RetainVis: Visual Analytics with Interpretable and Interactive Recurrent Neural Networks on Electronic Medical Records

Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi,
Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo

Fig. 1. A screenshot of RetainVis that consists of five areas: (A) *Overview* shows an overview of all patients (left) and an attribute summary view (right) of patients. (B) *Patient Summary* shows the summary of patient cohorts built from (A). (C) *Patient List* shows individual patients in a row of rectangles. In Patient List, users can select a patient of interest to view details in (E) *Patient Detail*. Users can open (D) *Patient Editor* to conduct a what-if analysis, and (E) *Patient Detail* shows the updated results.

Abstract— In the past decade, we have seen many successful applications of recurrent neural networks (RNNs) on electronic medical records (EMRs), which contain histories of patients' diagnoses, medications, and other various events, in order to predict the current and future states of patients. Despite the strong performance of RNNs, it is often very challenging for users to understand why the model makes a particular prediction. Such *black box* nature of RNNs can impede its wide adoption in clinical practice. Furthermore, we have no established method to interactively leverage users' domain expertise and prior knowledge as inputs for steering the model. Therefore, our design study aims to provide a visual analytics solution to increase interpretability and interactivity of RNNs via a joint effort of medical experts, artificial intelligence scientists, and visual analytics researchers. Following the iterative design process between the experts, we design, implement, and evaluate a visual analytics tool called RetainVis, which couples a recently proposed, interpretable RNN-based model called RETAIN and visualizations for users' exploration of EMR data in the context of prediction tasks. Our study shows the effective use of RetainVis for gaining insights into how RNN models EMR data, using real medical records of patients with heart failure, cataract, or dermatological symptoms. Our study also demonstrates how we made substantial changes to the state-of-the-art RNN model called RETAIN in order to make use of temporal information and increase interactivity. This study will provide a useful guideline for researchers who aim to design more interpretable and interactive visual analytics tool for RNNs.

Index Terms—Interactive Artificial Intelligence, XAI (Explainable Artificial Intelligence), Interpretable Deep Learning, Health, Visual Analytics, Recurrent Neural Network, Electronic Medical Records (EMR)

1 Introduction

In the past decade, we have seen many successful applications of deep learning techniques such as recurrent neural networks (RNNs) on electronic medical records (EMRs), which contain histories of patients' diagnoses, medications, and other various events, in order to predict the current and future states of patients. This recent movement is related to two key factors. First, artificial intelligence scientists have been continuously making great advancements in deep learning algorithms and technologies. Second, although some challenges and concerns (*e.g.* security and privacy issues) still remain, public and private sectors have started to recognize the needs to make electronic medical records more accessible in order to fully leverage the power of deep learning techniques in clinical practice. These two factors have created a surge of deep learning applications for EMRs, many of which are adopting RNN-based approaches.

Despite the popularity and the ever-increasing performance of RNNs,

- *Bum Chul Kwon is with IBM T.J. Watson Research Center. E-mail: bumchul.kwon@us.ibm.com.*
- *Min-Je Choi, Joanne Taery Kim, Jaegul Choo are with Korea University. E-mail: devnote5676, tengyee, jchoo@korea.ac.kr.*
- *Edward Choi and Jimeng Sun are with Georgia Institute of Technology. E-mail: mp2893, jsun@gatech.edu.*
- *Young Bin Kim is with Chung-Ang University. E-mail: ybkim85@cau.ac.kr.*
- *Soonwook Kwon is with Catholic University of Daegu. E-mail: anatomy3@cu.ac.kr.*

there are many challenges to overcome before the full adoption by clinical practice. A key challenge is for domain experts to understand why the model makes a particular prediction. The experts also need to be involved in improving the performance of RNNs by providing relevant guidance in order to reduce the risk of costly Type II errors. Yet, we have no established method to interactively leverage users' domain expertise and prior knowledge as inputs for steering the model.

Thus, our study aims to tackle the problem of intepretability and interactivity by designing a visual analytics solution with an RNN-based model for predictive analysis tasks on EMR data. Our design study involved many iterative design, assessment, and discussion activities between medical experts, artificial intelligence scientists, and visual analytics researchers. After we characterized target users' tasks, we designed, implemented, and evaluated a visual analytics tool called RetainVis with a more interactive, interpretable RNN-based model that we name RetainEX in order to fulfill the users' needs.

Our study shows the effective use of RetainVis for gaining insights into how RNN models EMR data, using real medical records of patients with heart failure and cataract. Our study also demonstrates how we made substantial changes to the state-of-the-art RNN model called RETAIN, thereby inventing a new model called RetainEX, in order to make use of temporal information and increase interactivity and interpretability at the same time. Various visualizations coupled with the new model allow users to test their hypotheses and to learn interesting stories from patients' medical history. This study will provide a useful guideline for researchers who aim to design interpretable and interactive visual analytics tools with RNNs.

Here we describe the three main contributions of our study:

1. We introduce an interpretable, interactive deep learning model, called RetainEX, for prediction tasks using EMR data by improving the state-of-the-art model (RETAIN) with additional features for improved interactivity and temporal information.
2. We design and develop a visual analytics tool, called RetainVis, which tightly integrates the improved deep learning model with the design of visualizations and interactions.
3. We conduct both quantitative experiments and a case study with real medical records of patients and discuss the lessons we learned.

Section 2 discusses related work from three different perspectives. Section 3 introduces our target user, data and defined tasks. Section 4 briefly explains the structures of our backbone model and also describes a number of new features we added to fulfill our designated tasks. Section 5 is a step-by-step view of the novel features of our visual analytics system. Section 6 shows the quantitative and qualitative experiments we conducted using an actual EMR dataset. Section 7 shows a user case study we conducted with our framework. Section 8 covers the lessons and implications we learned from three different perspectives. Lastly, Section 9 includes our concluding remarks and suggestions for future work.

2 Related Work

In this section, we review existing works using three main axes on which our work rests: deep learning applications for EMR data, visualization techniques of black-box deep models, and machine learning platforms that allow a wide range of user interactivity.

2.1 Deep learning for electronic medical records

Although the most prevalent use of deep learning techniques in medical domains is to diagnose conditions such as breast cancer [1, 27, 69] and brain tumor [22,28,35] by training models on medical images, there has also been an increase in the application of deep learning to longitudinal EMR data. RNN-based models have been extensively used for tasks such as patient diagnosis [60,71], risk prediction [7,11,15,32,51,56,72], representation learning [12, 13] and patient phenotyping [8, 34, 52], outperforming rule-based and conventional machine learning baselines such as logistic regression and ensemble classifiers.

A less-considered but important issue to consider when designing prediction models using medical data is the interpretability of the model. Medical tasks such as patient diagnosis are performed by clinicians who have sufficient domain knowledge and can explain the reasons of their diagnoses by relating it to past visits of the patient. It is important for machine learning models to incorporate a similar level of interpretability, but many deep learning-based studies in this field fail to address this aspect.

To the best of our knowledge, RETAIN [14] is one of the few deep learning models applied to medical data that both harnesses the performance of RNNs and preserves interpretability as to how each data point is related to the output. In RETAIN, it is possible to decompose the output score to the level of individual medical codes that occurred in a patient's previous visits. While there are other models for EMR data that suggest interpretability such as Dipole [54], the level of interpretability is limited to each visit and thus does not provide a complete decomposition of the prediction as RETAIN. For this reason, we use the RETAIN framework for ensuring interpretability of our tool.

2.2 Visualization of deep learning models

A major concern in the application of deep learning models is the 'black-box' nature of neural networks. As a result, many approaches have been suggested for visualizing the dynamics in various types of neural networks. Especially for vision applications where convolutional neural networks (CNNs) have enjoyed a great success, various methods for visualization include heatmaps [3, 67, 82, 83], blurring [77], and dimensionality reduction [53, 61] of the filters and activation maps obtained during computation and backpropagation, and visualizing the model structure itself [79]. This led to a large number of studies dedicated to developing visualization frameworks that help users to better understand their networks [18, 31, 33, 59, 68].

Compared to CNNs, RNNs have received less attention in visualization, mainly because of its intertwined structure and its popularity in textual data. Though it is possible to visualize the activations of hidden state cells [37, 55, 70], they do not propose the level of interpretability as in CNNs. In this aspect, our work makes a substantial contribution in that it aims to provide direct interpretations of the outputs computed using RNNs, supported with a visual analytics tool.

2.3 Interactive machine learning platforms

A topic of emerging importance in the field of visual analytics is integrating machine learning models with various user-led interactions [62]. Instead of passively observing the computed results of machine learning models projected on the screen, users can make certain interactions to the inputs or outputs which, in turn, further influence the model. This setting enables users to conduct what-if case analyses by adding, editing, or removing data items and then recomputing the outputs. Additionally, a user can instill the model with his/her prior knowledge to correct errors and further improve model performance.

There have been a number of studies to develop tools where users can interact with the results of machine learning tasks such as classification [25, 29, 50], topic modeling [16, 26, 48], dimensionality reduction [44] and clustering [17, 43, 47]. However, there are only a small number of works that apply user interaction to tasks that require deep learning models, such as object segmentation [78]. To the best of our knowledge, we are the first to apply such user interaction to RNN-based models for medical tasks. RetainVis is also the first to enable direct interaction with the visualized results computed from a deep learning model.

3 Users, Data, and Tasks

In this section, we describe the target users, the input data, and the analytic tasks (questions) the users desire to solve. Based on the description, we review requirements for our model and visualization framework.

3.1 Physicians, Health Professionals, and Researchers

The target users of our visual analytics system include physicians, health professionals, and medical researchers who have access to electronic medical records (EMRs). They need to answer various questions related to diagnosis, prescription, and other medical events. One of their tasks is to accurately estimate the current and future states of patients. In addition, they want to investigate the common patterns of patients with the same target outcome (e.g., diabetes). In particular,

domain experts often want to conduct what-if analysis on individual patients by testing hypothetical scenarios.

