This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

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 consisting 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 the patient cohort 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 Details*. Users can open (D) *Patient Editor* to conduct a what-if analysis, and (E) *Patient Details* shows the updated results.

Abstract— We have recently 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 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 methods 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 newly improved, interpretable, and interactive RNN-based model called RetainEX 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 individual medical codes contribute to making risk predictions, using EMRs of patients with heart failure and cataract 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 that aim to design an interpretable and interactive visual analytics tool for RNNs.

Index Terms—Interactive Artificial Intelligence, XAI (Explainable Artificial Intelligence), Interpretable Deep Learning, Healthcare

1 Introduction

- *Bum Chul Kwon is with IBM T.J. Watson Research Center. E-mail: bumchul.kwon@us.ibm.com.*
- *Min-Je Choi, Joanne Taery Kim, and Jaegul Choo, who is the corresponding author, 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.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

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 events, 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 techniques. 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 EMRs more accessible 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, there exist many challenges to overcome before the full adoption by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

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 task is to predict the risk of a patient's future diagnosis in heart failure and cataract, based on information from previous medical visits in our EMR dataset. Our design study involved iterative design, assessment, and discussion activities between medical experts, artificial intelligence scientists, and visual analytics researchers. After we characterized users' tasks, we designed, implemented, and evaluated a visual analytics tool called RetainVis with an 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 simultaneously increase interactivity and interpretability. Various visualizations coupled with the new model allow users to observe the patterns, 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.

The following three items summarize our main contributions:

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 four different perspectives. Section 3 introduces our target user, data, a prediction variable, and user tasks. Section 4 reviews the backbone model (RETAIN) and our new model (RetainEX) with a number of new features for our predictive model. Section 5 introduces the novel features of our visual analytics system (RetainVis). Section 6 shows the quantitative and qualitative experiments we conducted using a real EMR dataset on RetainEX and other RNN-based models. Section 7 shows a case study and Section 8 provides lessons, limitations, and implications learned from our study. Lastly, we conclude this study with future work in Section 9.

2 Related Work

This section reviews previous studies using four axes on which our work rests: deep learning applications for EMR data, visualization techniques of black-box models, machine learning platforms that allow user interactivity, and the interpretability of machine learning models.

Deep learning for electronic medical records. Though the most prevalent use of deep learning techniques in medical domains is to predict diagnosis of a disease, such as breast cancer [1, 28] and brain tumor [29, 37] by training models on medical images, there has also been an increase in deep learning applications for longitudinal EMR data. RNN-based models have been extensively used for tasks such as patient diagnosis [65, 76], risk prediction [12, 16, 34], representation learning [13, 14] and patient phenotyping [9, 36, 56], outperforming rule-based and conventional machine learning baselines.

An 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 [15] 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 exist other models for EMR data that suggest interpretability such as Dipole [59], the level of interpretability is limited to each visit, thus failing to provide a complete decomposition of the prediction as RETAIN. For this reason, we use the RETAIN framework for ensuring interpretability of our tool.

Visualization of deep learning models. A major concern in the application of deep learning models is the 'black-box' nature. As a result, many approaches have been investigated for visualizing the dynamics in various types of neural networks. Especially for vision applications where convolutional neural networks (CNNs) have enjoyed a great success, many visualization methods such as heatmaps [85], blurring [79], dimensionality reduction [57, 66] of the filters and activation maps obtained during computation and backpropagation, and visualizing the model structure itself [82] were used. This led to a large number of studies dedicated to developing visualization frameworks that help users better understand their networks [19, 33, 35, 64, 74].

Compared to CNNs, RNNs have received less attention in visualization, mainly because of its intertwined structure and its popularity in text data analysis. Though it is possible to visualize the activations of hidden state cells [39, 61, 75], 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.

Interactive machine learning platforms. A topic of emerging importance in the visual analytics field is to integrate machine learning models with various user interactions [68]. Instead of passively observing the outcomes of machine learning models visually, users can make updates to the inputs, outputs, or both, which can 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 [24, 30, 53], topic modeling [17, 25, 50], dimensionality reduction [6, 46] and clustering [18, 45, 49]. However, there are only a small number of studies that apply user interaction to tasks requiring deep learning models, such as object segmentation [81]. To the best of our knowledge, our work is one of the first to apply such user interaction to RNN-based models for medical tasks, supporting direct interaction with the visualized results computed from a deep learning model.

