INPREM: An Interpretable and Trustworthy Predictive Model for Healthcare

Xianli Zhang
National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University
xlbryant@stu.xjtu.edu.cn

Buyue Qian
School of computer science and technology, Xi'an Jiaotong University
qianbuyue@xjtu.edu.cn

Shilei Cao
Tencent Jarvis Lab, Shenzhen, Guangdong 518075, China
eliasslcao@tencent.com

Yang Li
Key Laboratory of Intelligent Networks and Network Security (Xi'an Jiaotong University), Ministry of Education
vigilee@stu.xjtu.edu.cn

Hang Chen
School of computer science and technology, Xi'an Jiaotong University
cissy2898@stu.xjtu.edu.cn

Yefeng Zheng
Tencent Jarvis Lab, Shenzhen, Guangdong 518075, China
yefengzheng@tencent.com

Ian Davidson
Department of Computer Science, University of California, Davis, CA
davidson@cs.ucdavis.edu

ABSTRACT

Building a predictive model based on historical Electronic Health Records (EHRs) for personalized healthcare has become an active research area. Benefiting from the powerful ability of feature extraction, deep learning (DL) approaches have achieved promising performance in many clinical prediction tasks. However, due to the lack of interpretability and trustworthiness, it is difficult to apply DL in real clinical cases of decision making. To address this, in this paper, we propose an interpretable and trustworthy predictive model (INPREM) for healthcare. Firstly, INPREM is designed as a linear model for interpretability while encoding non-linear relationships into the learning weights for modeling the dependencies between and within each visit. This enables us to obtain the contribution matrix of the input variables, which is served as the evidence of the prediction result(s), and help physicians understand why the model gives such a prediction, thereby making the model more interpretable. Secondly, for trustworthiness, we place a random gate (which follows a Bernoulli distribution to turn on or off) over each weight of the model, as well as an additional branch to estimate data noises. With the help of the Monto Carlo sampling and an objective function accounting for data noises, the model can capture the uncertainty of each prediction. The captured uncertainty, in turn, allows physicians to know how confident the model is, thus making the model more trustworthy. We empirically demonstrate that the proposed INPREM outperforms existing approaches with a significant margin. A case study is also presented to show how the contribution matrix and the captured uncertainty are used to assist physicians in making robust decisions.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403087

CCS CONCEPTS

• **Applied computing** → **Health informatics**; • **Information systems** → **Data mining**.

KEYWORDS

Healthcare informatics; model uncertainty; model interpretability; attention mechanism

ACM Reference Format:
Xianli Zhang, Buyue Qian, Shilei Cao, Yang Li, Hang Chen, Yefeng Zheng, and Ian Davidson. 2020. INPREM: An Interpretable and Trustworthy Predictive Model for Healthcare. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403087

1 INTRODUCTION

Precision medicine typically relies on personalized estimates of outcome probabilities and treatment recommendations. Achieving this goal highly depends on accurate, meaningful, and reliable outcome prediction models. The broad adoption and immense accumulation of Electronic Health Records (EHRs) have opened the possibility of building such predictive models. However, the inherent issues of EHR data such as temporality, heterogeneity, high-dimensionality, and bias create obstacles to conventional data mining approaches. Contrastively, deep learning shares the superior ability in mining meaningful features from complicated data. Therefore, numerous state-of-the-art (SOTA) predictive models based on deep learning have been proposed for various clinical tasks [6, 22, 24].

Although these models achieved satisfactory performance, it is difficult for physicians to understand the output of these models, which creates obstacles for the transition from academic research to

clinical applications. A primary reason is that deep learning based models lack transparency, which can be further detailed as the lack of both **interpretability** and **trustworthiness** needed in clinical practice. These models suffer from two major issues: i) they are unable to tell physicians which medical events are most relevant to the output and ii) the models do not allow physicians to know how confident the predicted probability can be trusted. These two drawbacks of deep learning based models greatly undermine the reliability of their predictions, as well as prohibit them from being accepted in clinical practice. Consequently, it is crucial to develop an interpretable and trustworthy model for clinical prediction tasks.

In clinical routine, **model interpretability** requires the model being able to identify the contribution of each medical event to the corresponding prediction results. Recently, various interpretable predictive models have been proposed to facilitate decision makings, such as RETAIN [6] and Dipole [22], achieving reasonable performance. However, **model trustworthiness** was overlooked in their prediction results—they usually interpret the predicted probabilities at the end of the pipeline (*e.g.*, the output of softmax in classification setting) as the confidence scores on the model prediction, which have been proved to be erroneous in [11]. Conceptually, improving the model trustworthiness requires predictive models being able to represent uncertainty about the prediction. Nevertheless, standard deep learning tools are deterministic functions [9], which means that even if input with a randomly generated data point, the model would still output a deterministic result. From another perspective, without uncertainty at hand, we cannot know whether the model is making sensible predictions or just a random guess. For example, in the risk prediction setting, a model is likely to return a "low-risk" result with a probability score of 0.05 on a truly "high-risk" patient (red dash line in Fig. 1). Since the score is very close to zero, traditional models mistakenly treat it as a highly confident prediction, thus preventing the patient from early diagnosis. However, if the model provides an extra uncertainty estimation of the prediction with low confidence, this case may be then specially processed with an extra diagnosis, thereby a medical accident could be avoided.

Existing uncertainty in deep learning models could be roughly classified into two categories, *i.e.*, *aleatoric uncertainty* and *epistemic uncertainty* [18]. Formally, aleatoric uncertainty is usually caused by the noise inherent in the observations, including sensor noise, record error or missing value; whereas epistemic uncertainty (also referred to as model uncertainty) accounts for uncertainty over the model parameters. In this sense, the model trustworthiness can be obtained by estimating the epistemic uncertainty. It is worth noting that the former cannot be reduced by observing more data, but the latter can. There were several clinical predictive models [7, 8, 29] proposed to provide either *aleatoric uncertainty* or *epistemic uncertainty* estimation alone, and our work novelly proposes to model both.

To tackle all the aforementioned limitations, in this paper, we develop an **in**ter**p**retable and t**r**ustworthy pr**e**dictive **m**odel namely INPREM for healthcare prediction applications. The proposed INPREM can not only identify the contribution of each medical event to the predictions but also capture two types of uncertainty (*i.e.*, *aleatoric uncertainty* and *epistemic uncertainty*) within a single framework. Specifically, the INPREM is designed as a linear model for interpretability while encoding the non-linear relationships into the learning weights for capturing the intricate dependencies between and within each visit. The learning weights are implemented with a visit attention module for modeling dependencies between visits and a variable attention module for modeling dependencies within each visit. The contribution matrix can be obtained with the attention matrices due to the whole linearity of the model. Based on the interpretable model, we then place a random gate (following a Bernoulli distribution) over each weight of the built network for epistemic uncertainty estimation (trustworthiness), which extends the model to be a Bayesian Neural Network (BNN). Afterward, an additional branch is attached to the end of the pipeline to estimate the noises of each data point (aleatoric uncertainty). With the help of the Monto Carlo sampling and an objective function accounting for the noises inherent in data, the proposed model can capture both aleatoric uncertainty and epistemic uncertainty, meanwhile maintaining high-precision and a low computation cost.