3.2 Data

The dataset used in our visual analytics system, collected between years 2014 and 2015, was provided by the Health Insurance Review and Assessment Service (HIRA) [39], the national health insurance service in the Republic of Korea. The HIRA dataset contains the medical information of approximately 51 million Koreans. In particular, the National Patients Sample (HIRA-NPS) dataset consists of information on approximately 1.4 million patients (3% of all Korean patients) and their related prescriptions. The HIRA-NPS dataset was constructed using age- and gender-stratified random sampling. The representativeness and validity of this sample dataset have been confirmed by thorough evaluation against the entire patient population of Korea [40]. The HIRA-NPS contains each patient's encrypted, unique, anonymized identification (ID) number, medical institution ID number, demographic information, gender, age, primary and secondary diagnoses, inpatient or outpatient status, surgical or medical treatment received, prescriptions, and medical expenses. Each diagnosis is encoded based on the Korean Standard Classification of Disease, Ninth Revision (KCD-9).

3.3 Tasks

In this section, we report our target users' tasks. We iteratively identified our target tasks based on weekly discussions among co-authors of this paper, who are experts in visual analytics, deep learning, and medical domains. We initially generated research questions of our target users' potential interest (led by medical experts), derived visual analytics tasks (led by deep learning and visual analytics experts), and then further evaluated them iteratively by closely following a design study methodology [64]. In particular, all leading authors, who were experts in two of the three domains of interest (i.e., deep learning, visual analytics, medical domains), each played the liaison role to fill the gap between domain and technical areas, as Simon et al. [66] suggests.

The following list shows the main user tasks:

T1: **Gain an overview** of patients with respect to their demographic information and medical records. Users want to understand the entire dataset. They also aim to learn about the summary of demographic attributes, such as the mean and distribution of patients' age and gender, as well as the model's prediction scores.

T2: Discover and select **interesting patient cohorts**. Users often want to test their hypotheses against prior knowledge by selecting specific patient cohorts. In particular, they want to define the cohorts based on various patient attributes.

T3: **View a summary of selected patients** based on visits, medical codes, and prediction scores. Users want to grasp the summary of selected patients. The summary should include the temporal overview of visits, medical codes, and prediction scores.

T4: **Check details** of visits, medical codes, and prediction scores of patients. Users want to inspect the details of individual patients.

T5: **Understand why** each prediction is made based on patients' visits and medical records. It is challenging but required for users to understand why a deep learning model predicts particular outcomes. Especially, users want to understand the reason in a interpretable manner, such as how particular visits contribute to the prediction.

T6: **Conduct what-if case analyses** on individual patients (e.g., add/edit/remove medical code, change visit intervals). Users want to test their hypothetical scenarios on individual patients. For instance, users can check whether the prediction score decreases as they insert a hypothetical visit with a series of treatments.

T7: **Evaluate and steer the predictive model** by viewing the summary of prediction scores and providing feedback on visits and medical codes. Users want to check the model if it acts in line with users' prior knowledge. If the model behaves in a undesirable manner, users can provide relevant feedback to the model so that they can improve the model's prediction and interpretation.

By reviewing the tasks, we agreed that a visual analytics system with a recurrent neural network (RNN) model would be a suitable combination to help users accomplish their goals. In particular, we needed a variant of RNN that is capable of revealing interpretable outcomes. Thus, we chose the state-of-the-art, interpretable RNN-based model, called RETAIN [14]. However, we quickly realized that RETAIN needed significant improvement in order to fulfill our target users' needs, especially by considering temporal aspects of EMR (i.e., days between visits) and by allowing users to steer the model based on user inputs (**T3**−**T7**). In Section 4, we introduce the improved model called RetainEX. In Section 5, we describe the design of our visual analytics tool, called RetainVis, and how it fulfill users' needs together with RetainEX.

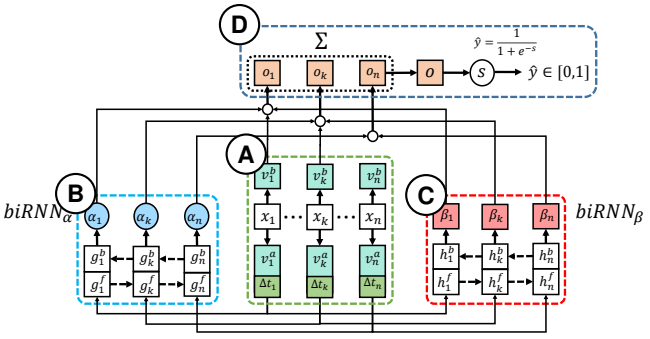

Fig. 2. Overview of RetainEX. See Appendix A for a larger diagram. (A) Using separate embedding matrices, the binary vectors $\mathbf{x}_1,\ldots,\mathbf{x}_T$ are transformed into embedding vectors $\mathbf{v}^a{}_1,\ldots,\mathbf{v}^a{}_T$ and $\mathbf{v}^b{}_1,\ldots,\mathbf{v}^b{}_T$, with time interval information appended to the former. (B) $\mathbf{v}^a{}_1,\ldots,\mathbf{v}^a{}_T$ are fed into a bidirectional RNN to produce scalar weights α. (C) $\mathbf{v}^a{}_1,\ldots,\mathbf{v}^a{}_T$ are fed into another biRNN, this time to generate vector weights β. (D) α, β and $\mathbf{v}^b$ are multiplied over all timesteps, then are summed to form a single vector o, which goes through linear and nonlinear transformation to produce a probability score $\hat{y}$.

4 Model Description

In this section, we describe the structure of our prediction model, which we name RetainEX (RETAIN with extra time dimensions and embedding matrices). We explain the additional features that we incorporated into the original model for greater interactivity, and show how they are capable of fulfilling the user tasks we defined in the previous section.

4.1 Structure of EMR data

A patient's EMR data contain information of a patient's visits over time. It is usually recorded as a sequence of medical codes, where each code corresponds to either a doctor's diagnosis of a patient, a treatment or surgery, or a prescribed medicine. In this sense, we can consider the data of a patient as a sequence of vectors $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_T$, with T as the total number of visits. For each binary vector $\mathbf{x}_t \in \{0,1\}^{|C|}$ with C as the number of unique codes, $\mathbf{x}_{t,c}$ is set to 1 if code c is observed in visit t; otherwise set to 0. Note that each visit may contain more than one code, which results in each $\mathbf{x}_t$ containing multiple values of 1. In this paper, we focus on using such sequential data on a prediction task, 'learning to diagnose (L2D)' [51], where a model observes the visit a patient's data and returns a prediction score indicating the probability of the patient being diagnosed with a target disease in near future.

4.2 RetainEX: An interactive time attention model

In this part we go through the computational process of our model. Note that due to the recent popularity of deep learning models in the data visualization community, we assume that the readers are familiar with how RNNs generate hidden state vectors and leave a detailed explanation at Appendix A.1.

The overview of our model is displayed in Fig. 2. As seen in Fig. 2a, our model takes as input the patient visit sequence as C-dimensional vectors $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_T$ along with the time intervals between each visit, $\Delta t_1,\Delta t_2,\ldots,\Delta t_T$. Our model uses two embedding matrices $\mathbf{W}^a_{emb} \in \mathbb{R}^{m\times C}$ and $\mathbf{W}^b_{emb} \in \mathbb{R}^{m\times C}$ to convert the binary vectors into continuous vectors. We obtain a representation vector for each visit as

$$\mathbf{v}^a_t = \mathbf{W}^a_{emb}\mathbf{x}_t,. \tag{1}$$

The vectors $\mathbf{v}_1^b, \ldots, \mathbf{v}_T^b$ are obtained likewise. As each visit is associated with a time interval, we compute the three different time values (explained in Section 4.4 and append the values to each vector $\mathbf{v}_t^a$.

Figs. 2b and c represent the bidirectional RNNs that take in the time-attached visit representations and return attention values of different scales. For each $\mathbf{v}_t^a$, biRNN$_\alpha$ computes the forward and backward hidden states, $\mathbf{g}_t^f$ and $\mathbf{g}_t^b$, which are concatenated as a single $2m$-dimensional vector. We use a parameter $\mathbf{w}_\alpha \in \mathbb{R}^{2m}$ to compute a scalar value for each timestep as

$$e_t = \mathbf{w}_\alpha \left[\mathbf{g}_t^f; \mathbf{g}_t^b\right], \tag{2}$$

Then, we apply the softmax function on all scalar values $e_1, \ldots, e_T$ to obtain $\alpha_1, \alpha_2, \ldots, \alpha_T$, a distribution of attention values that sum to one. Similarly, the concatenated hidden state vectors generated using biRNN$_\beta$ are multiplied by $\mathbf{W}_\beta \in \mathbb{R}^{\mathrm{m} \times 2m}$ and return an m-dimensional vector β_t for the t-th timestep as

$$\beta_t = \mathbf{W}_\alpha \left[\mathbf{h}_t^f; \mathbf{h}_t^b\right], \tag{3}$$

.

Once we obtain both alpha and beta values, we multiply these values with our other set of embedding vectors, $\mathbf{v}_1^b, \ldots, \mathbf{v}_T^b$, and add up the values to obtain the context vector $\mathbf{o}$ as in

$$\mathbf{o} = \sum_{t=1}^{T} \alpha_t \left(\mathrm{W}_{emb}^b[:, c] \odot \beta_t\right), \tag{4}$$

with $\odot$ indicating elementwise multiplication of two vectors. Finally, we obtain a prediction score $\hat{y}$ ranged between 0 and 1 as

$$s = \mathbf{w}_{out}^T \mathbf{o}, \tag{5}$$

$$\hat{y} = \frac{1}{1 + e^{-s}}, \tag{6}$$

where $\mathbf{w}_{out} \in \mathbb{R}^m$. We train our model by optimizing all learnable parameters to minimize the cross-entropy loss defined as

$$\mathscr{L} = -\frac{1}{N} \sum_{i=1}^{N} y_i log\left(\hat{y_i}\right) + (1 - y_i) log(1 - \hat{y_i}) \tag{7}$$

with y_i as the target value of 0 or 1 for the i-th patient and N the total number of patients.

4.3 Bidirectionality

An increasing trend in RNNs is the shift towards bidirectional models. Compared to traditional RNNs which are limited to processing the input sequence in one direction (*i.e.* from the first input to the last input), bidirectional RNNs (biRNNs) introduce another set of hidden state vectors that are computed by starting from the last input and processing backwards. This is obtained as

$$\begin{aligned} \mathbf{h}^{\mathrm{f}}{}_1, \mathbf{h}^{\mathrm{f}}{}_2, \ldots, \mathbf{h}^{\mathrm{f}}{}_T &= \mathrm{RNN}_{\mathrm{f}}\left(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T\right), \\ \mathbf{h}^{\mathrm{b}}{}_T, \mathbf{h}^{\mathrm{b}}{}_{T-1}, \ldots, \mathbf{h}^{\mathrm{b}}{}_1 &= \mathrm{RNN}_{\mathrm{b}}\left(\mathbf{v}_T, \mathbf{v}_{T-1}, \ldots, \mathbf{v}_1\right), \end{aligned} \tag{8}$$

where the final hidden state vector $\mathbf{h}_t$ for each timestep is obtained by concatenating the two hidden states $\mathbf{h}^{\mathrm{f}}{}_t$ and $\mathbf{h}^{\mathrm{b}}{}_t$, resulting from the forward and backward RNNs.

