Patient Subtyping via Time-Aware LSTM Networks

Inci M. Baytas
Computer Science and Engineering
Michigan State University
428 S Shaw Ln.
East Lansing, MI 48824
baytasin@msu.edu

Cao Xiao
Center for Computational Health
IBM T. J. Watson Research Center
1101 Kitchawan Rd
Yorktown Heights, NY 10598
cxiao@us.ibm.com

Xi Zhang
Healthcare Policy and Research
Weill Cornell Medical School
Cornell University
New York, NY 10065
sheryl.zhangxi@gmail.com

Fei Wang
Healthcare Policy and Research
Weill Cornell Medical School
Cornell University
New York, NY 10065
few2001@med.cornell.edu

Anil K. Jain
Computer Science and Engineering
Michigan State University
428 S Shaw Ln.
East Lansing, MI 48824
jain@cse.msu.edu

Jiayu Zhou
Computer Science and Engineering
Michigan State University
428 S Shaw Ln.
East Lansing, MI 48824
jiayuz@msu.edu

ABSTRACT

In the study of various diseases, heterogeneity among patients usually leads to different progression patterns and may require different types of therapeutic intervention. Therefore, it is important to study patient subtyping, which is grouping of patients into disease characterizing subtypes. Subtyping from complex patient data is challenging because of the information heterogeneity and temporal dynamics. Long-Short Term Memory (LSTM) has been successfully used in many domains for processing sequential data, and recently applied for analyzing longitudinal patient records. The LSTM units are designed to handle data with constant elapsed times between consecutive elements of a sequence. Given that time lapse between successive elements in patient records can vary from days to months, the design of traditional LSTM may lead to suboptimal performance. In this paper, we propose a novel LSTM unit called Time-Aware LSTM (T-LSTM) to handle irregular time intervals in longitudinal patient records. We learn a subspace decomposition of the cell memory which enables time decay to discount the memory content according to the elapsed time. We propose a patient subtyping model that leverages the proposed T-LSTM in an auto-encoder to learn a powerful single representation for sequential records of patients, which are then used to cluster patients into clinical subtypes. Experiments on synthetic and real world datasets show that the proposed T-LSTM architecture captures the underlying structures in the sequences with time irregularities.

CCS CONCEPTS

•Applied computing →Health informatics; •Mathematics of computing →*Time series analysis;* **•Computing methodologies →***Cluster analysis;*

KDD '17, Halifax, NS, Canada

DOI: 10.1145/3097983.3097997

KEYWORDS

Patient subtyping, Recurrent Neural Network, Long-Short Term Memory

ACM Reference format:
Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. 2017. Patient Subtyping via Time-Aware LSTM Networks. In *Proceedings of KDD '17, Halifax, NS, Canada, August 13-17, 2017,* 10 pages.
DOI: 10.1145/3097983.3097997

1 INTRODUCTION

Clinical decision making often relies on medical history of patients. Physicians typically use available information from past patient visits such as lab tests, procedures, medications, and diagnoses to determine the right treatment. Furthermore, researchers use medical history and patient demographics to discover interesting patterns in patient cohorts, to study prognosis of different types of diseases, and to understand effects of drugs. In a nut shell, large-scale, systematic and longitudinal patient datasets play a key role in the healthcare domain. Examples such as Electronic Health Records (EHRs), whose adoption rate increased by 5% between 2014 and 2015 [1] in the healthcare systems in the United States, facilitate a systematic collection of temporal digital health information from variety of sources.

With the rapid development of computing technologies in healthcare, longitudinal patient data are now beginning to be readily available. However, it is challenging to analyze large-scale heterogeneous patient records to infer high level information embedded in patient cohorts. This challenge motivates the development of computational methods for biomedical informatics research [5, 17, 24, 28, 31]. These methods are required to answer different questions related to disease progression modeling and risk prediction [9, 10, 14, 22, 30, 32].