With the experiments conducted in diagnosis prediction and disease risk prediction tasks, we demonstrate that the proposed INPREM achieves significant performance improvement upon other SOTA methods. In summary, the main contributions of this paper are as follows:

- We propose INPREM, an end to end, novel, and robust model to predict future health conditions for patients, which can not only output the probability of the outcome but also provide evidence and confidence in assisting the prediction.
- We empirically show that the proposed INPREM outperforms existing approaches with a significant margin, thanks to the well-designed model structure and the uncertainty modeling.
- By visualizing and analyzing the contribution matrix of the input and the probability distribution derived from Monto Carlo sampling, we demonstrate that the extra information provided by INPREM could help physicians make more robust decisions.

2 RELATED WORK

Deep Learning for Mining EHRs: EHRs contain rich historical health information about patients. Mining useful features from EHRs to build predictive models for personalized healthcare is a promising application. Recently, deep learning approaches, especially Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), have been successfully applied to mining the complicated EHR data. Attributed to the excellent capabilities in temporal data modeling, RNNs were naturally used to model sequential EHRs for a variety of healthcare applications, including risk prediction [6, 38], diagnosis prediction [4, 22], mortality prediction [12], forecasting length of stay (LOS) [13], *etc.* Compared to RNNs, CNNs focused more on the local dependence among EHRs, which were applied to predict disease risk [2, 3] or other future status of patients [34]. More recently, researchers exploited the effect of attention mechanisms and posterior regularization in incorporating domain knowledge to guide deep models making accurate predictions [5, 21, 23, 24, 39]. Besides, other advanced deep learning model structures, such as Transformer [37], were also studied for mining EHRs [32]. Compared to these works, our work innovatively provides prediction power with not only interpretability but also trustworthiness, both promoting the application in clinical practice.

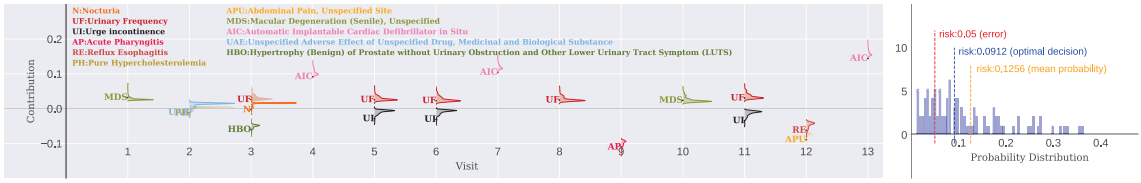

Figure 1: A case study of the proposed INPREM on the risk prediction task (illustrated with heart failure). The left part visualized the contribution matrix of medical events (y-axis) along with different visits (x-axis). Note, we show each contribution item with a distribution. The right part is the probability distribution derived from the learned variational parameters distribution $q^*(w)$ (to be detailed in Section 3.3). The red dash line is a wrong and over-confident prediction from a deterministic model, the blue dash line is the optimal decision threshold (with recall > 0.8), and the orange dash line is the mean value of the probability distribution.

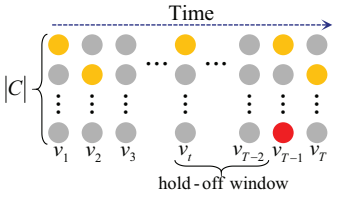

Figure 2: An illustration of a patient's EHR data. The EHR data of one specific patient consist of a sequence of visits $v_1, v_2, \cdots, v_T$. Each visit contains a subset of medical codes, which could be represented by a binary vector $v_t \in \{0,1\}^{|C|}$, where the i-th element is set to 1 (the orange point) if the t-th visit contains the medical code c_i, otherwise 0 (the gray point). The red point denotes the medical code of the target disease for the risk prediction task.

It is worth noting that the interpretability was also formulated in RETAIN [6], where a two-level attention mechanism was leveraged to detect influential past visits and significant clinical variables within those visits. By rethinking RETAIN, our work, on the one hand, enhances the interpretability with a sparsification technique [26] (will be introduced in Section 3.2) to attention weights; on the other hand, it provides uncertainty estimation assisting the trustworthy prediction.

Uncertainty Modeling: Capturing uncertainty is an indispensable part of many applications with deep learning, especially in medical prediction scenarios. There is a great demand for uncertainty estimation in a forecasting model. As a powerful tool in statistics, the Gaussian process has been applied in many tasks to capture uncertainty by modeling distributions over functions [27, 30, 35]. It has been widely recognized that a neural network with its weights treated as random variables could be regarded as a BNN [25]. As BNNs are notoriously hard for inference, many variational inference techniques [16] are proposed to address this challenge, such as stochastic variational inference and sampling-based variational inference [28, 31, 36]. More recently, Lakshminarayanan *et al.* [20] proposed an alternative to BNN-based methods named Deep Ensemble to quantify model uncertainty. However, all above methods come with huge computational costs. Gal *et al.* [33] demonstrated that the use of dropout in deep neural networks could be regarded as an approximate Gaussian process. Our work adopts dropout to provide uncertainty estimation due to its low computation cost and high-efficiency.

It is worth noting that the work of Heo *et al.* [15] was with a similar motivation of uncertainty modeling and interpretability to our method, but with a different emphasis and task. They focused more on the input-dependent uncertainty in attention-level for generating attention for each feature with varying degrees of noise based on the given input; whereas our model concentrates on the output-level uncertainty, which makes the output logit of each example be with varying degrees of noise based on itself. Besides, the accuracy performance gains of our model are not only attributed to the uncertainty modeling but also a more efficient network design with the support of interpretability.

3 METHOD

3.1 Basic Notation & Problem Defination

Before a detailed description, we summarize the notations used in this paper. We treat the medical events taking place in EHR as medical codes, which are denoted as $c_1, c_2, \cdots, c_{|C|} \in C$, where $|C|$ is the total number of unique medical codes. As illustrated in Fig. 2, the EHR data of one specific patient consist of a sequence of visits $v_1, v_2, \cdots, v_T$, where we denote the number of visits in total as T. Each visit contains a subset of medical codes, and we denote each visit as a binary vector $v_t \in \{0,1\}^{|C|}$, where the i-th element is set to 1 if the t-th visit contains the medical code c_i, otherwise 0. The visits $v_1, v_2, \cdots, v_T$ are stacked to form an input matrix $\mathbf{X} \in \{0,1\}^{|C| \times T}$, which we use as the input for the network.

Based on the notations, we introduce the problems of diagnosis prediction and disease risk prediction as follows:

Diagnosis Prediction is also referred as Encounter Sequence Modeling (ESM) [4]. Specifically, given a sequence of visits $v_1, v_2, \cdots, v_T$, the goal of this task is to predict the medical codes occurring at next visit v_{t+1}. In this sense, this task can also be regarded as a multi-label classification problem.

Disease Risk Prediction can be seen as a special case of ESM where only one disease outcome (the red point in Fig. 2) is predicted for binary classification. Different from diagnosis prediction, another particular setting of this task is that the visit sequence must be given before a hold-off window (as illustrated in Fig. 2) to account for its clinical significance.