BiRNNs are known to outperform their unidirectional counterpart in most tasks that utilize sequential data processing. In the case of EMR-based diagnosis, this closely matches a clinician's behavior where one may observe patient history in a chronological order to see how patient status progresses through time, while also tracing back to look for possible cues that may strengthen or weaken his confidence of the patient's current state. While the original RETAIN model uses unidirectional RNNs in a reverse direction, we formulate a more intuitive and accurate prediction model by processing the input data with biRNNs.

4.4 Application of time interval data

Each visit in EMR data is associated with a timestamp t_i, which enables us to calculate time intervals between visits. Though basic RNNs only consider the order of the input sequences and not the time intervals, the temporal aspect is actually a key to the disease diagnosis. For instance, a burst of similar events in a short time period may forebode the manifestation of a serious illness, while a long hibernation between two events may indicate that the event may not be influential for diagnosis.

To harness temporal information, we incorporate visit dates as an additional feature to the input vectors of our RNN model. Given a sequence of T timestamps $t_1, t_2, \ldots, t_T$, we can obtain T interval values $\Delta t_1, \Delta t_2, \ldots, \Delta t_T$ with $\Delta t_i = t_i - t_{i-1}$. We assume that the first visit is unaffected by time constraints by fixing Δt_1 to 1. For each Δt_i, we follow the time-decaying scheme of [2] to calculate different representations of time which are obtained as

$$\begin{aligned} \mathrm{time}_1\left(\Delta t_i\right) &= \Delta t_i, \\ \mathrm{time}_2\left(\Delta t_i\right) &= 1/\Delta t_i, \\ \mathrm{time}_3\left(\Delta t_i\right) &= 1/\log\left(e + \Delta t_i\right). \end{aligned} \tag{9}$$

These three values are concatenated to the input vectors of each step.

4.5 Understanding the interpretability of RetainEX

The 'black-box' nature of RNN models makes it impossible to track the direct relation between the inputs and predicted values. However, our model is capable of calculating the degree of how much each input code contributes to the output. Using this feature, we can discover which features heavily influenced the model's prediction (e.g., heart failure prediction, readmission prediction).

T4&T5: Understanding how predictions are made RetainEX achieves its transparency by multiplying the RNN-generated attention weights α_ts and β_ts to the visit vectors $\mathbf{v}_t$ to obtain the context vector $\mathbf{o}$, which is used, instead of the RNN hidden state vectors, to make predictions. As seen in Eqs. 5, 8, and 9, each input vector $\mathbf{x}_t$ has a linear relationship with the final score s. Considering that each $\mathbf{x}_t$ is a binary vector where the c-th value is set to 1 only when the c-th code is present in the t-th visit, we can formulate an equation that directly measures the contribution score of the code c at timestep t to s by reformulating the above equations as

$$s_{t,c} = \alpha_t \mathbf{w}_{out} \left(\mathbf{W}_{emb}^b[:, c] \odot \beta_t\right). \tag{10}$$

where $\mathbf{W}_{emb}^b[:,c]$ is the c-th column of $\mathbf{W}_{emb}^b$.

In our model we provide two levels of interpretability: visit- and code-level. The code-level interpretation score is the contribution score of code c at timestep t as described in the above equation. We can also formulate a visit-level interpretation score s_t that represents the importance of each visit as

$$s_t = \sum_{c \in \mathbf{x}_t} s_{t,c}.$$

T3: A summary of selected patients It is possible to create a vectorized representation of each patient using these contribution scores. We assign a 1400-dimensional zero vector $\mathbf{S}$ to each patient, compute all individual contributions scores for all codes in every visit that a patient had, and add the contribution score of each one (e.g., $s_{t,c}$) to the corresponding row of $\mathbf{S}$, i.e., $\mathbf{S}[c]$. The dimension size of 1,400 is due to our preprocessed dataset containing 500 treatment codes, 268 diagnosis codes, and 632 prescription codes (details are covered in Appendix B). The resulting $\mathbf{S}$ can be seen as a patient embedding whose sum of elements and direction each indicate the prediction score and distribution of input features. We later use these vectors to create an overview of the patient distribution for further exploratory analyses.

4.6 Interactions featured in RetainEX

While interpretability is crucial for a visual analytics tool incorporating deep learning, as it shows the underlying process beneath a model's prediction, interactivity, where users can modify inputs and recompute new results, enables another level of capability. We introduce three types of interaction: adding or removing input codes, modifying visit periods, and retraining the model while enforcing it to increase or decrease contribution scores of individual codes. In the following paragraphs, we explain the technical details that enables such interactions.

T6: Conducting what-if case analysis It is easy to add or remove a number of codes per visit, as this only requires a simple modification to $\mathbf{x}_t$ by changing an element to 1 or 0 with all other input vectors fixed, and then feeding all inputs into the model again for recomputation. The modification of time intervals can be applied in a similar manner. Our model provides a slider for users to freely modify the time intervals between different visits. Once a time interval Δt_i changes, the corresponding three time values described in Eq. 8 are also updated, and put into the model for recomputation. This is a simple and effective scheme for incorporating temporal information into the model, which also guarantees improved performance.

T7: Evaluating and steering the predictive model We implemented a more challenging type of user interaction where a user, after observing the contribution scores of each visit, can decide to increase or decrease the scores of individual visits according to his or her medical knowledge. While the earlier two types of interaction are straightforward in that we modify the inputs and recompute to obtain new results, the third type of interaction is more complex since the model now has to update itself according to the user's actions so that it can place more weight to the specified inputs without harming the overall distribution of attentions assigned to different visits and codes.

In the original RETAIN, the visit embeddings $\mathbf{v}_1,\ldots,\mathbf{v}_T$, which are obtained from the binary vectors $\mathbf{x}_1,,\ldots,\mathbf{x}_T$ and the embedding matrix $\mathbf{W}_{emb}$, are used to both (1) compute alpha and beta weights, and (2) obtain the final outputs by multiplying itself with the obtained alpha and beta values and then summing up across all timesteps to form a single context vector $\mathbf{o}$. As we only want to change the contributions of our designated code(s) at a particular visit without changing the alpha and beta attentions at other visits, we can formulate an optimization problem where we minimize $\mathscr{L}_{retrain}$ as

$$\mathscr{L}_{retrain} = e^{-s_{pos}+s_{neg}},$$

with s_{pos} and s_{neg} being the sums of user-selected contribution scores $s_{t,c}$ to either increase or decrease. The retraining process thus becomes equivalent to performing a number of gradient descent operations to the parameters, which we restrict to $\mathbf{W}_{emb}$.

After retraining the model for a small number (e.g., 10) of iterations with a small learning rate, ideally we would like to observe a situation where only the contributions of the inputs of our interest are modified while all other contributions remain the same. However, one problem regarding this setting in the original model is that changing $\mathbf{W}_{emb}$ changes all alpha and beta values as well, which results in affecting every contribution score. In order to preserve the overall attention distribution while only changing the weights of medical codes, the embeddings used for calculating alpha and beta values need to be separated from the embeddings that are involved in retraining.

Here we apply relaxation to our model by introducing two embedding matrices $\mathbf{W}^a{}_{emb}$ and $\mathbf{W}^b{}_{emb}$, subsequently producing two sets of visit embeddings $\mathbf{v}^a{}_1,\ldots,\mathbf{v}^a{}_T$ and $\mathbf{v}^b{}_1,\ldots,\mathbf{v}^b_T$. The first set is used to compute the alpha and beta attention weights, while these weights are multiplied to the second set for the final outputs. Using this setting, we can control the influence of individual codes without altering the overall attention by retraining our model with respect to $\mathbf{W}^b{}_{emb}$.

Not only were we able to allow for a greater variety of user interaction by adding such modifications to our model, but we also experienced increased model performance. Briefly put, the enhanced version of RETAIN we created captures interpretability, user interaction, and high performance of deep learning all at once.

5 RetainVis: Visual Analytics with RetainEX

In this section, we describe views that tightly integrate the improved RETAIN model, RetainEX, introduced in the previous section. We also explain how users are able to trigger the computed contribution scores as interactive features, such as what-if analysis and retraining and view results on visualizations. We also show how each view and interaction feature fulfills the user tasks. Many design decisions are made through several trials-and-errors. We provide our design rationale with some failure stories as well.

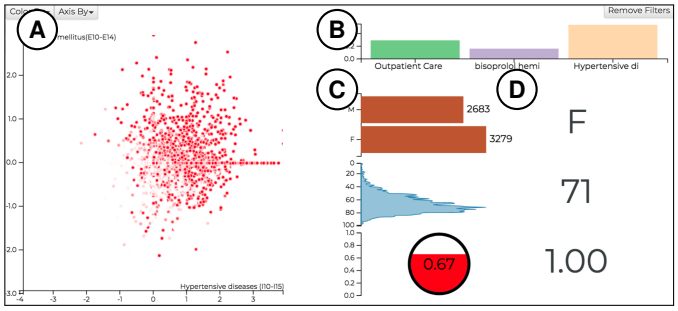

Fig. 3. *Overview* shows all patients in (A) a scatter plot; (B) A bar chart shows the top three contributors and their mean scores; (C) Three charts (i.e., bar chart, area chart, and circle chart) show the mean and distribution of gender, age, and prediction scores, respectively; (D) The text area shows a selected patient's gender, age, and prediction score.

5.1 Overview

Overview aims to provide a summary of patients with respect to their medical codes, contribution scores, and prediction scores (**T1**). To derive this overview, we use $\mathbf{S}$ (see Section 4.5 for detail), which is a list of all patients vectorized with contribution scores of medical codes. We ran *t-SNE* on $\mathbf{S}$ to derive two dimensional vector list that represents the position of patients on 2-D space. Figure 1 shows that Overview presents patients and their differences with the distance between points.