Interpretability of deep learning models. The definition of model "interpretability" has not been fully established. Model interpretation can be realized in several forms. For instance, the weights of a logistic regression model can show what the model has learned. In addition, 2-D projection of word embeddings can show how the model interprets each word by showing its distance from others. Likewise, we can illustrate interpretability using the linear or nonlinear relationship between inputs and outputs, defined by a learned model. In case of deep learning models, it is difficult to describe the relationship between the input and the output [54]. In this paper, RetainEX aims to achieve this notion of interpretability, similar to weights of logistic regression models.

Prior approaches attempted to resolve the interpretability issue in deep learning models. Visualizing the learned representation is one approach to understand the relationship between the input and the output, as discussed in Section 2. For example, explanation-by-example, such as 2-D projection of word embeddings using t-SNE [60] or PCA, can explain model's interpretation of data by showing which words are closely located in the latent space. An alternative form of interpretation is model-agnostic approaches such as partial dependence plot [27], Shapley values [71] and LIME [67], where they provide some form of explanation as to how a set of features affect the model output.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

Deep learning models provide interpretable results by showing attention [3, 58]. For example, we can imagine a RNN-based model that predicts a sentiment score based on a sentence (i.e., a sequence of words). Given a sentence, the model can compute and assign a score per each word. This score represents importance or contribution that is related to the predicted sentiment score. In our diagnosis risk prediction task, it is natural to think that particular visits of a patient are more important than the rest to predict whether the patient will be diagnosed with heart failure in the future. Therefore, by using the attention mechanism, we can train our model to assign greater weights to more important visits and use these weighted visits for the prediction task. Attention has proven to improve the performance of many sequence-to-sequence based deep learning tasks such as machine translation [23], question answering [83], and speech generation [80].

Inspired by the approaches, we aim to increase the interpretability of RNN-based model by using the attention mechanism as well as visualization methods in the development of RetainEX and RetainVis.

3 Users, Data, and Tasks

This section describes target users, input data, and 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 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). The experts often want to conduct what-if analysis on 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) [41], 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 [42]. 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 Predicting Diagnosis Risk

Of the various types of tasks in the medical domain where machine learning and deep learning methods can be applied, we chose the task of diagnosis risk prediction. Our medical domain experts were especially interested in the task of predicting whether a patient would later become diagnosed with illnesses such as heart failure using prior visit information. Therefore, we formulated a task setting where we first observe all visits of a patient who has not yet been diagnosed with a target illness (e.g., heart failure), and then predict whether he or she becomes diagnosed with that sickness during a visit in a latter stage, presumably within the next six months. This problem becomes a binary classification task over sequential data, which is a common setting in sequential neural network models as the one we will propose.

3.4 User Tasks

This section reports 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 by closely following a design study methodology [70]. 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. [73] suggests.

The following list shows the main user tasks:

T1: **View Patient Demographics and Medical Records Summary**. Users gain an overview of patients with respect to their demographic characteristics and medical history. The goal of this task is for users to understand the overall distribution of ages, gender, medical history. This enables users to understand the patient groups and to select a subset of them based on their interest.

T2: **Select Interesting Patient Cohorts**. Users test their hypotheses against prior knowledge on specific patient cohorts. In particular, they want to define the cohorts based on various patient attributes.

T3: **View 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: **Investigate Single Patient History**. Users investigates a patient's history, especially which visits (i.e., sequence and timing) and medical codes (i.e., event type) contribute to the prediction scores. Users compare contribution scores between medical codes and visits.

T5: **View Contributions of Medical Records for Prediction**. Users want to understand why each prediction is made based on patients' visits and medical records. In particular, users want to understand the reason by showing the relationship between inputs (patient records) and outputs (prediction scores).

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 Predictive Model** by viewing summary of prediction scores and providing feedback on contribution scores of visits and medical codes. Users want to check whether the model 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 can reveal interpretable outcomes. Thus, we chose the state-of-the-art, interpretable RNN-based model, called RETAIN [15]. However, 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 fulfills users' needs together with RetainEX.