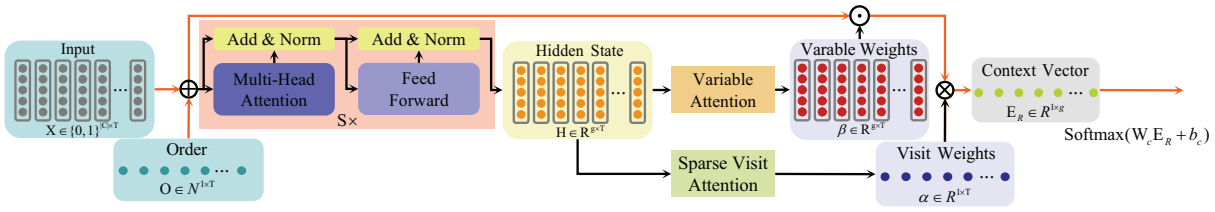

Figure 3: An overview architecture of the proposed INPREM. The INPREM is designed as a linear model for interpretability while encoding non-linear relationships into the learnable weights (see Eq. (3) and (4) for ease of understanding). Specifically, the INPREM encodes the relationship between the prediction target to each of the medical events with a linear part (orange line) and model the dependencies between and within each visit with a non-linear part (black line) for weighting the linear relationship.

3.2 Basic Framework for Interpretability

We build our framework with a linear-part and a non-linear part. To make our model interpretable, we propose to estimate the relationship between the prediction target to each of the medical events with a linear part, which facilitates us to account for the contribution of each medical event to the decision. Since different visits usually show implicit dependencies with regular patterns, a non-linear part is proposed to capture such dependencies. An overview architecture of the proposed INPREM is shown in Fig. 3.

3.2.1 A Linear Model for Interpretability. Given the input visit sequence $\mathbf{X}$, we employ an embedding layer to learn the representation of each visit, which models the relationships between different medical codes within each visit:

$$\mathbf{E}_v = \mathbf{W}_v \mathbf{X}, \tag{1}$$

where $\mathbf{W}_v \in \mathbb{R}^{g \times |C|}$ is the parameters to learn, and $\mathbf{E}_v \in \mathbb{R}^{g \times T}$ is the learned visit embedding. Here, g ($g = 256$ in our experiments) is the dimension of the embedding space. Since the stacked multi-head self-attention (to be detailed later) contains no recurrence thus losing the order information of each visit, we similarly employ an extra embedding layer to encodes such order information:

$$\mathbf{E}_o = \mathbf{W}_o \mathbf{O}, \tag{2}$$

where $\mathbf{W}_o \in \mathbb{R}^{g \times 1}$ is the parameters to learn, $\mathbf{O} \in \mathbb{N}^{1 \times T}$ denotes the orders of each visit in time, and $\mathbf{E}_o \in \mathbb{R}^{g \times T}$ is the order embedding.

To develop a predictive model, a patient-level representation $\mathbf{E}_R \in \mathbb{R}^{1 \times g}$ needs to be obtained from the input embeddings first. To this end, we encode the relations between the patient representation ($\mathbf{E}_R$) and input embeddings ($\mathbf{E}_v$ and $\mathbf{E}_o$) with a linear mapping:

$$\mathbf{E}_R = \alpha\big(\beta \odot (\mathbf{E}_v + \mathbf{E}_o)\big)^\top, \tag{3}$$

where $\alpha \in \mathbb{R}^{1 \times T}$ encodes the non-linear dependencies between the visits, $\beta \in \mathbb{R}^{g \times T}$ encodes the non-linear dependencies between medical events within each visit, $\odot$ denotes element-wise multiplication, and the visit and order embeddings are fused with a simple addition operation. The implementation of α and β will be described later.

With the patient representation, the logit serving for prediction can be easily computed as:

$$\tilde{y} = \mathbf{W}_c^\top \mathbf{E}_R^\top + b_c, \tag{4}$$

where $\mathbf{W}_c \in \mathbb{R}^{g \times l}$ and $b_c \in \mathbb{R}^{l \times 1}$ are model parameters to learn, and the value of l (the number of classes to predict) depends on the task. Then, the Softmax$(\cdot)$ is used to estimate the probability y^* for prediction:

$$y^* = \text{Softmax}(\tilde{y}). \tag{5}$$

Note, the risk prediction is naturally a binary classification, while the diagnosis prediction can be treated as multiple binary classifications, since, it is a multi-label classification as mentioned above.

Interpretability for the Prediction: Thanks to the linearity of the model, we can easily calculate the contribution of each medical event by inferring from the predicted $\hat{y}$ back to the input $\mathbf{X}$. According to Eqs. (1), (2), (3), (4), and (5), we can get:

$$\begin{aligned} y^* &= \text{Softmax}(\mathbf{W}_c^\top \mathbf{E}_R^\top + b_c) \\ &= \text{Softmax}\Big(\mathbf{W}_c^\top \sum_{i=1}^{T} \sum_{j=1}^{|C|} \alpha[i]\beta[:,i] \odot (v_i[j]\mathbf{W}_v[:,j] + i\mathbf{W}_o) + b_c\Big). \end{aligned} \tag{6}$$

The contribution of each medical event is thus calculated as follows:

$$\text{CM}[i,j] = \mathbf{W}_c^\top \left(\alpha[i]\, \beta[:,i] \odot \mathbf{W}_v[:,j]\right), \tag{7}$$

where $\text{CM} \in \mathbb{R}^{T \times |C| \times l}$ is the contribution matrix, and we use $\text{CM}[i,j][k]$ to denote the contribution of the j-th medical event in the i-th visit to the prediction when the predicted class is k.

3.2.2 Non-linear Part for Modeling Dependencies. As different medical events usually happen in a sequence with specific patterns and different visits reflect the different conditions of the patient, the linear model suffers from insufficient ability in capturing such dependencies between and within each visit, which, however, are critical information for the prediction. We propose to encode the non-linearity into α and β in Eq. (3). To this end, we first employ stacked multi-head attention to strengthen the deep semantics in medical events and visits, then output a hidden state with strong representation power. After that, a sparse visit attention module and a variable attention module are proposed to weigh the importance of different visits and different medical events within each visit, respectively. We describe them next.

Stacked Multi-head Attention: The multi-head attention is formed by multiple self-attention layers running in parallel for enriching the representation of each visit. A self-attention layer is fed with a set of key-query pairs, as well as corresponding values. The key-query pairs are used to compute the inner dependency weights, which are then used to update the values. Mathematically,

the self-attention could be formalized as follow:

$$\text{Att}(Q,K,V) = V\big(\text{Softmax}(\frac{Q^\top K}{\sqrt{d_k}})\big),$$
$$Q = \mathbf{W}_1(\mathbf{E}_v + \mathbf{E}_o); K = \mathbf{W}_2(\mathbf{E}_v + \mathbf{E}_o); \text{and } V = \mathbf{W}_3(\mathbf{E}_v + \mathbf{E}_o), \quad (8)$$

where $Q, K \in \mathbb{R}^{d_k \times T}$, $V \in \mathbb{R}^{d_v \times T}$ and $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d_k \times g}$,$\mathbf{W}_3 \in \mathbb{R}^{d_v \times g}$ are corresponding learnable weights on the addition of the input embeddings $\mathbf{E}_v$ and $\mathbf{E}_o$ for outputing Q, K, and V, respectively. Note, we set $d_k = d_v = 256$ in this paper. The multi-head attention concatenates multiple individual self-attention and fuses all the subspace information by a fully-connected layer:

$$\text{MultiHeadAtt}(Q,K,V) = \mathbf{W}_o \text{Concat}(\text{Att}_1, \text{Att}_2, \cdots, \text{Att}_m), \quad (9)$$

where $\mathbf{W}_o \in \mathbb{R}^{d_{model} \times m d_v}$ is the parameter to learn, and m is a hyperparameter that indicates the number of heads. Similar to the Transformer [37], each multi-head self-attention is followed by a feed-forward layer, which consists of two 1D convolutional layers with kernel size 1 and ReLU activation. Besides, the dimensionality of the input and output of the feed-forward layer is indicated by d_{model}, while the inner-layer has d_{inner} channels. We employ a residual connection [14] around the multi-head attention and the feed-forward layer followed by layer normalization [1].