In Overview, users can choose to map between patient attributes (e.g., age, gender, contribution scores) and two graphical variables: color and position (axes). For instance, users can map prediction scores to a linear color scale of white to red (0 to 1) as shown in Figure 3. Users can also show male and female patients in different colors. Then, users can also switch axes by choosing two out of any attributes. Figure 3 shows that the user chose two comorbidities of heart failure patients, namely hypertensive diseases (x) and diabetes mellitus (y). The chart shows the model's overall high scores around the region except for lower left corners–which indicates patients with low contribution scores of both hypertension and diabetes. From the view, we can hypothesize that predicting patients without strong contributions of any of these comorbidities will be difficult for the model.

The right side of Overview shows four charts: code bar chart, gender bar chart, age area chart, and prediction circle chart, from top to bottom. The four charts mainly summarize patients by their attributes. To avoid overplotting, we only show the top three highest contributors in code bar chart. The contribution scores were computed by patient-wise mean of the corresponding codes in score vectors of patients ($\mathbf{S}$). Users can see the distribution of age and gender in gender bar chart and age area chart, respectively. Prediction circle chart shows the mean prediction score as the gauge filled in the circle. This particular icon is consistently used to show individual patient's prediction scores in Patient List as well. The bottom right corner of Overview shows gender, age, and prediction score of a selected/highlighted patient from other views.

The five charts in Overview not only serves as summary of patients but also acts as custom cohort builders (**T2**). Using coordinated interaction between the five charts, users can define customized patient groups by setting filters on each view. In scatter plot, users can draw a polygonal area with a lasso drawing tool. Once users complete drawing, the points surrounded by the drawn region are highlighted. In addition, other views quickly show a summary of the highlighted points: 1) dotted bars for the mean values of selected patients in code bar chart; 2) distributions of selected patients in yellow bars and yellow areas in gender bar chart and age area chart, respectively; and 3) mean prediction score as a dotted horizontal line in prediction circle chart. In a similar fashion, users can set filters in other views: by clicking bars or brushing axes. Thus, Overview highlights patients that satisfy all conditions set by users. For instance, users can select a small cohort of six patients by drawing an area representing positive contributions from both ischaemic heart diseases (x) and pulmonary heart diseases (y) as well as choosing the age group between 60 and 80 in our data set.

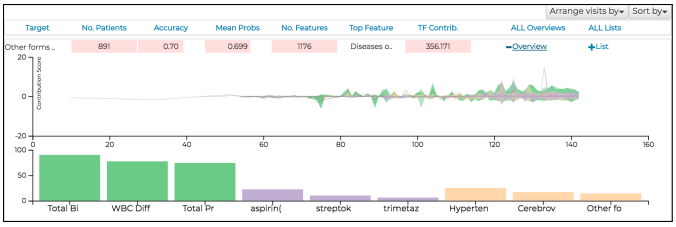

Fig. 4. *Patient Summary* shows a summary of selected patients. Table summarizes description of selected patients. In the middle, an area chart shows aggregated contribution scores of nine medical codes over time. It shows mean and standard deviation as an area. Users can also see the medical codes and their mean contribution scores in bar chart.

5.2 Patient Summary

Patient Summary aims to give a summary of selected patients. The difference of Patient Summary from Overview is temporality, especially in contribution progress chart (**T3**).

There are three charts vertically shown from top to bottom in Patient Summary. The first chart is a table that summarizes selected patients: 1) the number of patients, 2) accuracy (the number of correct prediction / number of patients), 3) mean prediction scores, 4) the number of medical codes, 5) the name of top contributing medical code, 6) sum of contribution scores. Then, it provides an interaction handle that toggles contribution progress chart and code bar chart.

In contribution progress chart, users can see a temporal overview of nine selected medical contribution scores over sequences or time. The temporal area chart is constructed in the following way: 1) we align all sequences of medical codes to the final visit; 2) starting from the final visit backward, we compute the mean and standard deviation of contribution scores of the corresponding codes across patients; 3) we visualize the computed means and standard deviation over time as area paths along with the horizontal axis. The thickness represents variance, and the vertical spikes show the mean around each visit (with respect to the most recent visit). Since patients with longer sequences, such as 120 visits, are rare, it tends to show almost a single line toward the left side. Figure 4 shows that the green codes (diagnosis) show higher variance over time than other types. We can also observe some negative contributions (i.e., downward spikes) of the green codes in the middle.

Code bar chart shows the top nine contributors of the patients: three per each of three different code types (diagnosis, medication, and disease). Users can also filter contribution progress of a selected code by hovering over the corresponding bar. By selecting one of the nine codes, users can also sort Patient List by the contribution scores of it. The three views provide an overview of selected patients. After observing peculiar, downward spikes of the contribution scores of aspirin around 10-to-15 visits before the end, users can sort patients by the contribution scores of aspirin in Patient List (**T3**).

5.3 Patient List

Patient List provides a list of selected patients, where users can explore and compare multiple patients. In Patient List, each patient's visit record is represented as rectangular boxes arranged horizontally inspired by prior work in visualizing sequences [45,46]. Each box, which represents a single visit, shows a color from the blue-to-white-to-red (negative-to-0-to-positive) scale, representing the sum of contribution scores of all codes in the visit. At the rightmost end of the visit boxes, a prediction circle icon, which was also used in Overview, shows the strength of the prediction score. In this view, users can quickly glance the temporal pattern of contribution scores of individual patients and select one patient for a deep-dive analysis. Figure 5 shows a list of patients with high prediction scores. Patients tend to have visits with high contribution scores spread towards more recent visits, but exceptions can be seen (**T5**). Interestingly, in case of Patient−6 (highlighted), the very first visit was the single positively contributed visit. In Patient List, users can invoke Patient Detail and Patient Editor of a selected patient.

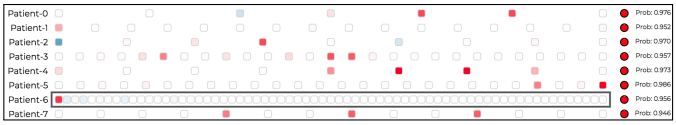

Fig. 5. Users can explore patients' visits in Patient List. One row represents a patient, where rectangular boxes (visits) are colored based on sum of contribution scores of the day (red: high, blue: low) and horizontally distributed based on sequences or dates. At the rightmost end, a circle represents the model prediction score.

5.4 Patient Detail

Patient Detail shows a focused view of a single patient (**T4**). It consists of three different views as shown in Figure 6. The first view is a line chart of prediction scores. The prediction scores over time (sequences) are calculated in the following way: 1) starting from the very first visit, we compute prediction scores by considering only the preceding visits until the corresponding visit; 2) then, we can compute N prediction scores per patient, where N is the total number of visits per patient; 3) we also compute the contribution scores of individual medical codes per prediction score, which will be used in temporal code chart.

Temporal code chart initially shows contribution scores of all medical codes for each patient. The view is similarly arranged horizontally per visit as in Patient List. Temporal code chart unpacks individual visits into separate medical codes. The medical codes are represented as colored symbols: green plus (diagnosis), purple diamond (prescription), yellow rectangle (sickness), according to their type. The colored symbols are placed vertically with respect to their contribution scores. This way, users can easily observe the contributions of the medical codes of different types. Code bar chart shows the top nine contributing medical codes (same as in Patient Summary).

In this view, users can understand the progression of prediction scores and why the scores are made (**T5**). When users hover along with the x-axis of the line chart of prediction scores, users can also see the contribution scores of medical codes of preceding visits until the point of time. Users can also filter symbols in temporal code chart by their medical types. Users can choose to show only the temporal progress of contribution scores of sickness codes, which reveal the highly negative contribution score of a medical code in the visit number 5.

5.5 Patient Editor

Patient Editor allows users to conduct what-if analyses (**T6**). There are two ways to invoke Patient Editor. First, users can select a patient in Patient List, and open a pop-up dialog of Patient Editor (see Figure 7). It provides a dedicated space for editing a selected patient's medical codes. Patient Editor presents each visit horizontally in a temporal manner and lists each visit's medical codes downward as shown in Figure 7. Second, users can convert Patient Detail into Patient Editor by simply choosing a context menu option. In doing so, users can maintain the context while editing the patient visits and medical codes if they were focusing on Patient Detail. However, users will lose the original version if they directly edit on Patient Detail. Since there is a

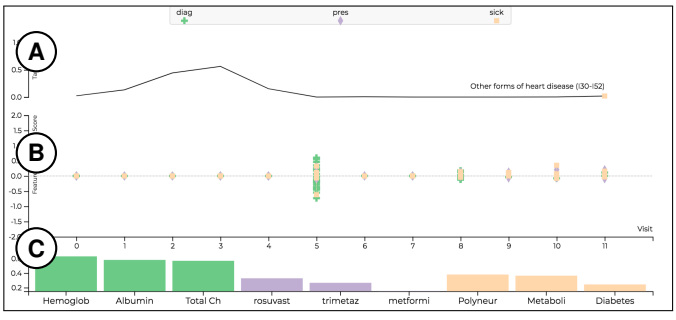

Fig. 6. Patient Detail shows a patient's information in three views: A) shows prediction scores over time. B) shows contribution scores of medical codes over time. C) shows the nine highest contributors.

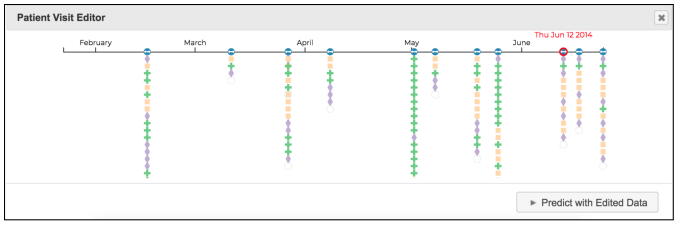

Fig. 7. In Patient Editor, users can conduct various what-if analyses on a selected patient. Users can create/edit/remove medical codes, change the dates, and retrain the model by updating contribution scores.

tradeoff between the two approaches, we implemented both features and allowed users to choose one at their convenience.