Patient Subtyping, which seeks patient groups with similar disease progression pathways, is crucial to address the heterogeneity in the patients which ultimately leads to precision medicine where patients are provided with treatments tailored to their unique health status. Patient subtyping facilitates the investigation of a particular type of complicated disease condition [5]. From the data mining

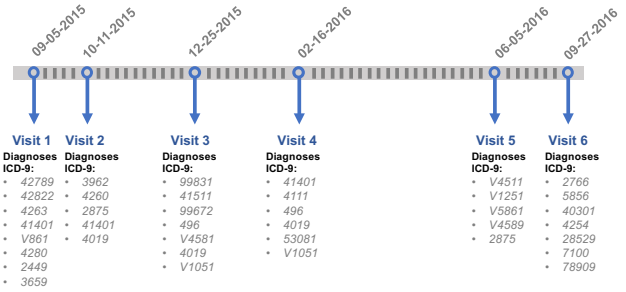


Figure 1: An example segment of longitudinal patient records. The patient had 6 office visits between Sept 5, 2015 and Sept 27, 2016. In each visit, diagnosis of the patient was given by a set of ICD-9 codes. Time spans between two successive visits can vary, and may be months apart. Such time irregularity results in a significant challenge in patient subtyping.

perspective, patient subtyping is posed as an unsupervised learning task of grouping patients according to their historical records. Since these records are longitudinal, it is important to capture the relationships and the dependencies between the elements of the record sequence in order to learn more effective and robust representations, which can then be used in the clustering stage to obtain the patient groups.

One powerful approach which can capture underlying structure in sequential data is Recurrent Neural Networks (RNNs), which have been applied to many areas such as speech recognition [16], text classification [21], video processing [13, 26], and natural language processing [27]. In principle, time dependencies between the elements can be successfully captured by RNNs, however traditional RNNs suffer from vanishing and exploding gradient problems. To handle these limitations, different variants of RNN have been proposed. Long-Short Term Memory (LSTM) [18] is one such popular variant which can handle long term event dependencies by utilizing a gated architecture. LSTM has recently been applied in health informatics [4, 6] with promising results.

One limitation of the standard LSTM networks is that it cannot deal with irregular time intervals. But, the time irregularity is common in many healthcare applications. To illustrate this, one can consider patient records, where the time interval between consecutive visits or admissions varies, from days to months and sometimes a year. We illustrate this in Figure 1 using a sample medical record segment for one patient. Notice that the time difference between records varies from one month to a few months. Such varying time gaps could be indicative of certain impending disease conditions. For instance, frequent admissions might indicate a severe health problem and the records of those visits provide a source to study progression of the condition. On the other hand, if there are months between the two successive records, dependency on the previous memory should not play an active role to predict the current outcome.

To address the aforementioned challenges in patient subtyping, we propose an integrated approach to identify patient subtypes using a novel Time-Aware LSTM (T-LSTM), which is a modified LSTM architecture that takes the elapsed time into consideration between the consecutive elements of a sequence to adjust the memory content of the unit. T-LSTM is designed to incorporate the time irregularities in the memory unit to improve the performance of the standard LSTM. The main contributions of this paper are summarized bellow:

- A novel LSTM architecture (T-LSTM) is proposed to handle time irregularities in sequences. T-LSTM has *forget, input, output* gates of the standard LSTM, but the memory cell is adjusted in a way that longer the elapsed time, smaller the effect of the previous memory to the current output. For this purpose, elapsed time is transformed into a weight using a time decay function. The proposed T-LSTM learns a neural network that performs a decomposition of the cell memory into short and long-term memories. The short-term memory is discounted by the decaying weight before combining it with the long-term counterpart. This subspace decomposition approach does not change the effect of the current input to the current output, but alters the effect of the previous memory on the current output.
- An unsupervised patient subtyping approach is proposed based on clustering the patient population by utilizing the proposed T-LSTM unit. T-LSTM is used to learn a single representation from the temporal patient data in an auto-encoder setting. The proposed T-LSTM auto-encoder maps sequential records of patients to a powerful representation capturing the dependencies between the elements in the presence of time irregularities. The representations learned by the T-LSTM auto-encoder are used to cluster the patients by using the k-means algorithm.

Supervised and unsupervised experiments on both synthetic and real world datasets show that the proposed T-LSTM architecture performs better than standard LSTM unit to learn discriminative representations from sequences with irregular elapsed times.