To strengthen the semantics, the multi-head attention module is stacked S times for outputing the hidden state $\mathbf{H} \in \mathbb{R}^{d_{model} \times T}$ with strong ability in representation.

Sparse Visit Attention Module: In the real diagnosis scenarios, physicians typically put different weights on different visits in the diagnosis process. In this sense, we propose a visit-level attention mechanism to emphasize important visits. Besides, under the requirements of clinical practice for interpretability, the attention weights should be sparse, thus, the most important visits can be highlighted. Based on the prior knowledge of diagnosing habits and the requirements of model interpretability, we propose a sparse visit attention module to guide the model to focus on the visits containing important features. We augment the Softmax$(\cdot)$ with a Sparsemax$(\cdot)$ [26] for pursuing a sparse attention weight. Specifically, we first compute a correlation vector $\delta \in \mathbb{R}^{1 \times T}$ from the hidden state, and then the visit attention weight $\alpha \in \mathbb{R}^{1 \times T}$ can be obtained by applying a combination of Sparsemax$(\cdot)$ and Softmax$(\cdot)$ which is formalized as follows:

$$\delta = \mathbf{W}_\delta \mathbf{H} + b_\delta,$$
$$\alpha = (\text{Sparsemax}(\delta) + \text{Softmax}(\delta))/2, \quad (10)$$

where $\mathbf{W}_\delta \in \mathbb{R}^{1 \times d_{model}}$ and $b_\delta \in \mathbb{R}^{1 \times T}$ are the parameters to learn.

Variable Attention Module: The variable attention module is designed to enforce the model paying attention to the important features within a single visit. The idea is similar to RETAIN [6], yet implemented more efficiently. To be specific, we share the same hidden state with the visit and variable attention modules to save computation cost, which leads to a similar performance compared to equipping with a separate hidden state for each. The variable attention weight $\beta \in \mathbb{R}^{g \times T}$ takes the dependencies among different medical events within each visit into consideration, which is formalized as follows:

$$\beta = \tanh(\mathbf{W}_\beta \mathbf{H} + b_\beta) \quad (11)$$

where $\mathbf{W}_\beta \in \mathbb{R}^{g \times d_{model}}$ and $b_\beta \in \mathbb{R}^{g \times 1}$ are the parameters to learn.

3.3 Extension to BNN for Trustworthiness

BNNs implement the Bayesian probability theory with neural networks (NNs) which are usually treated as a tool for capturing epistemic uncertainty. The core idea of BNNs is to place a prior distribution $p(f)$ over the space of a function f, and then search for the posterior distribution $p(f|\mathbf{D_X}, \mathbf{D_Y})$ over function space of f given the dataset $(\mathbf{D_X}, \mathbf{D_Y})$. In practical, the computation of the posterior distribution with NNs is usually intractable. Researchers resort to approximation to address this problem. Specifically, they condition the model on a finite set of random variables $\mathbf{w}$ based on the assumption that the functions depend on these variables alone [10]. Given a new data point $\mathbf{x}^*$, the prediction can be obtained by integrating over all plausible parameters $\mathbf{w}$ as:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{D_X}, \mathbf{D_Y}) = \int p(\mathbf{y}^*|f^*)p(f^*|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathbf{D_X}, \mathbf{D_Y})\mathrm{d}f^*\mathrm{d}\mathbf{w}. \quad (12)$$

However, the distribution $p(\mathbf{w}|\mathbf{D_X}, \mathbf{D_Y})$ cannot be evaluated analytically as well. The variational inference is raised in this situation to define an approximating variational distribution $q(\mathbf{w})$, with a regularzation that minimizes the Kullback-Leibler (KL) divergence to make $q(\mathbf{w})$ as close as posible to $p(\mathbf{w}|\mathbf{D_X}, \mathbf{D_Y})$. In this way, Eq. (12) can be rewritten as:

$$q(\mathbf{y}^*|\mathbf{x}^*) = \int p(\mathbf{y}^*|f^*)p(f^*|\mathbf{x}^*, \mathbf{w})q(\mathbf{w})\mathrm{d}f^*\mathrm{d}\mathbf{w}. \quad (13)$$

The objective function of variational inference $\mathcal{L}_{vi}$, in this sense, is equivalent to maximizing the log evidence lower bound:

$$\mathcal{L}_{vi} = \int q(\mathbf{w})p(f|\mathbf{D_X}, \mathbf{w})\log p(\mathbf{D_Y}|f)\mathrm{d}f\mathrm{d}\mathbf{w} - \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})). \quad (14)$$

In this paper, we adopt the Bernoulli distribution to approximate the variational distributions. Specifically, we place a random gate (which follows the Bernoulli distribution to turn on or off) over the weights of each layer in the framework mentioned above. Mathematically, we define the variational distribution $q(\mathbf{W}_k)$ for each layer k as:

$$\mathbf{W}_k = \mathbf{B}_k \odot [\mathrm{u}_{k,j}]_{j=1}^{Z_k}, \mathrm{u}_{k,j} \sim \text{Bernoulli}(p_k), \quad (15)$$

where $p_k \in (0,1)$ is the probability in Bernoulli, $\mathbf{B}_k$ denotes variational parameters of the model, and Z_k indicates the number of elements in the k-th layer. For the computation of $\mathcal{L}_{vi}$, we employ Monte Carlo integration to obtain an unbiased estimator $\hat{\mathcal{L}_{vi}}$ of Eq. (14), which makes the integral tractable:

$$\hat{\mathcal{L}_{vi}} = \sum_{i=1}^{N} E(\mathbf{y}_i, f(\mathbf{x}_i, \hat{\mathbf{w}}_i)) - \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})), \quad \hat{\mathbf{w}}_i \sim q(\mathbf{w}), \quad (16)$$

where $E(\cdot)$ is the likelihood function, whose implementation depends on tasks, $(\mathbf{x}_i, \mathbf{y}_i)$ is a data pair in the dataset $(\mathbf{D_X}, \mathbf{D_Y})$, and N is the size of the dataset. Now, we can replace the variational distribution $q(\mathbf{w})$ in Eq. (14) with the Bernoulli approximating variational distribution $q(\mathbf{W}_k)$. Besides, the KL divergence in Eq. (16) can be approximated in the way as [11], which is implemented by

an L_2 regularization with a balancing weight of λ:

$$\tilde{y}_i = f(\mathbf{x}_i, q(\mathbf{w})), \quad q(\mathbf{w}) = \{\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_k\}$$
$$\mathcal{L} = \sum_{i=1}^{N} E(\mathbf{y}_i, \tilde{y}_i) + \lambda \sum_{k=1}^{L} (\|\mathbf{W}_k\|_2^2 + \|\mathbf{b}_k\|_2^2), \tag{17}$$

where $\mathbf{W}_k, \mathbf{b}_k$ are weights and bias of the k-th layer of our framework, and L is the number of layers in the network.

Placing the Bernoulli distribution over the parameters of the model is only able to capture epistemic uncertainty. To further capture aleatoric uncertainty, we adopt a Gaussian distribution over the output logit $\tilde{y}$ in our framework. Specifically, we predict an extra output for regressing the variance of σ of the Gaussian distribution. Then, we use the Gaussian noise to corrupt the logit $\tilde{y}$ to output a logit $\hat{y}$, which shares the ability in capturing aleatoric uncertainty:

$$\hat{y} = \tilde{y} + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{18}$$

Similar to the calculation of the integral mentioned above, we approximate the Gaussian distribution with Monto Carlo sampling. Since only $\hat{y}$ is sampled from $\tilde{y}$, the test time will not be dramatically increased.