4 Model Description

This section describes 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 article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

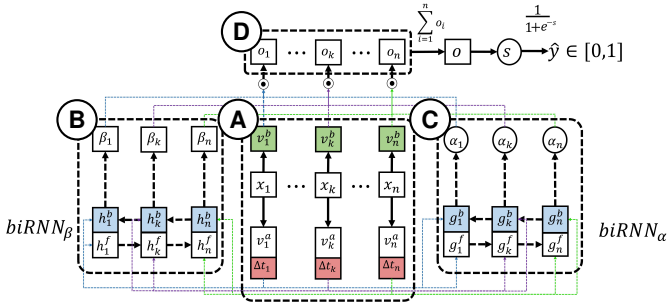

Fig. 2. Overview of RetainEX. See Appendix A for a larger diagram. Our modifications to the original RETAIN model are highlighted by three colors: (i) red (time information, Section 4.4), (ii) blue (bidirectional RNNs, Section 4.3), and (iii) green (an additional embedding matrix, Section 4.5). (A) Using separate embedding matrices, a binary vector $\mathbf{x}_t$ is represented as embedding vectors $\mathbf{v}_t^a$ and $\mathbf{v}_t^b$, with time interval information appended to the former. (B, C) $\mathbf{v}_t^a$ is fed into two biRNNs to obtain scalar α and vector β attention weights (D) α, β and $\mathbf{v}^b$ are multiplied over all timesteps, then summed to form a single vector o, which is linearly and nonlinearly transformed to a probability score $\hat{y}$.

this paper, we focus on using such sequential data on a prediction task, 'learning to diagnose (L2D)' [55], where a model observes the visits of a patient and returns a prediction score indicating the probability of the patient being diagnosed with a target disease in near future.

4.2 RetainEX: Interactive temporal attention model

As Figure 2 (A) shows, our model takes 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 $\mathbf{W}_{emb}^a \in \mathbb{R}^{m \times C}$ and $\mathbf{W}_{emb}^b \in \mathbb{R}^{m \times C}$ to convert the binary vectors into continuous vectors. We obtain a representation vector for each visit as $\mathbf{v}_t^a = \mathbf{W}_{emb}^a \mathbf{x}_t$. 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 and use the three different time values (see details in Section 4.4) per vector $\mathbf{v}_t^a$.

Figures 2 (B) and (C) represent the bidirectional RNNs that take in the time-attached visit representations and return attention values (e.g., contribution scores) of different scales. We follow the original RETAIN settings and compute two attention types, α and β. For the tth visit, α_t is a single value that represents the importance of the particular visit. Meanwhile, β_t is a m-dimensional vector that represents the importance of each medical code within a particular visit. As α and β are two separate attention types, our model uses two RNN modules. We included the details of how RNNs are computed in Appendix A.1.

We also modified the unidirectional RNN modules to bidirectional ones, as discussed in Section 4.3. 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 $e_t = \mathbf{w}_\alpha \left[\mathbf{g}_t^f; \mathbf{g}_t^b\right]$. 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 $\mathbf{W}_\beta \in \mathbb{R}^{m \times 2m}$ and return an m-dimensional vector β_t for the t-th timestep as $\beta_t = \mathbf{W}_\alpha \left[\mathbf{h}_t^f; \mathbf{h}_t^b\right]$.

Once we obtain both alpha and beta values, we multiply these values with the other set of embedding vectors, $\mathbf{v}_1^b, \ldots, \mathbf{v}_T^b$ as in Figure 2 (D), and add up the values to obtain the context vector $\mathbf{o}$ with $\odot$ indicating elementwise multiplication of two vectors. Lastly, we compute the *final contribution score*, $s = \mathbf{w}_{out}^T \mathbf{o}$. This scalar value is transformed to compute a prediction value $\hat{y} = \frac{1}{1+e^{-s}}$, ranging between 0 and 1 where $\mathbf{w}_{out} \in \mathbb{R}^m$. The predicted value indicates the diagnosis risk of a patient, with a value closer to 1 indicating a higher risk. We train our model by optimizing all parameters to minimize the cross-entropy loss, $\mathscr{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)$, with y_i as the target value for the i-th patient among all patients (N).

4.3 Bidirectionality

Compared to traditional RNNs, which process the input sequence in one direction from the beginning to the end, bidirectional RNNs (biRNNs) introduce another set of hidden state vectors computed in a reverse order from the end to the beginning.

In EMR-based diagnosis, clinicians can observe a patient's history in a chronological order to see how the patient's status progresses over time and also trace backward in a reverse order from the end to identify possible cues that may strengthen or weaken their 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. The structure of a bidirectional RNN is discussed in Appendix A.2.