The rest of the paper is organized as follows: related literature survey is summarized in Section 2, technical details of the proposed approach are explained in Section 3, experimental results are presented in Section 4, and the study is concluded in Section 5.

2 RELATED WORK

Computational Subtyping with Deep Networks. A similar idea as presented in this study was proposed in [25], but for supervised problem settings. Pham *et al.* introduced an end-to-end deep network to read EHRs, saves patient history, infers the current state and predicts the future. Their proposed approach, called "DeepCare", used LSTM for multiple admissions of a patient, and also addressed the time irregularities between the consecutive admissions. A single vector representation was learned for each admission and was used as the input to the LSTM network. Forget gate of standard LSTM unit was modified to account for the time irregularity of the admissions. In our T-LSTM approach, however the memory cell is adjusted by the elapsed time. The main aim of [25] was answering the question "What happens next?". Therefore, the authors of [25] were dealing with a supervised problem setting whereas we deal with an unsupervised problem setting.

There are several studies in the literature using RNNs for supervised tasks. For instance, in [14], authors focused on patients suffering from kidney failure. The goal of their approach was to predict whether a patient will die, the transplant will be rejected,

or transplant will be lost. For each visit of a patient, the authors tried to answer the following question: which one of the three conditions will occur both within 6 months and 12 months after the visit? RNN was used to predict these aforementioned endpoints. In [22], LSTM was used to recognize patterns in multivariate time series of clinical measurements. Subtyping clinical time series was posed as a multi-label classification problem. Authors stated that diagnostic labels without timestamps were used, but timestamped diagnoses were obtained. LSTM with a fully connected output layer was used for the multi-label classification problem.

In [10] authors aimed to make predictions in a similar way as doctors do. RNN was used for this purpose and it was fed by the patient's past visits in a reverse time order. The way RNN was utilized in [10] is different than its general usage. There were two RNNs, one for visit-level and the other for variable-level attention mechanisms. Thus, the method proposed in [10] could predict the diagnosis by first looking at the more recent visits of the patient, and then determining which visit and which event it should pay attention.

Another computational subtyping study [9] learned a vector representation for patient status at each time stamp and predicted the diagnosis and the time duration until the next visit by using this representation. Authors proposed a different approach to incorporate the elapsed time in their work. A softmax layer was used to predict the diagnosis and a ReLU unit was placed at the top of the GRU to predict the time duration until the next visit. Therefore, the elapsed time was not used to modify the GRU network architecture but it was concatenated to the input to be able to predict the next visit time. On the other hand, authors in [4] aimed to learn patient similarities directly from temporal EHR data for personalized predictions of Parkinson's disease. GRU unit was used to encode the similarities between the sequences of two patients and dynamic time warping was used to measure the similarities between temporal sequences.

A different approach to computational subtyping was introduced in [11]. Their method, called Med2Vec, was proposed to learn a representation for both medical codes and patient visits from large scale EHRs. Their learned representations were interpretable, therefore Med2Vec did not only learn representations to improve the performance of algorithms using EHRs but also to provide interpretability for physicians. While the authors did not use RNN, they used a multi-layer perceptron to generate a visit representation for each visit vector.

Auto-Encoder Networks. The purpose of our study is patient subtyping which is an instance of unsupervised learning or clustering, therefore we need to learn powerful representations of the patient sequences that can capture the dependencies and the structures within the sequence. One of the ways to learn representations by deep networks is to use auto-encoders. Encoder network learns a single representation of the input sequence and then the decoder network reconstructs the input sequence from the representation learned by the encoder at the end of the input sequence. In each iteration, reconstruction loss is minimized so that the learned representation is effective to summarize the input sequence. In [26] LSTM auto-encoders were used to learn representations for video sequences. Authors tested the performance of the learned representation on supervised problems and showed that the learned representation is able to increase the classification accuracy.

Auto-encoders are also used to generate a different sequence by using the representation learned in the encoder part. For instance, in [7], one RNN encodes a sequence of symbols into a vector representation, and then the decoder RNN map the single representation into another sequence. Authors of [7] showed that their proposed approach can interpret the input sequence semantically and can learn its meaningful representation syntactically.