Formulation for Risk Prediction: The objective function (17) for the risk prediction task can be rewriten as follows:

$$\hat{y}_t = \tilde{y} + \sigma\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I),$$
$$\mathcal{L}_{risk} = \log \frac{1}{T_{mc}} \sum_{i=1}^{N} \sum_{t=1}^{T_{mc}} \exp(\hat{y}_{i,t} - \log \sum_{j=1}^{l} \exp(\hat{y}_{i,t,j}))$$
$$+ \lambda \sum_{k=1}^{L} (\|\mathbf{W}_k\|_2^2 + \|\mathbf{b}_k\|_2^2), \tag{19}$$

where i is the index of data points, T_{mc} denotes the number of times for Monto Carlo sampling with t indexing each sampling, and $l = 2$ denotes the number of risk labels (*i.e.*, 0, 1 in this case).

Formulation for Diagnosis Prediction: As aforementioned, the diagnosis prediction task can be seen as a multi-label classification problem, which can be further implemented with multiple binary classifications (thus, l also equals to 2). To make aleatoric uncertainty depend on the data rather than the task, we share the variance σ of the Gaussian distribution among all the binary classification tasks for each of data points. The objective function of the diagnosis prediction task can be formulated as follows:

$$\hat{y}_{l,t} = \tilde{y}_l + \sigma\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I),$$
$$\mathcal{L}_{esm} = \log \frac{1}{T_{mc}} \sum_{i=1}^{N} \sum_{c}^{C} \sum_{t=1}^{T_{mc}} \exp(\hat{y}_{i,c,t} - \log \sum_{j=1}^{l} \exp(\hat{y}_{i,c,t,j}))$$
$$+ \lambda \sum_{k=1}^{L} (\|\mathbf{W}_k\|_2^2 + \|\mathbf{b}_k\|_2^2). \tag{20}$$

In the test phase, we perform Monto Carlo integration over the learned variational distribution $q^*(\mathbf{w})$ with a Softmax$(\cdot)$ activation to estimate the prediction:

$$p(y = l|\mathbf{x}, \mathbf{D_X}, \mathbf{D_Y}) \approx \frac{1}{T_{test}} \sum_{t=1}^{T_{test}} \text{Softmax}(f^{\hat{\mathbf{w}}_t}(\mathbf{x}) + \sigma_t\epsilon),$$
$$\hat{\mathbf{w}}_t \sim q^*(\mathbf{w}), \quad \epsilon \sim \mathcal{N}(0, I), \tag{21}$$

where T_{test} denotes the number of times for Monto Carlo sampling with t indexing each sampling.

In the experiment parts, we qualitatively evaluate the ability of INPREM in capturing epistemic uncertainty, which assists physicians in making trustworthy decisions. Since the robustness of INPREM to data noise is not the focus of this work, we do not present experiments results on aleatoric uncertainty. However, we present quantitative indicators of both the aleatoric uncertainty and epistemic uncertainty in Appendix 6.4 for clinical usage.

4 EXPERIMENTS

In order to evaluate the effectiveness of the proposed **INPREM**, we conduct experiments on two datasets, including the publicly available MIMIC-III [17],[1] and a real-world longitudinal EHR database with three cohorts, including Heart Failure, Diabetes, and Chronic Kidney Disease (CKD). See Appendix 6.1 for details of all datasets. Moreover, several analytical experiments and a case study are presented to show the interpretability and trustworthiness of our model.

4.1 Baselines

In order to fairly evaluate the effectiveness of the proposed INPREM, we compare it with several SOTA methods, including CNN, RNN, RNN with attention mechanism, RETAIN [6], and Dipole [22]. We give a detailed description of all these methods in Appendix 6.2 due to the limitation of lengths. Besides, in ablation experiments, we also present three variants of INPLIM as follows:

INPREM$_{b-}$: A variant of the proposed INPREM that does not perform Bernoulli approximating variational inference.

INPREM$_{s-}$: This version of the INPREM employs the standard Softmax$(\cdot)$ instead of combining Sparsemax$(\cdot)$ in the visit attention module.

INPREM$_{o-}$: In this variant of the INPREM, we remove the order embeddings entirely.

The implement details of all baselines and our methods are presented in Appendix 6.3.

4.2 Evaluation Metrics

Diagnosis Prediction: For this task, we use the same metrics proposed in [24], which are *visit-level precision@k* and *code-level accuracy@k*. The *visit-level precision@k* is defined as the number of correct medical codes among the ranked top k predictions divided by $\min(k, |\mathrm{y}_t|)$, where $|\mathrm{y}_t|$ is the number of category labels appeared in the $(t+1)$-th visit. We report the average *visit-level precision@k* of all visits. The *code-level accuracy@k* measures the overall accuracy of predictions, which is defined as the number of correctly predicted codes divided by the total number of predicted codes among the ranked top k predictions.

Risk Prediction: The positive/negative labels in the dataset of this task are usually imbalanced. Therefore, we use Area Under Receiver Operator Curve (AU-ROC) to measure the performance of all approaches for this task.

[1] https://mimic.physionet.org/

Table 1: Results of Diagnosis Prediction Task on the MIMIC-III dataset [17]

	Model	Code-Level Accuracy@k						Visit-Level Precision@k					
		5	10	15	20	25	30	5	10	15	20	25	30
Baselines	CNN	0.6399	0.5840	0.6267	0.6984	0.7626	0.8160	0.3026	0.4824	0.6025	0.6921	0.7590	0.8140
	RNN	0.6213	0.5686	0.6196	0.6905	0.7550	0.8069	0.2938	0.4700	0.5947	0.6841	0.7541	0.8082
	RNN+	0.6214	0.5672	0.6147	0.6884	0.7559	0.8070	0.2938	0.4686	0.5903	0.6821	0.7534	0.8079
	RETAIN	0.6284	0.5760	0.6318	0.7018	0.7687	0.8212	0.2959	0.4758	0.5974	0.6855	0.7584	0.8137
	Dipole	0.6325	0.5758	0.6203	0.6921	0.7571	0.8083	0.2986	0.4746	0.5950	0.6841	0.7535	0.8083
INPREM	INPREM	0.6886	0.6247	0.6625	**0.7306**	0.7878	**0.8314**	**0.3204**	**0.4992**	0.6179	0.7080	0.7728	**0.8199**
	INPREM$_{b-}$	0.6796	0.6152	0.6593	0.7269	0.7848	0.8290	0.3175	0.4892	0.6162	0.6973	0.7706	0.8185
	INPREM$_{o-}$	0.6891	**0.6253**	0.6626	0.7306	**0.7890**	0.8308	0.3210	0.4991	**0.6189**	**0.7082**	**0.7733**	0.8193
	INPREM$_{s-}$	**0.6902**	0.6241	**0.6626**	0.7302	0.7881	0.8307	0.3204	0.4985	0.6184	0.7081	0.7725	0.8193