4.4 Data with non-uniform time intervals

Though general RNNs do not consider time intervals between visits, the temporal aspect is a key to the disease diagnosis. For instance, a burst of the same events over a short time period may forebode the manifestation of a serious illness, while a long hibernation between events may indicate that they 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 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 calculate different time representations of a single interval, which are (1) Δt_i (the time interval itself), (2) $1/\Delta t_i$ (its reciprocal value), and (3) $1/\log(e+\Delta t_i)$ (an exponentially decaying value). The first representation was introduced in [15] where time interval information was incorporated to RETAIN, and the latter two were proposed by [4]. These three values are concatenated to the input vectors of each step, to enrich the information for our model. We added the three representations of time intervals because our model can learn to use multiple types of time information and their contributions to prediction results. Experimental results in Section 6 show that the addition of time information significantly improves predictive performance.

4.5 Understanding the interpretability of RetainEX

In this section, we show how we achieve interpretability by computing contribution scores using the attention mechanism in RetainEX.

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. Each input vector $\mathbf{x}_t$ has a linear relationship with the final contribution score s. Thus, we can formulate an equation that measures the contribution score of the code c at timestep t to s by reformulating the aforementioned equations as $s_{t,c} = \alpha_t \mathbf{w}_{out} \left(\mathbf{W}_{emb}^b[:,c] \odot \beta_t\right)$, 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 contribution score is the contribution score of code c at timestep t as described in the above equation. We can derive a visit-level contribution score s_t by aggregating contribution scores of codes for each visit as $s_t = \sum_{c \in \mathbf{x}_t} s_{t,c}$.

T3: Summary of selected patients. It is possible to create a vectorized representation of each patient using the learned contribution scores. We assign a 1400-dimensional zero vector $\mathbf{S}$ to each patient, compute all individual contribution scores for all codes in every visit that a patient had, and add the contribution score of each one code (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 (see details in Section 6.1). The resulting $\mathbf{S}$ can be seen as a patient embedding whose sum of elements and direction each indicate the predicted diagnosis risk and distribution of input features. We later use these vectors to create the scatter plot view in fig. 1 (A) for exploratory analyses.

4.6 Interactions featured in RetainEX

We increase the interpretability of the contribution score by allowing users to experiment with how outputs change according to input

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

changes in an interactive manner. We provide three ways to edit inputs: adding or removing input codes, modifying visit periods, and modifying the contribution scores of individual codes.

T6: Conducting what-if case analysis. Adding or removing codes 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. For modification of time intervals, once a time interval Δt_i changes, the corresponding three time values are also updated, and put into the model for recomputation. It is a simple and effective scheme for incorporating temporal information into the model, which also guarantees improved performance.

T7: Evaluating and steering predictive model. This interaction allows users to provide feedback onto contribution scores of individual visits or codes based on their domain expertise. While the earlier two types of interaction are straightforward in that we modify the inputs and insert them into the model to obtain new results, the third type of interaction is more complex since the model now has to update the learned parameters 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) compute the final outputs by multiplying itself with the alpha and beta values and then summing up across all timesteps to form a single context vector $\mathbf{o}$. As we want to change the contributions of specific code(s) at a particular visit without changing the alpha and beta attentions at other visits, we formulate an optimization problem where we minimize $\mathscr{L}_{retrain} = e^{-s_{pos}+s_{neg}}$ 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 is equivalent to performing a number of gradient descent operations to the parameters, which we restrict to $\mathbf{W}_{emb}$.

Our loss function has to take a monotonically increasing form while maintaining a positive value. We also have to train our model to increase/decrease a number of contribution scores at the same time. As can be seen in the paper, minimizing $\mathscr{L}_{retrain}$ is equivalent to maximizing positive contributions while minimizing negative contributions.

Though nonlinear functions such as sigmoid and tanh functions can be used for computing the loss, we chose the exponential function. When having to reduce negative contribution scores, we discovered that the input value $-s_{pos}+s_{neg}$ is often a real number larger than 2. While this would result in saturated gradients close to 0 for the aforementioned nonlinear functions, the exponential function would get a high gradient value from this calculation. This value combined with an adequate learning rate can optimize the parameters of the embedding matrix.

To preserve the overall attention distribution while changing the weights of specific medical codes, the embeddings used to calculate alpha and beta values need to be separated from the embeddings used for retraining. Thus, 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 the weights are multiplied to the second set for the final outputs. Due to the relaxation, we can control the influence of individual codes without altering the overall distribution of attention with respect to $\mathbf{W}^b_{emb}$.

