993.

[42] R. M. Neal, "Bayesian learning via stochastic dynamics," in *Advances in neural information processing systems*, pp. 475–482, 1993.

[43] D. Barber and C. M. Bishop, "Ensemble learning for multi-layer networks," in *Advances in neural information processing systems*, pp. 395–401, 1998.

[44] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*, pp. 2348–2356, 2011.

[45] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *International Conference on Machine Learning*, pp. 1861–1869, 2015.

[46] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *arXiv preprint arXiv:1505.05424*, 2015.

[47] C. Steiner, R. Andrews, M. Barrett, and W. A., "Hcup projections: mobility/orthopedic procedures 2003 to 2012. 2012," *HCUP Projections Report*, no. 03, 2012.

[48] S. Matthew and N. P. Sheth, "Projected volume of primary and revision total joint arthroplasty in the united states, 2030-2060," in *2018 Annual Meeting of the American Academy of Orthopaedic Surgeons (AAOS)*, AAOS, 2018.

[49] "Spending on shoppable services in health care," in *Healthcare Cost Institute, Issue Brief 11, March 2016*, HCCI, 2016.

[50] D. A. Haas and K. Rober S, "Variation in the cost of care for primary total knee arthroplasties," *Arthroplasty Today*, vol. 3, no. 1, pp. 33–37, 2016.

[51] C. Xiao, T. Ma, A. B. Dieng, D. M. Blei, and F. Wang, "Readmission prediction via deep contextual embedding of clinical concepts," *PloS one*, vol. 13, no. 4, p. e0195024, 2018.

[52] C. Che, C. Xiao, J. Liang, B. Jin, J. Zho, and F. Wang, "An rnn architecture with dynamic temporal matching for personalized predictions of parkinson's disease," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 198–206, SIAM, 2017.

[53] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware lstm networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 65–74, ACM, 2017.

[54] 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.

[55] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[56] T. Ma, C. Xiao, and F. Wang, "Health-atm: a deep architecture for multifaceted patient health record representation and risk prediction," in *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 261–269, SIAM, 2018.

[57] N. Razavian, S. Blecker, A. M. Schmidt, A. Smith-McLallen, S. Nigam, and D. Sontag, "Population-level prediction of type 2 diabetes from claims data and analysis of risk factors," *Big Data*, vol. 3, no. 4, pp. 277–287, 2015.

[58] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[59] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[60] Y. Choi, C. Y.-I. Chiu, and D. Sontag, "Learning low-dimensional representations of medical concepts," *AMIA Summits on Translational Science Proceedings*, vol. 2016, p. 41, 2016.

[61] W. Farhan, Z. Wang, Y. Huang, S. Wang, F. Wang, and X. Jiang, "A predictive model for medical events based on contextual embedding of temporal sequences," *JMIR medical informatics*, vol. 4, no. 4, 2016.

[62] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *Journal of biomedical informatics*, vol. 83, pp. 112–134, 2018.

[63] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, ACM, 2008.

[64] 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*, vol. 37, no. 4-5, pp. 421–436, 2018.

[65] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

[66] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic gaussian process regression," in *Proceedings of the 22nd international conference on Machine learning*, pp. 489–496, ACM, 2005.

[67] F. H. Knight, *Risk, uncertainty and profit*. Courier Corporation, 2012.

[68] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

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

[70] R. Qiu, Y. Jia, F. Wang, P. Divakarmurthy, S. Vinod, B. Sabir, and M. Hadzikadic, "Predictive modeling of the total joint replacement surgery risk: a deep learning based approach with claims data," *AMIA Summits on Translational Science Proceedings*, vol. 2019, p. 562, 2019.

[71] W. K. Diprose, N. Buist, N. Hua, Q. Thurier, G. Shand, and R. Robinson, "Physician understanding, explainability, and trust in a hypothetical machine learning risk calculator," *Journal of the American Medical Informatics Association*, vol. 27, no. 4, pp. 592–600, 2020.