Table 2: AUROC of the Risk Prediction Task

	Model	Heart Failure	Diabetes	CKD
Baselines	CNN	0.7194	0.6490	0.7478
	RNN	0.7243	0.6587	0.7472
	RNN+	0.7228	0.6483	0.7464
	RETAIN	0.7312	0.6596	0.7508
	Dipole	0.7333	0.6608	0.7537
INPREM	INPREM	**0.7590**	0.6757	**0.7745**
	INPREM$_{b-}$	0.7525	0.6688	0.7706
	INPREM$_{o-}$	0.6991	0.6270	0.7366
	INPREM$_{s-}$	0.7589	**0.6758**	0.7744

4.3 Performance Analysis

Table 1 reports the *visit-level precision@k* and *code-level accuracy@k* of both INPREM and baselines on the diagnosis prediction task. We can observe that the proposed INPREM outperforms all baselines on both evaluation metrics. Specifically, the code-level accuracy improves 5.88% and the visit-level accuracy improves 7.86% at $k = 5$ when comparing the best results of our method with the best results of baseline methods. Another observation is that even though the INPREM$_{b-}$ has a better performance than all baselines on all values of k, it still performs worse than INPREM. This proves that both the designed model structure and uncertainty modeling are effective to improve the prediction performance. Comparing INPREM with INPREM$_{o-}$, we find that the proposed order embedding module damages the performance on the diagnosis prediction task in some values of k. The reason may be that the number of visits is relatively small (average 2.67 as shown in Table 4 in the Appendix) in the MIMIC-III dataset [17], on which the effect of the order information does not bring into play.

Table 2 lists the AUROC of INPREM and baselines on the risk prediction task. Comparing with all baselines, we find that the proposed INPREM achieves the best performance. Specifically, the AUROC improves 3.50% on the Heart Failure cohort, 2.26% on the Diabetes cohort, and 2.75% on the Chronic Kidney Disease cohort when comparing the best results of our method with the best results of baseline methods. An important observation is that INPREM$_{o-}$ performs worse than any other methods, including other variants of INPREM and all baselines methods. This is because the length of the sequence in the datasets of risk prediction task is long (average visit number of a patient is 15.03, 16.92, and 16.44 in the three cohorts, respectively, see Table 5 in the Appendix). This may be that the order embedding module is very important when the sequence is long. Similar to the diagnosis prediction task, the INPREM has a better performance than INPREM$_{b-}$ because of modeling uncertainty.

We also find that INPREM and INPREM$_{s-}$ achieve similar performance on both tasks. However, INPREM has better interpretability than INPREM$_{s-}$, which will be discussed in the next subsection. In summary, benefiting from the well-designed model structure and uncertainty modeling, the proposed INPREM achieves the best performance compared with all the SOTA methods.

4.4 Interpretability Analysis

Table 3 lists the top-10 medical codes that are most relevant to heart failure in the test set of the Heart Failure cohort. Then, a group of cardiologists is invited to verify these medical events. As expected, all medical events in Table 3 are able to accompany or increase the risk of heart failure.

In order to evaluate the sparsity of the proposed sparse visit attention module, we visualize the first 16 dimensions of $\alpha\beta^\top$, which are shown in Fig. 4. The left heatmap of each example is the attention weights without Sparsemax$(\cdot)$, while the right one is that being applied with Sparsemax$(\cdot)$. We can observe that in each example, the right one is sparser than the left one. We find that similar features are highlighted when equipping with or without Sparsemax$(\cdot)$, yet the sparse attention weights are able to provide more distinct discrimination. This means the proposed sparse visit attention module is able to effectively enhance the interpretability of the model.

4.5 Uncertainty Analysis

In order to prove that the INPREM is capable of capturing epistemic uncertainty, we visualize the probability distribution of the prediction derived from the learned variational parameter distribution $q^*(\mathbf{w})$ and that of 200 INPREM$_{b-}$ ensembles, which are shown

Table 3: The Top 10 Medical Events with the Highest Contribution to the Heart Failure in the Test Set

ICD-9	Code Description
V43.3	Heart Valve Replaced by Other Means
V42.2	Heart Valve Replaced by Transplant
424.0	Mitral Valve Disorders
V12.50	Personal History of Unspecified Circulatory Disease
V45.02	Automatic Implantable Cardiac Defibrillator In Situ
412	Old Myocardial Infarction
571.5	Cirrhosis of Liver Without Mention of Alcohol
410.10	Acute Myocardial Infarction of Other Anterior Wall, Episode of Care Unspecified
426.3	Other Left Bundle Branch Block
425.4	Other Primary Cardiomyopathies

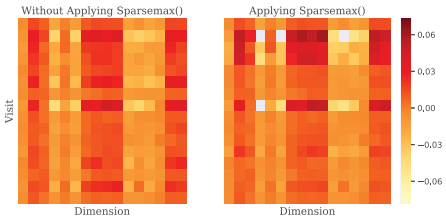

Figure 4: Heatmaps of attention weights $\alpha\beta^{\top}[:16,:16]$. The left one is the attention weights without applying Sparsemax$(\cdot)$, while the right one is with Sparsemax$(\cdot)$ applied. The white blocks are values which are lower than the minimum weight in the left heatmap.

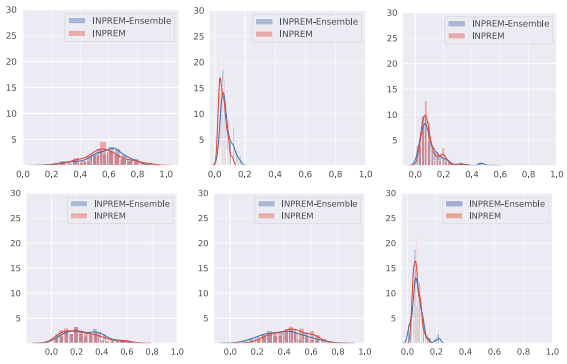

Figure 5: Predicted probability distribution of 200 $\text{INPREM}_{b\_}$ (non-Bayesian version of the original INPREM) ensembles and that derived from the learned variational parameters distribution $q^*(w)$ of INPREM. We can find that the proposed INPREM is able to capture uncertainty similar to that of ensembles.

in Fig. 5. We find that the proposed INPREM is able to capture uncertainty similar to that of ensembles.

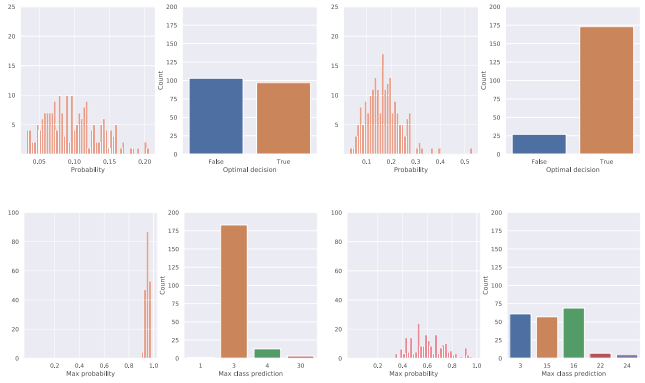

Figure 6: Top row shows two predicted probability distributions on the risk prediction task. Bottom row shows two examples of the predicted max probability distribution on the diagnosis prediction task. These two sets of examples proved that the captured uncertainty of our model is able to help physicians in clinical decision-making.

We are also interested in if the captured epistemic uncertainty could assist physicians in decision-making. Therefore, following the analytical method in [8], we first determine an optimal decision threshold (shown with the blue dash line in right part of Fig. 1) on the validation set of the risk prediction task for heart failure. The optimal decision threshold requires the recall of positive samples higher than 80%. Subsequently, two examples are visualized in the top row of Fig. 6. We find that the predicted results with high uncertainty lead to confusing decisions, which is in accordance with clinical practice. Another set of examples on the diagnosis prediction is also visualized in the bottom row of Fig. 6, which shows a similar situation.