While our retraining scheme helps improve the performance of our model, we also want to maintain the real-time interactivity. The retraining process is completed in real-time as the only parameters that are actually modified during the process are the weights from $\mathbf{W}^b_{emb}$. From a deep learning prospective, backpropagating the weights of a single embedding matrix from a single data sample can be done in milliseconds. Throughout experiments, we discovered that retraining was the most effective when optimized for around 10 to 20 iterations with a learning rate of 0.01. It took an average of 0.015 seconds to retrain the model according to the data of a patient with 20 visits where 5 codes were modified. Due to the simplicity of the scheme, the model was updated within a second after 20 iterations of retraining.

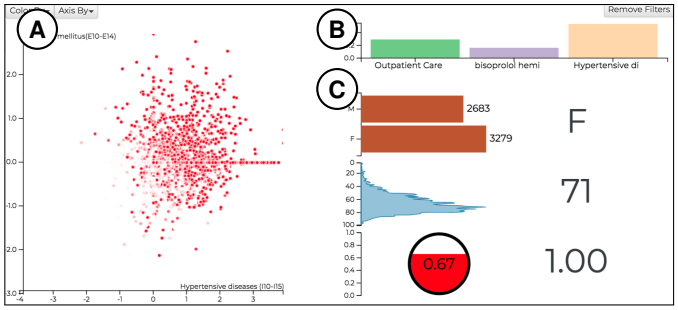

Fig. 3. *Overview* shows all patients in (A) a scatter plot; (B) A bar chart shows the top three contributors and their mean scores; (C) Three charts (i.e., bar chart, area chart, and circle chart) show the mean and distribution of gender, age, and predicted diagnosis risks, respectively.

5 RetainVis: Visual Analytics with RetainEX

This section introduces visualization and interaction features that integrate RetainEX and describe how the design fulfills user tasks.

5.1 Overview

Overview aims to provide a summary of patients with respect to their medical codes, contribution scores, and predicted diagnosis risks (**T1**). To derive this overview, we use $\mathbf{S}$ (see Section 4.5 for details), 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 in order to projects patients on 2-D space. Figure 3 (A) shows that Overview depicts patients' differences by 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), as Figure 3 (A) shows. For instance, users can map predicted diagnosis risks to a linear color scale of white to red (0 to 1) as shown in Figure 3 (A). 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 (A) 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 predicted risks 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. This approach has limitations, where it only shows three measures at a time. We could have fit more bars by showing narrower, horizontal bars. It is certainly a direction that future designers can implement. In this particular implementation, we wanted to maintain the consistency between other bar charts in other views, where the bar height was consistently used as an indicator for contribution scores. The contribution scores were computed by patient-wise mean of the corresponding codes in score vectors of patients ($\mathbf{S}$). Users can see the distributions of age and gender in gender bar chart and age area chart, respectively. Prediction circle chart shows the mean predicted diagnosis risk as the gauge filled in the circle. This particular icon is consistently used to show individual patient's predicted diagnosis risk in Patient List as well. The bottom right corner of Overview shows gender, age, and predicted risk of a selected/highlighted patient.

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. The scatter plot view initially shows 2D projection of patients using a nonlinear manifold learning technique, t-SNE. In scatter plot, users can draw a polygonal area with a lasso drawing tool. Our interaction approach, namely Lasso-selection in 2D projection derived from multidimensional data, has limitations. The 2D projection cannot provide the most faithful distances between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

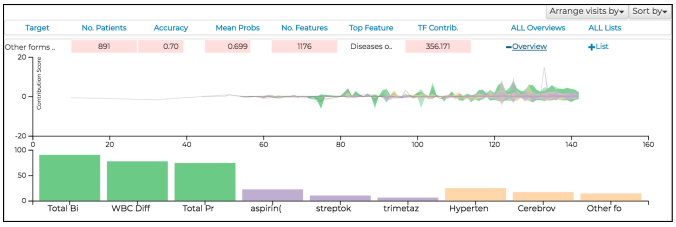

Fig. 4. *Patient Summary* shows a summary of selected patients. Table summarizes description of selected patients. In the middle, an area chart shows aggregated contribution scores of nine medical codes over time. It shows mean and standard deviation as an area. Users can also see the medical codes and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

points. Users may not be able to correctly express their regions of interests due to such information loss. Many prior techniques attempt to expose such artifacts of dimension reduction (e.g., CheckViz [51]) and to resolve the issues by providing some guidance [2, 31]. Alternatively, we can provide visual summaries of axes of nonlinear projection (e.g., AxiSketcher [46]). In order to provide more options for scatter plot axes, we allowed users to select and switch axes to other measures, such as prediction certainty, contribution scores of medical codes, and demographic information (e.g., age). The 2D projection view may also suffer the clutter issue, which can be minimized by showing only representative points based on clustering as shown in Clustervision [45].