3 METHODOLOGY

3.1 Time-Aware Long Short Term Memory

3.1.1 Long Short-Term Memory (LSTM). Recurrent neural network (RNN) is a deep network architecture where the connections between hidden units form a directed cycle. This feedback loop enables the network to keep the previous information of hidden states as an internal memory. Therefore, RNNs are preferred for problems where the system needs to store and update the context information [3]. Approaches such as Hidden Markov Models (HMM) have also been used for similar purposes, however there are distinctive properties of RNNs that differentiates them from conventional methods such as HMM. For example, RNNs do not make the assumption of Markov property and they can process variable length sequences. Furthermore, in principle, information of past inputs can be kept in the memory without any limitation on the time in the past. However, optimization for long-term dependencies is not always possible in practice because of vanishing and exploding gradient problems where the value of gradient becomes too small and too large, respectively. To be able to incorporate the long-term dependencies without violating the optimization process, variants of RNNs have been proposed. One of the popular variants is Long Short-Term Memory (LSTM) which is capable of handling long-term dependencies with a gated structure [18].

A standard LSTM unit comprises of forget, input, output gates, and a memory cell, but the architecture has the implicit assumption of uniformly distributed elapsed time between the elements of a sequence. Therefore, the time irregularity, which can be present in a longitudinal data, is not integrated into the LSTM architecture. For instance, the distribution of the events in a temporal patient record is highly non-uniform such that the time gap between records can vary from days to years. Given that the time passed between two consecutive hospital visits is one of the sources of decision making in the healthcare domain, an LSTM architecture which takes irregular elapsed times into account is required for temporal data. For this purpose, we propose a novel LSTM architecture, called Time-Aware LSTM (T-LSTM), where the time lapse between successive records is included in the network architecture. Details of T-LSTM are presented in the next section.

3.1.2 Time-Aware LSTM (T-LSTM). Regularity of the duration between consecutive elements of a sequence is a property that does not always hold. One reason of the variable elapsed time is the nature of the EHR datasets, where the frequency and the number of patient records are quite unstructured. Another reason is missing information in the longitudinal data. In case of the missing data, elapsed time irregularity impacts predicting the trajectory of the