4.6 Case Study

In order to further show the transparency of the proposed INPREM, we visualize the result of a case study (illustrated in Fig. 1) on the test set of the Heart Failure cohort. The left part of Fig. 1 shows that the *Automatic Implantable Cardiac Defibrillator In Situ (AIC)* is the most relevant event to the heart failure, while *Acute Pharyngitis (AP)* shows a negative value of contribution, which means that the AP has no apparent correlations to the target disease. We also find that the distributions of each contribution item in contribution matrix are similar among the same events, but distinct among different events.

In the right part of Fig. 1, we show the predicting probability distribution. We find that a deterministic model is quite possible to output a risk lower than the optimal decision threshold. Benefiting from both the unbiased estimation of the probability distribution and the captured uncertainty, our model may increase the diagnosis accuracy to some extent, meanwhile, arouse the attention of physicians to the case with high uncertainty.

This case study demonstrates that the extra information provided by INPREM not only helps physicians in decision-making but also increases their confidence of the decision.

5 CONCLUSIONS

In this work, a novel predictive deep learning model namely INPREM was proposed, which focused on three dominant features of clinical predictive models, including performance, interpretability, and trustworthiness. We experimentally proved that the INPREM outperformed all SOTA approaches in terms of accuracy by conducting experiments on the publicly available MIMIC-III dataset and a real-world EHR database. The top-10 medical codes most relevant to heart failure were presented to show the interpretability of our proposed method. Moreover, by comparing the predicting probability distribution derived from the learned variational parameters distribution with that of the deep ensembles, we demonstrated that the INPREM was able to capture uncertainty. Last but not least, we used a case study to show the interpretability of the INPREM, as well as how to use the contribution matrix and the probability distribution to assist physicians in decision-making.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under grant number 2018YFC130078; National Natural Science Foundation of China No.61672420.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).
[2] Zhengping Che, Yu Cheng, Zhaonan Sun, and Yan Liu. 2017. Exploiting convolutional neural network for risk prediction with medical feature embedding. *arXiv preprint arXiv:1701.07474* (2017).
[3] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. 2016. Risk prediction with electronic health records: A deep learning approach. In *Proc. SIAM Int'l Conf. Data Mining*. SIAM, 432–440.
[4] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. Doctor AI: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*. 301–318.
[5] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun. 2017. GRAM: Graph-based attention model for healthcare representation learning. In *Proc. ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. 787–795.
[6] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*. 3504–3512.
[7] Ingyo Chung, Saehoon Kim, Juho Lee, Kwang Joon Kim, Sung Ju Hwang, and Eunho Yang. 2018. Deep mixed effect models using Gaussian process: A personalized and reliable prediction model for healthcare. *arXiv preprint arXiv:1806.01551* (2018).
[8] Michael W. Dusenberry, Dustin Tran, Edward Choi, Jonas Kemp, Jeremy Nixon, Ghassen Jerfel, Katherine Heller, and Andrew M. Dai. 2019. Analyzing the role of model uncertainty for electronic health records. *arXiv preprint arXiv:1906.03842* (2019).
[9] Yarin Gal. 2016. Uncertainty in deep learning. *University of Cambridge* (2016).
[10] Yarin Gal and Zoubin Ghahramani. 2015. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158* (2015).
[11] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proc. Int'l Conf. Machine Learning*. 1050–1059.
[12] Paulina Grnarova, Florian Schmidt, Stephanie L. Hyland, and Carsten Eickhoff. 2016. Neural document embeddings for intensive care patient mortality prediction. *arXiv preprint arXiv:1612.00467* (2016).
[13] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, and Aram Galstyan. 2017. Multitask Learning and Benchmarking with Clinical Time Series Data. *Scientific Data* 6, 1 (2017).
[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 770–778.
[15] Jay Heo, Hae Beom Lee, Saehoon Kim, Juho Lee, Kwang Joon Kim, Eunho Yang, and Sung Ju Hwang. 2018. Uncertainty-aware attention for reliable interpretation and prediction. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). 909–918.
[16] Geoffrey Hinton and Drew Van Camp. 1993. Keeping neural networks simple by minimizing the description length of the weights. In *Proc. ACM Conf. Computational Learning Theory*. 5–13.
[17] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, H. Lehman Li-wei, Mengling Feng, et al. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data* 3 (2016), 160035.
[18] Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*. 5574–5584.
[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*. 6402–6413.
[21] Yang Li, Buyue Qian, Xianli Zhang, and Hui Liu. [n.d.]. *Knowledge guided diagnosis prediction via graph spatial-temporal network*. 19–27.
[22] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proc. ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. 1903–1911.
[23] Fenglong Ma, Jing Gao, Qiuling Suo, Quanzeng You, Jing Zhou, and Aidong Zhang. 2018. Risk prediction on electronic health records with prior medical knowledge. In *Proc. ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. 1910–1919.
[24] Fenglong Ma, Quanzeng You, Houping Xiao, Radha Chitta, Jing Zhou, and Jing Gao. 2018. KAME: Knowledge-based attention model for diagnosis prediction in healthcare. In *Proc. ACM Int'l Conf. Information and Knowledge Management*. 743–752.
[25] David J. C. MacKay. 1992. A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation* 4, 3 (1992), 448–472.
[26] Andre Martins and Ramon Astudillo. 2016. From Softmax to Sparsemax: A sparse model of attention and multi-label classification. In *Proc. Int'l Conf. Machine Learning*. 1614–1623.
[27] Radford M. Neal. 2012. *Bayesian learning for neural networks*. Springer Science & Business Media.
[28] John Paisley, David Blei, and Michael Jordan. 2012. Variational Bayesian inference with stochastic search. *arXiv preprint arXiv:1206.6430* (2012).
[29] Riyi Qiu, Yugang Jia, Mirsad Hadzikadic, Michael Dulin, Xi Niu, and Xin Wang. 2019. Modeling the uncertainty in electronic health records: A Bayesian deep learning approach. *arXiv preprint arXiv:1907.06162* (2019).
[30] Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In *Summer School on Machine Learning*. Springer, 63–71.
[31] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* (2014).
[32] Huan Song, Deepta Rajan, Jayaraman J. Thiagarajan, and Andreas Spanias. 2018. Attend and diagnose: Clinical time series analysis using attention models. In *Proc. AAAI Conf. Artificial Intelligence*. 4091–4098.
[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958.
[34] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Aidong Zhang, and Jing Gao. 2017. Personalized disease prediction using a CNN-based similarity learning method. In *IEEE Int'l Conf. Bioinformatics and Biomedicine*. IEEE, 811–816.
[35] Michalis Titsias and Neil D. Lawrence. 2010. Bayesian Gaussian process latent variable model. In *Proc. Int'l Conf. Artificial Intelligence and Statistics*. 844–851.
[36] Michalis Titsias and Miguel Lázaro-Gredilla. 2014. Doubly stochastic variational Bayes for non-conjugate inference. In *Proc. Int'l Conf. Machine Learning*. 1971–1979.
[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
[38] Xianli Zhang, Buyue Qian, Xiaoyu Li, Jishang Wei, Yingqian Zheng, Lingyun Song, and Qinghua Zheng. [n.d.]. *An Interpretable Fast Model for Predicting The Risk of Heart Failure*. 576–584.
[39] Xianli Zhang, Buyue Qian, Yang Li, Changchang Yin, Xudong Wang, and Qinghua Zheng. 2019. KnowRisk: An interpretable knowledge-guided model for disease risk prediction. In *2019 IEEE International Conference on Data Mining (ICDM)*. 1492–1497.