As shown in Figure 7, users can move the visit along the time axis to change the date. Users can also add new codes in to a visit, and they can remove existing ones. In some cases, users may feel that they need to steer the model towards their prior knowledge or hypotheses. In Patient Editor, users can provide feedback to the model (**T7**) by requesting to increase contribution scores of selected medical codes. In such activities mentioned above, users can test hypothetical scenarios. Once users complete the changes, the model returns the newly generated prediction scores over time as well as contribution scores overlaid on top of the original records. For example, users might have felt the need to update contribution scores of selected medical codes and move some visits to different dates (Figure 7). The results are shown in Figure 8. The prediction score significantly increased; in particular, prediction scores of final two increased as the red dotted line shows in Figure 8. The increase was due to increases in contribution scores of medical codes from the four most recent visits, which are shown as the right upward trends in connected code symbols.

6 Experiments

This section describes the settings and results of the experiments we conducted to evaluate the performance of our model. We conducted both quantitative and qualitative analyses using our model trained using the HIRA-NPS dataset introduced in Section 3.2 targeted for predicting two medical conditions, heart failure and cataract. The details of our data preprocessing pipeline are discussed in Appendix B.

6.1 Experimental Setup

Our models are implemented on Pytorch 0.3.1 [58]. We trained our model using Adam [41] on learning rates of [0.01, 0.001, 0.0001] and hidden state sizes of [64, 128, 256], and tested them on the validation set to obtain the best performing models. We used an Ubuntu 14.04 server equipped with two Nvidia Titan X GPUs to train our models.

6.2 Quantitative analysis

Models are quantitatively evaluated by two metrics; Area under the ROC Curve (AUC) and Average Precision (AP). These measures show robustness to data imbalance in positive/negative labels as they measure how successfully positive cases are ranked above negative cases.

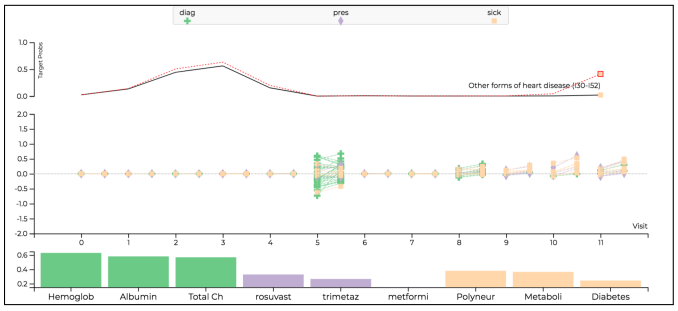

Fig. 8. Users conducted what-if analysis with edited contribution scores of medical codes and new dates. The result shows significant increase of prediction score and contribution scores of related medical codes.

Table 1. Model performances measured for medical predictions tasks

	(a) heart failure		(b) cataract	
Models	AUC	AP	AUC	AP
GRU	0.906	0.694	0.953	0.834
RETAIN	0.905	0.729	0.959	0.835
RetainEX w/o time	0.946	0.769	**0.975**	0.870
RetainEX	**0.954**	**0.818**	**0.975**	**0.878**

To further test our model, we implemented two baseline models for comparison: 1) GRU: We implemented a GRU model using the final hidden state, equivalent to β_T in our proposed model; 2) RETAIN: We implemented the original version of RETAIN.

We also tested the importance of adding time data to our model, which we name as RetainEX. We compare our model to an equivalent version that does not take in the time interval data which we propose. For the baseline models, we apply the same training and hyperparameter selection strategy as mentioned above.

We observed that our model outperforms the baseline models in all cases (see Table 1). The effect of adding an additional embedding matrix can be seen by comparing the original RETAIN model and our model that does not incorporate time interval data. Given the otherwise identical settings, this improvement in performance is due to having our model learn one type of embedding for computing the attention values and another for computing the final prediction output. Furthermore, we show that with the addition of time interval information the performance of the model increases even more across all settings. With the use of time data, our model can more accurately learn how to discriminate between important and unimportant visits.

6.3 Qualitative analysis

We also conducted an evaluation to determine whether the medical codes (treatment, diagnosis, and prescriptions) with high contribution scores for predicting heart failure (HF), are supported by general medical knowledge. Using the scheme introduced in Section 4.5, we generated a score vector $\mathbf{S}$ and an additional 1400-dimensional vector $\mathbf{C}$ for every case patient. $\mathbf{C}$ stores the total counts of each medical code per patient. After computing the vectors for every patient, we sum all $\mathbf{S}$'s and $\mathbf{C}$'s to obtain 1400-dimensional representations of the contribution scores and counts for the medical codes of all N patients, which we denote as $\mathbf{S}_{total}$ and $\mathbf{C}_{total}$. We normalize $\mathbf{S}_{total}$ in two different directions. Firstly, to identify common medical codes prevalent in most patients, we averaged all dimensions of $\mathbf{S}_{total}$ by N to obtain $\mathbf{S}_1$ (Table 2 in Appendix C). Secondly, to identify codes that are strongly associated with the development of heart failure, we divided each dimension of $\mathbf{S}_{total}$ by its corresponding $\mathbf{C}_{total}$ value to obtain $\mathbf{S}_2$ (Table 3 in Appendix C).

The top-5 $\mathbf{S}_1$ scores in diagnosis support the premise that hypertensive disease are associated with heart failure, as well as being a major cause of other diseases and comorbidities [36, 49, 76]. Hypertensive disease was the most frequently diagnosed co-morbidity in patients with heart failure (Table 2 in Appendix C). Likewise, ischaemic heart disease was also a major disease in patients with heart failure [23,57], as reflected by the relatively high $\mathbf{S}_1$ scores in the current study. Metabolic disorders, such as hemochromatosis, for which a relatively high $\mathbf{S}_1$ score was observed, were also shown to be likely to cause heart failure as a complication [6]. It was presumed that cerebrovascular disease would be diagnosed in a high number of patients with heart failure as hypertension is a characteristic of both diseases [4, 65]. Bisoprolol, a medicine ingredient for which a relatively high $\mathbf{S}_1$ score was recorded, is frequently prescribed for heart disease [21, 24], while aspirin and atorvastatin are commonly used to prevent it [19, 74, 81].

Carvedilol is a major prescription agent used to treat heart failure [20]. The results of this study demonstrated that medical codes involved in the prevention or treatment of heart failure (i.e., prescriptions) had relatively high scores in $\mathbf{S}_1$. Other medical codes with a relatively strong contribution score included obesity, confirming that it is a major risk factor for heart diseases [38]. Disorders of the thyroid gland are also known to cause heart failure [42]. These codes were found to have high scores in $\mathbf{S}_2$ (Table 3 in Appendix C). It was assumed that

isosorbide mononitrate and amlodipine besylate had relatively high $\mathbf{S}_2$ scores because they are used to treat hypertension, a major causative condition of heart failure. Heart failure is a clinical syndrome that is characterized by complicated pathophysiology. The results from the study show that our model is capable of identifying factors (*i.e.* medical codes) that are strongly associated with heart failure.

7 Case Study: Patients with Heart Failure

In this section, we provide a case study, developed and discussed by analyzing a subset of EMR data. To illustrate the story vividly, we introduce a fictitious character called Jane. Jane is a data analyst, who is a domain expert in medical field. She is very interested in analyzing patients with heart failure (HF) and determining treatment sequences that affect the onset of the disease.

Jane decided to conduct a predictive analysis using RetainVis with RetainEX trained by 63,030 patients (1:10 ratio between case and control) for the heart failure case study (see Section 6 for details). She pulled 5,730 patients diagnosed with heart failure in the latter half of a calendar year. She then launched RetainVis to see an overview of the patients in terms of contribution scores of 1,400 medical codes.

The initial overview showed a very interesting grouping in the upper right corner of Overview (see the highlighted area in Figure 1 (A)). Jane filtered patients by drawing a polygon area of interest over the region using the lasso tool. The initial selection provided 564 patients (F = 297) with very high prediction scores on average (.97), which indicates that the patients are explained well with RetainEX. She loaded the selection into Patient Summary. It showed the top three contributing diseases (comorbidities) as ischaemic heart disease, hypertensive disease, and cerebrovascular disease, all of which are known to be highly related to heart failures. In particular, the existence of hypertensive disease indicates its relevance to the S1 General HF group in Yan et al. [80]. The top three medications are bisoprolol hemifumarate, and trimetazidine. Bisoprolo is related to reducing hypertension, and trimetazidine is related to ischaemic heart disease. It was interesting to see aspirin among the top contributors as it is known to reduce the HF risk with potential side effects like kidney failure for long-term use.

Jane quickly broke down the group into a more granular level. The data points were subdivided into three subgroups, each of which tends to be cohesive within its group but separated from others. The first subgroup (N=201) showed the similar representation of what we saw with high hypertension and Bisoprolo (S1). The second group showed an interesting diagnosis called "syndrome of four (or Sasang) constitutional medicine" as one of the high contributing medical codes. The Sasang typology is a traditional Korean personalized medicine, which aims to cluster patients into four groups based on patient's phenotypic characteristics [5]. It was interesting to observe that a fair number of patients (N=230) showed the influence of this unique medical code. Recently, there have been studies investigating the relationship between the Sasang types and prediction of cardiovascular disease [10]. She thought it will be interesting to test such hypotheses later.

The third group showed another interesting cohort with relatively higher age (74.7 years old in average) than the other two groups (66.7 years old). In Patient Summary, Jane saw that the group is associated with hypertension and diseases of oesophagus. It has been reported that there might be relationship between heart disease (e.g., ischaemic heart disease) and diseases of oesophagus (e.g., Barrett's esophagus) [75]. The group also showed high contribution scores of bilirubin, suspected as a predictive marker of pulmonary arterial hypertension [73]. She conjectured that this group shows many severe diseases (mostly related to high blood pressure) with high prediction scores.

Jane decided to drill down into details in Patient List. She sorted the list by the number of visits then hovered over cerebrovascular disease (the top contributor of this group), and selected a patient with a very high volume of visits (N=150) over the period of six months. She observed cerebrovascular diseases recorded for almost every visit. By pulling the patient's detail in Patient Detail, she found that the patient is taking a variety of preventive medicines as top three contributors: glimepiride (anti-diabetic drug), pravastatin (prevent high cholesterol), and hydrochlorothiazide (prevent high blood pressure). By arranging the x-axis by dates, she also realized that the patient was prescribed the medicine periodically (once in every two weeks). The patient was also diagnosed with metabolic disorders nearly every visit. In summary, Jane could confirm many known stories about heart failures, where it is closely related to metabolic disorders, hypertension, and growing age.