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 predicted diagnosis risk as a dotted horizontal line in prediction circle chart. Users can also 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.

5.2 Patient Summary

Patient Summary, in contrast to Overview, shows a temporal summary in contribution progress chart (**T3**). There are three charts vertically shown from top to bottom in Patient Summary as Figure 4 shows. The first chart is a table that summarizes selected patients: 1) number of patients, 2) accuracy (number of correct prediction / number of patients), 3) mean predicted diagnosis risk, 4) number of medical codes, 5) name of top contributing medical code, and 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 toward the end of patient records than other types.

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 highlight a code in contribution progress chart by hovering over the corresponding bar. By clicking on 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 in Figure 4, 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 [47, 48]. Each box decorated with a diverging color scheme of the blue-to-white-to-red (negative-to-0-to-positive) scale represents 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 predicted diagnosis risk. 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. Contribution scores show how much each medical code or visit impacts upon prediction certainty. This contribution is the most essential unit of interpretability by showing the relationship between event sequences and predicted outcomes. In the patient list view, we aim to provide when and how much each visit impacted upon high or low prediction certainty for patients. The pattern shown in heatmap is used for users to select interesting patients who have unique patterns. Figure 1 (C) shows a list of patients with a high predicted diagnosis risk. Patients tend to have visits with high contribution scores towards more recent visits, but exceptions can be seen (**T5**). In Patient List, users can invoke Patient Details and Patient Editor of a selected patient.

5.4 Patient Details

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

Temporal code chart shows contribution scores of all medical codes for each patient. The view is similarly arranged horizontally per visit as in Patient List. In Patient Details, we mapped the horizontal space for temporal progression and the vertical space for contribution scores as well as diagnostic progression risks. In this way, users are able to observe correlation between contribution scores of medical codes and prediction risks. 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. Code bar chart shows the top nine contributing medical codes.

In Patient Details, users can understand the progression of predicted diagnosis risks and why such predictions are made (**T5**). When users hover over the x-axis, they can see the updated contribution scores of medical codes of preceding visits until the point of time.

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 1 (D)). 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 1 (D). User can sort medical codes of a visit either by contribution score (default) or by the code type. This layout enables users to easily select medical codes to make changes for interaction. Second, users can convert Patient Details into Patient Editor by simply choosing a context menu option. By doing so, users can maintain the context while editing the patient visits and medical codes if they were focusing on Patient Details. However, users will lose the original version if they directly edit on Patient Details. Since there is a tradeoff between the two approaches, we implemented both features and allowed users to choose one at their convenience.

As shown in Figure 1 (D), users can move the visit along the time axis to change the date. Users can also add new codes in to a visit, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

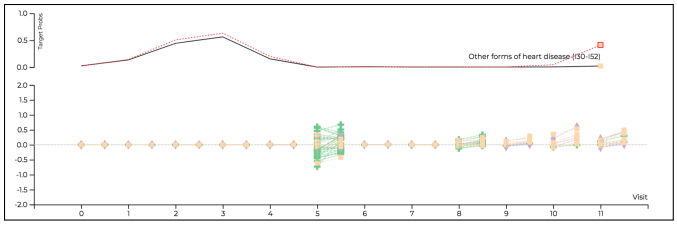

Fig. 5. The what-if analysis result shows increase in predicted diagnosis risks and contribution scores of related medical codes.

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 predicted diagnosis risk 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 1 (D)). The results are shown in Figure 5. The predicted risk significantly increased; in particular, predicted risks of final two increased as the red dotted line shows in Figure 5. 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 reports the methods and results of quantitative and qualitative experiments using our models trained on the HIRA-NPS dataset for predicting heart failure and cataract, respectively.

6.1 Experimental Setup

The primary purpose of RETAIN is to provide interpretatable 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. Both heart failure and cataract datasets showed a similar distribution of patients. For heart failure, the min, max, and average visit lengths were 5, 188, and 20.79. For cataract, these lengths were 5, 148, and 20.05.

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 with a visit in a 1,400-dimensional binary vector.

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 (63,030 patients) for heart failure and 10,692 batches (117,612 patients) for cataract. We split each dataset into train, validation and test sets with a ratio of 0.65/0.1/0.25.