6 APPENDIX

In this section, we present more details regarding the reproducibility and practical usage of our model. Specifically, we present the datasets we use for experiments in Section 6.1. And then, the implementation details about the parameters we used for training are presented in Section 6.3. Finally, in Section 6.4, we present quantitative indicators in representing aleatoric uncertainty and epistemic uncertainty to assist the quantitative analysis in clinical routine.

6.1 Datasets

Diagnosis Prediction Dataset: The MIMIC-III dataset [17] consists of medical records of 7499 patients from the Intensive Care Unit (ICU). Patients who have at least two visits are chosen in the dataset. As mentioned in Section 3.1, the goal of the diagnosis prediction task is to predict the medical codes of the next visit. We use the nodes in the second hierarchy of ICD-9 codes[2] as the category labels, which is similar to [22, 24]. Table 4 lists the details about MIMIC-III.

Three Risk Prediction Datasets: We identify three cohorts from a real-world EHR database, including Heart Failure, Diabetes, and Chronic Kidney Disease. For each cohort, we first identify a set of case-patients and then select several control patients for each patient case according to the demographic information. We set the number of controls for each case to five for the Heart Failure cohort, and three for Diabetes and Chronic Kidney Disease (CKD) cohorts. We set the hold-off windows (as Fig. 2 shows) for all cohorts to 180 days. Table 5 shows the details of the three datasets.

Table 4: Statistics of the MIMIC-III Datasets [17] for Diagnosis Prediction

	MIMIC-III
# of patients	7499
# of visits	19911
Avg.# of visits per patient	2.67
# of unique ICD-9 codes	4880
Avg.# of ICD-9 codes per visit	13.06
Max.# of ICD-9 codes per visit	39
# of category codes	171
Avg.# of category codes per visit	10.16
Max.# of category codes per visit	30

6.2 Baseline Description

In the below, we give a detailed description of all baselines we used in the experiments.

CNN: A baseline NN that consists of three convolutional layers with 256 channels and the kernel size varying from 3 to 5. An output layer is applied to predict the probability of each class. ReLu, dropout, and normalization layers are also employed to obtain better performance.

RNN: We first obtain the input embeddings as Eq. (1), then feed the embeddings to a Long Short-Term Memory (LSTM) layer. The hidden states produced by LSTM are directly used to predict results by a linear classifier.

RNN+: The difference between RNN+ and RNN is that RNN+ uses a location-based attention mechanism [22] to combine the hidden states before the output layer.

RETAIN[6]: RETAIN is a SOTA predictive model. It employs a two-level attention mechanism, which could enhance both the performance and interpretability of the model.

Dipole[22]: Dipole uses a bi-directional RNN with three attention mechanisms. We select the local-based attention to obtain the final context vector because this version of Dipole has been proved to perform better in [22]. The embedding layer of Dipole is a multi-layer perceptron (MLP) layer with ReLu.

[2]http://www.icd9data.com

6.3 Implementation Details

For each task, we randomly split each dataset into training, validation, and testing sets five times in a 75:10:15 ratio. For training all approaches, we use Adam [19] with the batch size of 32 and the learning rate of 0.0005. The weight decay is set to $\lambda = 0.0001$ and the dropout rate is set to 0.5 for all approaches. We set the dimensions of embeddings and the hidden state of all baselines to 256. For our model, we set $d_{model}, d_{inner} = 256$, which are the same as all baselines. And, we set the probability of Bernoulli distribution p_k to 0.5. The times of Monto Carlo sampling in the training phase (T_{mc}) is set to 50, and in the test pahse (T_{test}) to 100. For the stacked multi-head attention module in our model, we set the number of head $m = 2$ and the number of stacking times S = 2. All approaches are implemented with PyTorch 1.0 on two Nvidia Titan XP GPUs.

Table 5: Statistics of Three Datasets for Risk Prediction

Datasets	Heart Failure	Diabetes	CKD
# of cases	1150	1095	2005
# of controls	5750	3285	6015
# of visits	103848	74146	131854
Avg.# of visits per patient	15.05	16.92	16.44
# of unique ICD-9 codes	4482	4011	4658
Avg.# of codes per visit	2.32	2.38	2.37

6.4 Quantitative Indicators of Aleatoric Uncertainty and Epistemic Uncertainty

We present quantitative indicators of aleatoric uncertainty and epistemic uncertainty. **Aleatoric uncertainty:** The aleatoric uncertainty can be measured by the predicted variance σ of the Gaussian noise as:

$$\sigma^* = \frac{1}{T_{test}} \sum_{t=1}^{T_{test}} \sigma_t. \tag{22}$$

The estimation of aleatoric uncertainty makes INPREM robust to data noise. In future work, we will demonstrate that our INPREM can work well even with the EHR records being largely corrupted.

Epistemic uncertainty: Since we perform T_{test} Monto Carlo samplings for each data point in the test phase, which leads to T_{test} predictions, we propose two ways to quantize the epistemic

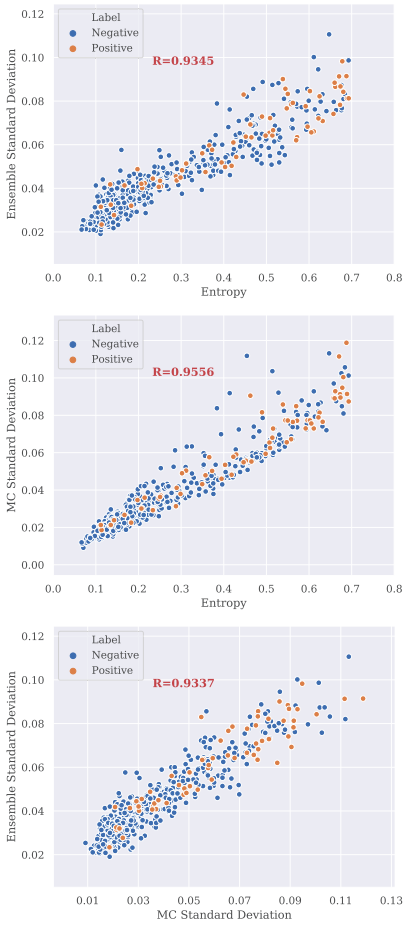

Figure 7: Correlation of the entropy (Epi_ϵ), MC standard deviation (Epi_ς), and the ensemble standard deviation.

uncertainty. The first one is to use the entropy of the probability p_i predicted by Eq. (21), which can be calculated as follow:

$$Epi_\epsilon = -\sum_{i=1}^{2} p_i \log(p_i). \tag{23}$$

The second quantization method is to compute the standard deviation of the total T_{test} predictions as Epi_ς. Note that we can also obtain T_{test} contribution matrices for each prediction, which further enable us to quantize the epistemic uncertainty Epi_ς for the contribution matrices.

To verify that both the Epi_ϵ and Epi_ς could take effect equally in measuring the epistemic uncertainty, we visualize the Pearson correlation coefficient of Epi_ϵ, Epi_ς, and the standard deviation of probability distribution derived from ensembles in Fig. 7. We find that they have a quite high linear correlation with each other, which verifies that both the proposed Epi_ϵ and Epi_ς can provide quantitative indicators for assisting the physicians in decision making.