Jane switched her gear to evaluate the performance of the predictive model. Since she in general believed that the model describes the heart failure prediction very well with associated comorbidities and medication as high contribution scores, she was curious of cases where the model failed. Could it be due to the data quality? She sorted the patients by prediction scores, and found three patients who were not predicted as HF (prediction score $< .5$). She selected a patient with the lowest score (.076). Interestingly, this patient did show the prevalence of aforementioned medical codes, such as hypertention, bilirubin, and aspirin, towards the end of June. However, there was a very unique aspect of this patient. There were major injuries recorded in May 20, namely head, leg, body injuries leading to medication prescriptions related to pain (e.g., tramadol) on next two visits. Also, the patient was diagnosed with arthrosis twice. Jane conjectured the mixture of major injuries with HF related diseases might be the issue. She promptly conducted a what-if analysis using Patient Editor. She removed injuries and related medications, and tightened dates between events towards the end of June. She selected hypertension, bilirubin, aspirin, and ischaemic heart disease, then chose the "increase the contribution score" option to retrain the model. The retrained model with new input increased the prediction score from .076 to .600 with hypertension as the highest contributor. She hypothesized that it will be difficult to perfectly predict HF when a patient is associated with parallel activities. She once again realized the danger of purely automatic solutions and the importance of collaboration between human and machine via visual analytics.

8 Discussion

In this section, we provide an in-depth discussion of our study. We start with our efforts to improve both the interpretability and interactivity of RNNs, and share lessons learned that can be applied to design similar RNN-based models and visual analytics applications. We also describe the type of needs medical domain experts have regarding deep learning-supported visualization tools. Finally, we discuss the limitations of using deep learning and visualization for predictive analysis.

8.1 Interactivity, Interpretability, and Model Performance

To open the black box of RNNs, we proposed a method of feature-level interpretation where the score of each individual feature represents its influence on a patient's future risk. To enable various what-if cases, we incorporated additional dimensions and embeddings to our model.

Although we observed the performance increase along with greater interactivity, this was actually the result of continuous trial and error. We initially tried to augment a time-decaying LSTM introduced from a recent study [2] to incorporate time data, only to experience a lower performance score compared to the vanilla model. Also, we had to test various loss functions and experimental settings to discover the current version that did not harm the attention distributions of other visits, not to mention having to add another embedding matrix.

Likewise, adding interpretability to a model while preserving its performance is also a challenging task. The strength of RNNs is that its intertwined structure that freely learns high-dimensional representations in the latent space provided by its hidden states. In this sense, our approach of improving interpretability using linear combinations can be seen as forcing the model to learn these representations at the expense of computational freedom. Thus, understanding the tradeoff between interactivity, interpretabiltiy and the performance of RNN models is crucial in designing a visual analytics tool. Target user tasks can be a key guidance to solve this deadlock. Our tasks derived from iterative discussion in Section 3.3 show an example.

One of the golden rules of data visualization is to preserve simplicity, which we discovered applies to the case of medical data as well. One expert expressed his thoughts of an ideal visualization tool for EMR data as a 'conversation partner' and the first step to fulfilling that role is to have visualization results as simple as possible. The domain expert

revealed that it is important for the model and results to be intepretable and interactive, but it also has to be easily explainable by design.

According to the expert, it is actually unlikely that clinicians would benefit much from the complex information provided by a machine. The contribution scores of each code and the predicted outcome presented by the model are, in a doctor's eyes, an additional piece of information that has to be looked at and verified. This might actually put extra burden to the doctor and hinder the process of decision making, instead of assisting it. This opinion was supported by another expert, who stated that some of the visualization methods were unintuitive and hard to interpret at first sight. This was contrary to our belief that presenting more information would lead to more precise diagnoses.

Both experts agreed that a more welcoming situation would be to develop a machine that simplifies the already complicated EMR data and pinpoints a number of points of attention, similar to our visit-level attention view. It is not a meticulous analysis tool that domain experts need, but an agent that can suggest interesting points of discussion by looking at the data from a different perspective, just as if it were a fellow expert. Of course, the machine should be able to sufficiently prove why it made such a prediction or emphasized on a certain visit of a patient, so interpretability still remains a prerequisite. However, that is only when the user asks the model to prove its prediction, and in general the level of visualization should remain as simple as possible.

This feedback led us to come up with different variations of our model that can be served for different purposes. The current complexity of our tool can be used to aid researchers who would like to freely explore the available data and conduct various what-if case analyses as seen from our case study in Section 7. Meanwhile, a more simplified version where only significant and anomalous events are highlighted can be adopted as an assistant tool for clinicians. The presented events may provide new insight to the domain expert that might have otherwise been overlooked. Thus, it is important that the designers and providers of such tools should be able to correctly identify their target users, and maximize the desired effects out of the given settings.

8.2 Towards Interactivity

Another important objective of our work was to apply various user interaction schemes to our visualization tool so that users can conduct various what-if analyses. To allow for a greater depth of user interaction, we added the following functions to the original model: (1) we used the interval information between visits as an additional input feature to our model, and (2) we introduced a retraining method using an additional embedding matrix to increase or decrease the contributions of individual codes according to the domain knowledge of the user. We also showed that not only do these additional functions ensure our proposed interaction features, but they have an auxiliary effect of improving the model's quantitative performance as well.

While making use of temporal data is an important concept that we present, here we would like to focus more on the retraining module. This strategy was especially effective in correcting the contribution scores that were incorrectly assigned to certain case and control patients. We selected one case patient and one control patient which were causing Type-II and Type-I errors. With the help of domain knowledge of medical experts, we could find out which codes were over- and underrepresented, and updated their contribution scores accordingly. Not only were we able to fix the prediction scores of the selected samples, but we also noticed that the average precision score of an independent test dataset increased from 0.812 to 0.814 as well. That is, our retraining scheme conducted at one or two samples ended up improving the overall performance of the model without affecting the model's integrity in computing attention scores for other samples.

The fact that retraining the contributions of two patients lead to improved model performance shows an example of human-computer interaction where not only does the machine provide results to the user, but in turn the user can also teach the machine by instilling domain knowledge. Such interactions can greatly improve the inherent problems of machine learning-based models, that the performance of a trained model does not improve until it is provided with additional training data.

In RetainVis, we were able to collect user inputs as contribution scores of medical codes for retraining. In other words, the feature-level interpretation can be used as an interaction handle, then the model can easily translate the user request using our updated model architecture. In this way, we were able to avoid the undesirable situation, where users need to directly update the model parameters, which can be challenging for domain experts. This study illustrates one way to reflect user's intent, namely using direct manipulation and menu selection on feature-level representation of data points.

8.3 Issues in Visualization and AI for Health

A major point of concern is the risk of the machine making false predictions. No matter how accurate a diagnosis prediction model may be, there is always the possibility that it will produce Type-II errors and fail to capture a serious condition of a patient. These are often heavily intertwined with life-or-death problems. Thus, solely relying on the information provided by a machine becomes risky because doctors have to take full responsibility of a patient's outcome. In addition, the performance metrics need to be more convincing. Though AUC and AP are proven to be effective metrics for measuring the performances in imbalanced datasets, a high score does not necessarily mean that a model makes a clear distinction between safe and suspected patients. While an ideal situation would be to have a threshold around 0.5 out of 1 to discriminate between positive and negative cases, we discovered that the threshold that maximizes the F-1 score of the predicted results was very low, near 0.2. This reflects a common problem in applying machine learning to medical prediction tasks, where a high score does not ensure that a model will not make a serious mistake.

Visualizations can be carefully used to communicate the performance of the model in a transparent way. Domain experts revealed that they often hesitate to accept what they see as facts because they are not able to tell various uncertainties propagated through the pipeline [63]. There might be artificial patterns created due to inconsistent EMR recording practice, which then can be amplified by incorrectly trained models. Even for some examples that were proven to be accurate, visualizations should indicate its inherent uncertainties involved. Thus, future researchers can investigate the design of uncertainty visualizations, especially when applying deep learning-based models for complex medical data.

Additionally, we learned that different tasks of the medical domain have to be modeled differently to produce satisfactory results. Another medical task that we did not include in this work is that of predicting the main sickness of a patient's next visit. Though the same settings were used, we discovered that even using the same number of input and output classes as in our proposed setting, RETAIN failed to outperform even the simplest baseline that returns the most frequent diagnosed sickness of each patient. While such problems can be left for future work, we would like to emphasize that in order for a machine learning-based model to prevail, a substantial amount of time and effort are required to tailor the problem setting and preprocess the available data.

9 Conclusion

In this study, we developed a visual analytics system called RetainVis by incorporating RetainEX into electronic medical datasets. In the process, we tackled the two most important challenges in visual analytics with deep learning: increasing interpretability and interactivity. Our iterative design process led us to improve interpretability as well as interactivity while maintaining its performance level against RETAIN. Our study shows that the design of RetainVis helps users to explore real-world EMRs, gain insights, and generate new hypotheses. We aim to extend our approach to more diverse medical records, including various measures from medical tests, sensor data from medical equipment and personal devices. We believe that the lessons learned from this study can better guide future researchers to build interpretable and interactive visual analytics systems for recurrent neural network models.

REFERENCES

[1] T. Arajo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy, A. Polnia, and A. Campilho. Classification of breast cancer histology images using convolutional neural networks. *PLOS ONE*, 12(6):1–14, 06 2017.

[2] 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 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 65–74, 2017.

[3] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. D. Jackel, U. Muller, and K. Zieba. Visualbackprop: visualizing cnns for autonomous driving. *CoRR*, abs/1611.05418, 2016.

[4] L. Caplan, P. Gorelick, and D. Hier. Race, sex and occlusive cerebrovascular disease: a review. *Stroke*, 17(4):648–655, 1986.

[5] H. Chae, J. Lee, E. S. Jeon, and J. K. Kim. Personalized acupuncture treatment with sasang typology. *Integrative Medicine Research*, 6(4):329–336, 2017.

[6] Z. Chati, F. Zannad, C. Jeandel, B. Lherbier, J.-M. Escanye, J. Robert, and E. Aliot. Physical deconditioning may be a mechanism for the skeletal muscle energy phosphate metabolism abnormalities in chronic heart failure. *American Heart Journal*, 131(3):560–566, 1996.