Our models are implemented on Pytorch 0.3.1 [63]. We trained our model on our training set using Adam [43] 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 hyperparameters and avoid overfitting. We used an Ubuntu 14.04 server equipped with two Nvidia Titan X GPUs to train our models. According to our setting, our RetainEX model takes 3.19 hours to train using 40,964 patients in our use case for 10 epochs, with 1148 seconds per epoch. For testing, it takes 366 seconds to make predictions of diagnosis risk scores as well as to generate contribution scores of patients' visits and codes.

Table 1. Model performances measured for medical predictions tasks

	(a) heart failure			(b) cataract		
Models	AUC	AP	time (s)	AUC	AP	time (s)
GRU	0.906	0.694	997	0.953	0.834	2367
RETAIN	0.905	0.729	1114	0.959	0.835	2700
RetainEX w/o time	0.946	0.769	1143	**0.975**	0.870	2619
RetainEX	**0.954**	**0.818**	1148	**0.975**	**0.878**	2632

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 [26].

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 RetainEX, so we compared it to an equivalent version without time intervals being used (RetainEX w/o time). For the baseline models, we apply the same training and hyperparameter selection strategy as mentioned above.

Table 1 shows that RetainEX outperforms the baseline models in all cases. The effect of adding an additional embedding matrix can be seen in comparison to the original RETAIN model and RetainEX without time. Given the otherwise identical settings, the improvement in performance is due to having two embeddings, one 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. By using the temporal dimension, our model can more accurately learn how to discriminate between important and unimportant visits. Last of all, the rightmost columns display the average time taken to train the model per epoch, that is to observe every training data once. Table 1 shows that RetainEX outperforms baseline models at the expense of only a small increase in training time.

6.3 Qualitative analysis

We also qualitatively reviewed 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 patients, which we denote as $\mathbf{S}_{total}$ and $\mathbf{C}_{total}$. We normalize $\mathbf{S}_{total}$ in two different directions: (i) 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 B); (ii) 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 $\mathbf{S}_2$ (see Table 3 in Appendix B).

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 [38, 52]. Hypertensive disease was the most frequently diagnosed co-morbidity in patients with heart failure (Table 2 in Appendix B). Likewise, ischaemic heart disease was also a major disease in patients with heart failure [62], 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 [8]. It was presumed that cerebrovascular disease would be diagnosed in a high number of patients with heart failure

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

as hypertension is a common characteristic of both diseases [5, 72]. Bisoprolol, a medicine ingredient for which a relatively high $\mathbf{S}_1$ score was recorded, is frequently prescribed for heart disease [22], while aspirin and atorvastatin are commonly used to prevent it [20].

Carvedilol is a major prescription agent used to treat heart failure [21]. 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 [40]. Disorders of the thyroid gland are also known to cause heart failure [44]. These codes were found to have high scores in $\mathbf{S}_2$ (Table 3 in Appendix B). 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 the medical field. She is very interested in analyzing patients with heart failure (HF) and determining sequences of medical codes that are related to the onset of the disease.

Jane decided to conduct a predictive analysis using RetainVis with RetainEX trained by 40,964 patients (1:10 ratio between case and control) for the heart failure case study (see Section 6 for details). She pulled 3,724 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. [84]. The top three medications are bisoprolol hemifumarate, aspirin, 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 [7]. 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 [11]. 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) [78]. The group also showed high contribution scores of bilirubin, suspected as a predictive marker of pulmonary arterial hypertension [77]. 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 Details, 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 by sharing lessons for designing visual analytics applications using RNN-based models on a diagnostic risk prediction task.

8.1 Interpretability and Model Performance

Adding interpretability to a model while preserving its performance is a challenging task. The strength of RNNs comes from 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 interpretabiltiy and the performance of RNN models is crucial in designing a visual analytics tool. Target user tasks can be a guidance to solving this deadlock. Our tasks in Section 3.4 show an example.

One golden rule of data visualization is to maintain 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 that the first step to fulfilling this role is to have visualization results as simple as possible. He also revealed that it is important for the model and results to be intepretable and interactive, but also easily explainable by design. He also pointed out clinicians may not 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2018.2865027, IEEE Transactions on Visualization and Computer Graphics

making, instead of assisting it. This opinion was supported by another expert stating that visualization methods were partly unintuitive and difficult 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 discussion points 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 particular 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 different variations of our model 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 in Section 7. Meanwhile, a more simplified version highlighting only significant and anomalous events can be adopted as an assistant tool for clinicians. The presented events may provide new insights that might have otherwise been overlooked. Thus, designers and providers need to correctly identify target users' needs and maximize the desired effects out of the given settings.