[7] Z. Che, Y. Cheng, S. Zhai, Z. Sun, and Y. Liu. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In *2017 IEEE International Conference on Data Mining*, pp. 787–792, 2017.

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

[9] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014.

[10] N. H. Cho, J. Y. Kim, S. S. Kim, and C. Shin. The relationship of metabolic syndrome and constitutional medicine for the prediction of cardiovascular disease. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 7(4):226–232, 2013.

[11] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, vol. 56, pp. 301–318, 2016.

[12] E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, and J. Sun. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1495–1504, 2016.

[13] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun. GRAM: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 787–795, 2017.

[14] 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 29*, pp. 3504–3512. Curran Associates, Inc., 2016.

[15] 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*, 24(2):361–370, 2017.

[16] J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, Dec 2013.

[17] J. Choo, C. Lee, C. K. Reddy, and H. Park. Weakly supervised nonnegative matrix factorization for user-driven clustering. *Data Mining and Knowledge Discovery*, 29(6):1598–1621, Nov 2015.

[18] S. Chung, C. Park, S. Suh, K. Kang, J. Choo, and B. C. Kwon. Re-VACNN: Steering convolutional neural network via real-time visual analytics. In *Future of Interactive Learning Machines Workshop at the 30th Annual Conference on Neural Information Processing Systems*, 2016.

[19] J. Cleland, I. Findlay, S. Jafri, G. Sutton, R. Falk, C. Bulpitt, C. Prentice, I. Ford, A. Trainer, and P. Poole-Wilson. The warfarin/aspirin study in heart failure (wash): a randomized trial comparing antithrombotic strategies for patients with heart failure. *American Heart Journal*, 148(1):157–164, 2004.

[20] J. N. Cohn, M. B. Fowler, M. R. Bristow, W. S. Colucci, E. M. Gilbert, V. Kinhal, S. K. Krueger, T. Lejemtel, K. A. Narahara, M. Packer, et al. Safety and efficacy of carvedilol in severe heart failure. *Journal of Cardiac Failure*, 3(3):173–179, 1997.

[21] P. De Groote, P. Delour, N. Lamblin, J. Dagorn, C. Verkindere, E. Tison, A. Millaire, and C. Bauters. Effects of bisoprolol in patients with stable congestive heart failure. *Annales de Cardiologie et d'Angeiologie*, 53(4):167–170, 2004.

[22] Q. Dou, H. Chen, L. Yu, L. Zhao, J. Qin, D. Wang, V. C. Mok, L. Shi, and P. A. Heng. Automatic detection of cerebral microbleeds from mr images via 3d convolutional neural networks. *IEEE Transactions on Medical Imaging*, 35(5):1182–1195, May 2016.

[23] R. N. Doughty, G. A. Whalley, G. Gamble, S. MacMahon, N. Sharpe, et al. Left ventricular remodeling with carvedilol in patients with congestive heart failure due to ischemic heart disease. *Journal of the American College of Cardiology*, 29(5):1060–1066, 1997.

[24] P. Dubach, J. Myers, P. Bonetti, T. Schertler, V. Froelicher, D. Wagner, M. Scheidegger, M. Stuber, R. Luchinger, J. Schwitter, et al. Effects of bisoprolol fumarate on left ventricular size, function, and exercise capacity in patients with heart failure: analysis with magnetic resonance myocardial tagging. *American Heart Journal*, 143(4):676–683, 2002.

[25] H. R. Ehrenberg, J. Shin, A. J. Ratner, J. A. Fries, and C. Ré. Data programming with ddlite: Putting humans in a different part of the loop. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pp. 13:1–13:6. ACM, 2016.

[26] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins. Progressive learning of topic modeling parameters: A visual analytics framework. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):382–391, Jan 2018.

[27] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li. Breast cancer multi-classification from histopathological images with structured deep learning model. *Scientific Reports*, 7(1):4172, 2017.

[28] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18 – 31, 2017.

[29] F. Heimerl, S. Koch, H. Bosch, and T. Ertl. Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2839–2848, Dec 2012.

[30] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, Nov 1997.

[31] F. Hohman, N. O. Hodas, and D. H. Chau. Shapeshop: Towards understanding deep learning representations via interactive experimentation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 1694–1699, 2017.

[32] B. Jin, C. Che, Z. Liu, S. Zhang, X. Yin, and X. Wei. Predicting the risk of heart failure with ehr sequential data modeling. *IEEE Access*, 6:9256–9261, 2018.

[33] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. P. Chau. Activis: Visual exploration of industry-scale deep neural network models. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):88–97, 2018.

[34] D. C. Kale, Z. Che, M. T. Bahadori, W. Li, Y. Liu, and R. C. Wetzel. Causal phenotype discovery via deep networks. In *American Medical Informatics Association Annual Symposium*, 2015.

[35] K. Kamnitsas, C. Ledig, V. F. J. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61–78, 2017.

[36] W. B. Kannel, W. P. Castelli, P. M. McNamara, P. A. McKee, and M. Feinleib. Role of blood pressure in the development of congestive heart failure: the framingham study. *New England Journal of Medicine*, 287(16):781–787, 1972.

[37] A. Karpathy, J. Johnson, and F. Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015.

[38] S. Kenchaiah, J. C. Evans, D. Levy, P. W. Wilson, E. J. Benjamin, M. G. Larson, W. B. Kannel, and R. S. Vasan. Obesity and the risk of heart failure. *New England Journal of Medicine*, 347(5):305–313, 2002.

[39] L. Kim, J.-A. Kim, and S. Kim. A guide for the utilization of health insurance review and assessment service national patient samples. *Epidemiology and Health*, 36:e2014008, 2014.

[40] L. Kim, J. Sakong, Y. Kim, S. Kim, S. Kim, B. Tchoe, H. Jeong, and T. Lee. Developing the inpatient sample for the national health insurance claims data. *Health Policy and Management*, 23(2):152–161, Jun 2013.

[41] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In

International Conference on Learning Representations, 2015.

[42] I. Klein and S. Danzi. Thyroid disease and the heart. *Circulation*, 116(15):1725–1735, 2007.

[43] B. C. Kwon, B. Eysenbach, J. Verma, K. Ng, C. D. Filippi, W. F. Stewart, and A. Perer. Clustervision: Visual supervision of unsupervised clustering. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):142–151, Jan 2018.

[44] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, Jan 2017.

[45] B. C. Kwon, S.-H. Kim, S. Lee, J. Choo, J. Huh, and J. S. Yi. Visohc: Designing visual analytics for online health communities. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):71–80, 2016.

[46] B. C. Kwon, J. Verma, and A. Perer. Peekquence: Visual analytics for event sequence data. *ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, 2016.

[47] H. Lee, J. Kihm, J. Choo, J. T. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31(3):1155–1164, 2012.

[48] T. Y. Lee, A. Smith, K. Seppi, N. Elmqvist, J. Boyd-Graber, and L. Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies*, 105:28 – 42, 2017.

[49] D. Levy, M. G. Larson, R. S. Vasan, W. B. Kannel, and K. K. Ho. The progression from hypertension to congestive heart failure. *Journal of the American Medical Association*, 275(20):1557–1562, 1996.

[50] H. Lin, S. Gao, D. Gotz, F. Du, J. He, and N. Cao. Rclens: Interactive rare category exploration and identification. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2018.

[51] Z. C. Lipton, D. C. Kale, C. Elkan, and R. C. Wetzel. Learning to diagnose with lstm recurrent neural networks. *International Conference on Learning Representations*, 2015.

[52] Z. C. Lipton, D. C. Kale, and R. C. Wetzel. Phenotyping of clinical time series with LSTM recurrent neural networks. In *Workshop on Machine Learning in Healthcare at the 29th Annual Conference on Neural Information Processing Systems*, 2015.

[53] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):91–100, 2017.

[54] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1903–1911, 2017.

[55] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu. Understanding hidden memories of recurrent neural networks. In *IEEE Conference on Visual Analytics Science and Technology*, 2017.

[56] P. Nguyen, T. Tran, N. Wickramasinghe, and S. Venkatesh. *mathttDeepr*: A convolutional net for medical records. *IEEE Journal of Biomedical and Health Informatics*, 21(1):22–30, Jan 2017.

[57] M. Ozbaran, S. B. Omay, S. Nalbantgil, H. Kultursay, K. Kumanlioglu, D. Nart, and E. Pektok. Autologous peripheral stem cell transplantation in patients with congestive heart failure due to ischemic heart disease. *European Journal of Cardio-Thoracic Surgery*, 25(3):342–350, 2004.

[58] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *The Future of Gradient-based Machine Learning Software and Techniques Workshop at the 31st Annual Conference on Neural Information Processing Systems*, 2017.

[59] N. Pezzotti, T. Hllt, J. V. Gemert, B. P. F. Lelieveldt, E. Eisemann, and A. Vilanova. Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):98–108, Jan 2018.

[60] A. Prakash, S. Zhao, S. A. Hasan, V. V. Datla, K. Lee, A. Qadir, J. Liu, and O. Farri. Condensed memory networks for clinical diagnostic inferencing. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 3274–3280, 2017.

[61] P. E. Rauber, S. G. Fadel, A. X. Falco, and A. C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, Jan 2017.

[62] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164 – 175, 2017.

[63] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim. The Role Of Uncertainty, Awareness, And Trust In Visual Analytics. *Visualization and Computer Graphics, IEEE Transactions on*, 22(1):240–249, 2016.

[64] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[65] K. Shimada, A. Kawamoto, K. Matsubayashi, and T. Ozawa. Silent cerebrovascular disease in the elderly. correlation with ambulatory pressure. *Hypertension*, 16(6):692–699, 1990.

[66] S. Simon, S. Mittelstädt, D. A. Keim, and M. Sedlmair. Bridging the gap of domain and visualization experts with a liaison. In *Eurographics Conference on Visualization 2015*, pp. 127–131, 2015.

[67] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[68] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg. Direct-manipulation visualization of deep networks. In *International Conference on Machine Learning*, 2016.

[69] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. Breast cancer histopathological image classification using convolutional neural networks. In *2016 International Joint Conference on Neural Networks*, pp. 2560–2567, 2016.

[70] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):667–676, 2018.

[71] Q. Suo, F. Ma, G. Canino, J. Gao, A. Zhang, P. Veltri, and A. Gnasso. A multi-task framework for monitoring health conditions via attention-based recurrent neural networks. In *American Medical Informatics Association Annual Symposium*, 2017.

[72] Q. Suo, H. Xue, J. Gao, and A. Zhang. Risk factor analysis based on deep learning models. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 394–403, 2016.

[73] Y. Takeda, Y. Takeda, S. Tomimoto, T. Tani, H. Narita, and G. Kimura. Bilirubin as a prognostic marker in patients with pulmonary arterial hypertension. *BMC Pulmonary Medicine*, 10(1):22, 2010.

[74] D. Tousoulis, C. Antoniades, C. Vassiliadou, M. Toutouza, C. Pitsavos, C. Tentolouris, A. Trikas, and C. Stefanadis. Effects of combined administration of low dose atorvastatin and vitamin e on inflammatory markers and endothelial function in patients with heart failure. *European Journal of Heart Failure*, 7(7):1126–1132, 2005.

[75] P. Tsibouris, M. T. Hendrickse, P. Mavrogianni, and P. E. Isaacs. Ischemic heart disease, factor predisposing to barretts adenocarcinoma: A case control study. *World Journal of Gastrointestinal Pharmacology and Therapeutics*, 5(3):183, 2014.

[76] H. A. Tyroler, S. Heyden, A. Bartel, J. Cassel, J. C. Cornoni, C. G. Hames, and D. Kleinbaum. Blood pressure and cholesterol as coronary heart disease risk factors. *Archives of internal medicine*, 128(6):907–914, 1971.

[77] F. Wang, H. Liu, and J. Cheng. Visualizing deep neural network by alternately image blurring and deblurring. *Neural Networks*, 97:162–172, 2018.

[78] Y. Wang, Z. Luo, and P.-M. Jodoin. Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters*, 96:66 – 75, 2017.

[79] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mané, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):1–12, 2018.

[80] C. Yan, Y. Chen, B. Li, D. Liebovitz, and B. Malin. Learning clinical workflows to identify subgroups of heart failure patients. In *AMIA Annual Symposium Proceedings*, vol. 2016, p. 1248, 2016.

[81] S. Yusuf, J. Wittes, and L. Friedman. Overview of results of randomized clinical trials in heart disease: Ii. unstable angina, heart failure, primary prevention with aspirin, and risk factor modification. *Journal of the American Medical Association*, 260(15):2259–2263, 1988.

[82] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pp. 818–833, 2014.

[83] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *International Conference on Learning Representations*, 2017.

Appendices

A Model Description

In this section, we will describe how RNNs and bidirectional RNNs, our building blocks, function. We also provide a larger overview of our proposed RetainEX.

A.1 Recurrent neural networks

RNNs have been used in numerous prediction tasks of different domains that require processing sequential data. A typical RNN model takes in a sequence of $m-$dimensional vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T$ and returns as output the same number of $n-$dimensional vectors $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T$, which are defined as hidden states. The hidden states are computed sequentially as

$$\mathbf{h}_t = \tanh\left(\mathbf{W}\mathbf{v}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}\right), \tag{11}$$

$$\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T = \mathrm{RNN}\left(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_T\right), \tag{12}$$

where $\mathbf{W} \in \mathbb{R}^{n\times m}$, $\mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$ are all learnable parameters. For binary prediction tasks, an additional parameter $\mathbf{w}_{out} \in \mathbb{R}^n$ is multiplied to either the last hidden state or the sum of all hidden states. The result is a scalar value, on which we apply the sigmoid function to obtain a continuous value between 0 and 1. We use the term RNN comprehensively and include its variants such as long-short term memory (LSTM) [30] and gated recurrent units (GRU) [9].

A.2 Large-scale diagram of RetainEX

(Figure in Next Page)

B Data preprocessing

The primary purpose of RETAIN is to provide interpretation results on data-driven medical prediction tasks. To apply our visualization framework to a case of medical prediction, we set two binary prediction tasks: predicting a patient's future diagnosis of (a) heart failure and (b) cataract. Our specific goal is to observe the medical records of a patients for the first six months to predict whether he/she will become diagnosed with that condition in the near future. For each task, we create a case set which consists of patients whose first diagnosis of the target condition occurred *after* the end of June. We discard all visits made after June, and remove patients who has made less than 5 visits. For each patient in the case set, we create a control set consisting of 10 patients who belong to the same gender and age groups and have a similar number of visits. We assign target labels of 1 and 0 to the case and control patients respectively. Thus, each batch contains 11 patients. We result in 5,730 batches for heart failure and 10,692 batches for cataract. We split each dataset into train, validation and test sets with a ratio of 0.65/0.1/0.25.

Each visit of a patient contains a timestamp and the codes related to the medical treatment, prescriptions, and diagnosed conditions of a patient. However, the number of codes are too diverse for our model to properly handle, and thus an additional step for reducing the total number of codes was taken. For diagnosis codes, we simply categorized each specific code according to the 268 mid-level categories according to KCD-9. However, since there were no provided classification schemes for treatment and prescription codes, for each type of code we selected the n-frequent codes that account for at least 95% of the entire data and discarded the rest. We were able to reduce more than 7,000 treatment codes to 500 and 3,800 prescription codes to 632 while preserving 94.7% of the original data. This enables us to represent all the codes associated to each visit in a 1,400-dimensional binary vector.

C Qualitative results

(Figure in Next Page)

Code Type	Code Name	Mean Score
Diagnosis	Hypertensive diseases	0.532
	Diseases of oesophagus, stomach and duodenum	0.218
	Ischaemic heart diseases	0.200
	Metabolic disorders	0.118
	Cerebrovascular diseases	0.091
Treatment	Outpatient care - established patient	0.286
	Glucose (Quantitative)	0.097
	Hematocrit	0.094
	Prothrombin Time	0.088
	Electrolyte examination (Phosphorus)	0.076
Prescription	Bisoprolol hemifumarate	0.156
	Aspirin (enteric coated)	0.081
	Atorvastatin (calcium)	0.059
	Carvedilol	0.047
	Rebamipide	0.032

Table 2. Top-5 contribution scores averaged over the total number of patients.

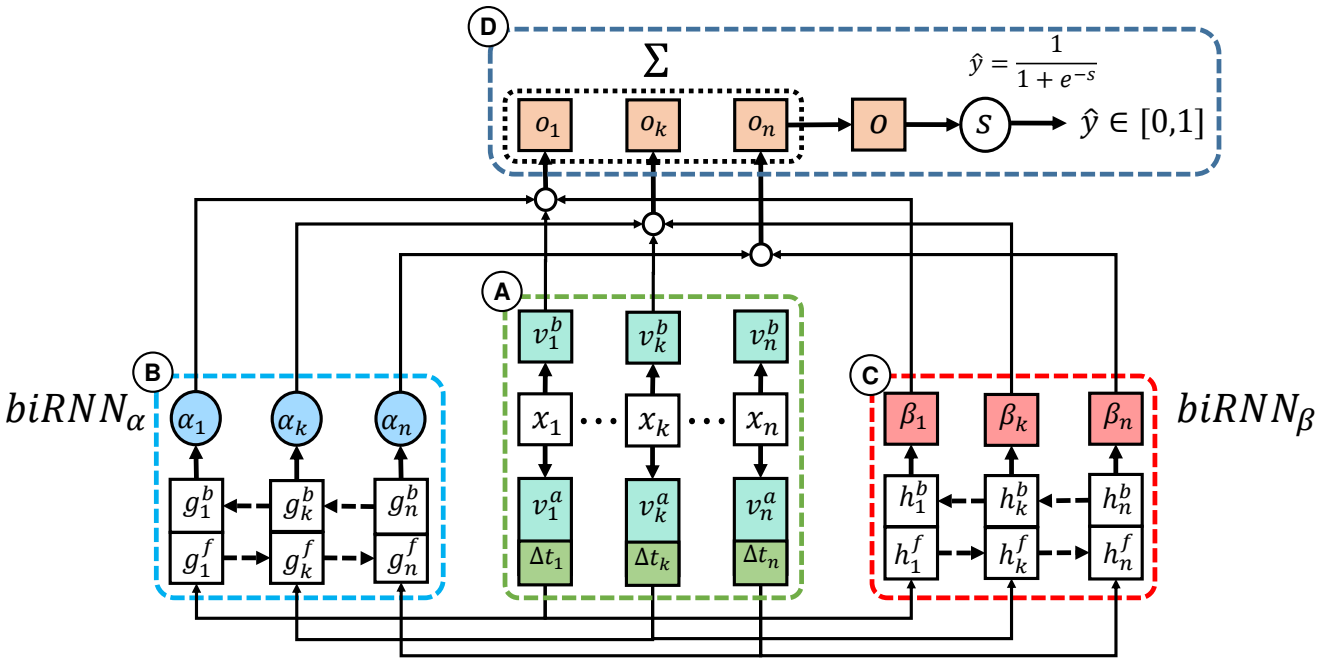

Fig. 9. Overview of the newly updated RetainEX (our version). (A) Using separate embedding matrices, the binary vectors $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are transformed into embedding vectors $\mathbf{v}^a{}_1, \ldots, \mathbf{v}^a{}_T$ and $\mathbf{v}^b{}_1, \ldots, \mathbf{v}^b{}_T$, with time interval information appended to the former. (B) $\mathbf{v}^a{}_1, \ldots, \mathbf{v}^a{}_T$ are fed into a bidirectional RNN to produce scalar weights α. (C) $\mathbf{v}^a{}_1, \ldots, \mathbf{v}^a{}_T$ are fed into another biRNN, this time to generate vector weights β. (D) α, β and $\mathbf{v}^b$ are multiplied over all timesteps, then are summed to form a single vector $\mathbf{o}$, which goes through linear and nonlinear transformation to produce a probability score $\hat{y}$.

Code Type	Code Name	Mean Score
Diagnosis	Obesity and other hyperalimentation	0.206
	Other infectious diseases	0.169
	Ischaemic heart diseases	0.156
	Hypertensive diseases	0.134
	Disorders of thyroid gland	0.119
Treatment	Prothrombin Time	0.299
	24hr blood pressure examination	0.278
	CA-19-9	0.253
	CK-MB	0.198
	Fibrinogen examination (functional)	0.185
Prescription	Bisoprolol hemifumarate	0.523
	Isosorbide mononitrate	0.243
	Amlodipine besylate	0.210
	Mmorphine sulfate	0.164
	Carvedilol	0.157

Table 3. Top-5 contribution scores averaged over the total number of occurrences.