8.2 Towards Interactivity

Another important objective of our work was to apply user interaction schemes for 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 also an important concept that we present, here we focus more on the retraining module. This strategy was effective in correcting the contribution scores that were assigned to certain visits or medical codes. We selected a case patient and a control patient, who were misclassified with the model trained initially (Type-II and Type-I errors). With the help of medical experts, we found which codes were over- and under-represented 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 mean diagnostic risk prediction scores of a test dataset increased from 0.812 to 0.814. 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 patients leads to improved performance shows an example of how users can teach the machine based on their domain knowledge. Such interactions can resolve inherent problems of machine learning-based models, where the performance does not improve without additional training data. In particular, RetainVis retrained the model using users' updates made to contribution scores of medical codes. In other words, the feature-level interpretation can be used as an interaction handle. In this manner, users do not have 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 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 in 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 relatively low, near 0.2. This reflects a common problem in applying machine learning to medical prediction tasks, where a high score cannot guarantee to prevent a serious mistake.

Visualizations can be carefully used to communicate the performance of the model in a transparent way. Domain experts often hesitate to accept what they see as facts since they are unable to tell various uncertainties propagated through the pipeline [69]. There might be artificial patterns created due to inconsistent EMR recording practice, which could then be amplified by incorrectly trained models. Even for examples that were proven to be accurate, visualizations should indicate its inherent uncertainties involved. Thus, future researchers can investigate the design of uncertainty visualizations, 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 given data.

8.4 Limitations and Future Work

Our tool has several limitations in terms of scalability due to its computationally expensive deep learning-based model and limited screen real estates. Since the scatter plot can suffer from overplotting given too many patients, our recommendation is not to go beyond 10K patients visualized in the scatter plot. The Lasso-selection in the scatter plot can mislead users as described in Section 5. Our line charts may not work when we visualize more than decades of records with more than thousands of visits. In such a case, we recommend to use a horizontal scroll for users to navigate through users' medical history. Our main interactions to view updated results based on changes the user made to input values, can be improved. We may compute and visualize all possible input value combinations that can lead to the improvement of outcomes. Though our case study demonstrates the usefulness of RetainVis, it has not been empirically tested whether and how various features of RetainVis can help users solve problems. Our future work aims to conduct a long-term user study, where health care professionals will provide feedback to improve the overall design and usability issues.

9 Conclusion

In this study, we developed a visual analytics system called RetainVis by incorporating RetainEX into electronic medical datasets. 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 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.

Acknowledgments

We thank Wonkyu Kim, who participated in discussion to improve the design of RetainVis, and our colleagues from IBM Research, Korea University, Georgia Institute of Technology, and other institutions, who provided constructive feedback. This research was partly supported by Korea Electric Power Corporation (Grant Number: R18XA05).

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] M. Aupetit, N. Heulot, and J.-D. Fekete. A multidimensional brush for scatterplot data analytics. In *Visual Analytics Science and Technology (VAST)*, pp. 221 – 222. IEEE, Oct. 2014.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*, 2015.

[4] 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.

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

[6] M. Cavallo and Ç. Demiralp. Exploring dimensionality reductions with forward and backward projections. *arXiv preprint arXiv:1707.04281*, 2017.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] Y. Ding, Y. Liu, H. Luan, and M. Sun. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[24] 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.

[25] 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.

[26] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. *Machine learning*, 31(1):1–38, 2004.

[27] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

[28] 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.

[29] 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.

[30] 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.

[31] N. Heulot, M. Aupetit, and J.-D. Fekete. ProxiLens: Interactive Exploration of High-Dimensional Data using Projections. In *VAMP: EuroVis Workshop on Visual Analytics using Multidimensional Projections*. The Eurographics Association, June 2013.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

[41] 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.

[42] 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.

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

International Conference on Learning Representations, 2015.

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] S. Lespinats and M. Aupetit. CheckViz: Sanity Check and Topological Clues for Linear and Non-Linear Mappings. *Computer Graphics Forum*, 30(1):113–125, 2010.

[52] 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.

[53] 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.

[54] Z. C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[55] 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.

[56] 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.

[57] 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.

[58] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015.

[59] 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.

[60] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(Nov), 2008.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

[68] 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.

[69] 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.

[70] 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.

[71] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

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

[80] W. Wang, S. Xu, and B. Xu. First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pp. 2243–2247, 2016. doi: 10.21437/Interspeech.2016-134

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

[82] 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.

[83] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., *Computer Vision – ECCV 2016*, pp. 451–466. Springer International Publishing, Cham, 2016.

[84] 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.

[85] 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.