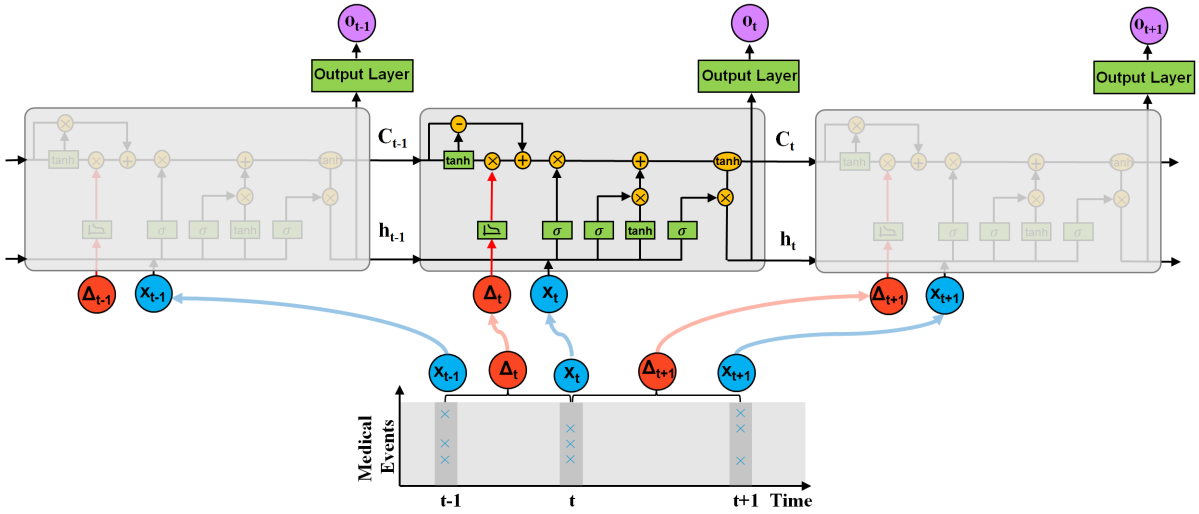


Figure 2: Illustration of the proposed time-aware long-short term memory (T-LSTM) unit, and its application on analyzing healthcare records. Green boxes indicate networks and yellow circles denote point-wise operators. T-LSTM takes two inputs, input record and the elapsed time at the current time step. The time lapse between the records at time $t-1$, t and $t+1$ can vary from days to years in healthcare domain. T-LSTM decomposes the previous memory into long and short term components and utilizes the elapsed time (Δ_t) to discount the short term effects.

temporal changes. Therefore, an architecture that can overcome this irregularity is necessary to increase the prediction performance. For EHR data, varying elapsed times can be treated as a part of the information contained in the medical history of a patient, hence it should be utilized while processing the records.

T-LSTM is proposed to incorporate the elapsed time information into the standard LSTM architecture to be able to capture the temporal dynamics of sequential data with time irregularities. The proposed T-LSTM architecture is given in Figure 2 where the input sequence is represented by the temporal patient data. Elapsed time between two immediate records of a patient can be quite irregular. For instance, time between two consecutive admissions/hospital visits can be weeks, months and years. If there are years between two successive records, then the dependency on the previous record is not significant enough to affect the current output, therefore the contribution of the previous memory to the current state should be discounted. The major component of the T-LSTM architecture is the subspace decomposition applied on the memory of the previous time step. While the amount of information contained in the memory of the previous time step is being adjusted, we do not want to lose the global profile of the patient. In other words, long-term effects should not be discarded entirely, but the short-term memory should be adjusted proportional to the amount of time span between the records at time step t and $t-1$. If the gap between time t and $t-1$ is huge, it means there is no new information recorded for the patient for a long time. Therefore, the dependence on the short-term memory should not play a significant role in the prediction of the current output.

T-LSTM applies the memory discount by employing the elapsed time between successive elements to weight the short-term memory content. To achieve this, we propose to use a non-increasing function of the elapsed time which transforms the time lapse into an appropriate weight. Mathematical expressions of the subspace decomposition procedure are provided in Equation Current hidden state. First, short-term memory component (C^S_{t-1}) is obtained by a network. Note that this decomposition is data-driven and the parameters of the decomposition network are learned simultaneously with the rest of network parameters by back-propagation. There is no specific requirement for the activation function type of the decomposition network. We tried several functions but did not observe a drastic difference in the prediction performance of the T-LSTM unit, however tanh activation function performed slightly better. After the short-term memory is obtained, it is adjusted by the elapsed time weight to obtain the discounted short-term memory ($\hat{C}^S_{t-1}$). Finally, to compose the adjusted previous memory back (C^*_{t-1}), the complement subspace of the long-term memory ($C^T_{t-1} = C_{t-1} - C^S_{t-1}$) is combined with the discounted short-term memory. Subspace decomposition stage of the T-LSTM is followed by the standard gated architecture of the LSTM. Detailed mathematical expressions of the proposed T-LSTM architecture are given below:

$$C^S_{t-1} = \tanh\left(W_d C_{t-1} + b_d\right) \quad \text{(Short-term memory)}$$
$$\hat{C}^S_{t-1} = C^S_{t-1} * g\left(\Delta_t\right) \quad \text{(Discounted short-term memory)}$$
$$C^T_{t-1} = C_{t-1} - C^S_{t-1} \quad \text{(Long-term memory)}$$
$$C^*_{t-1} = C^T_{t-1} + \hat{C}^S_{t-1} \quad \text{(Adjusted previous memory)}$$
$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad \text{(Forget gate)}$$
$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \quad \text{(Input gate)}$$
$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \quad \text{(Output gate)}$$
$$\tilde{C} = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \quad \text{(Canditate memory)}$$

$$C_t = f_t * C^*_{t-1} + i_t * \tilde{C} \quad \text{(Current memory)}$$
$$h_t = o * \tanh(C_t), \quad \text{(Current hidden state)}$$

where x_t represents the current input, h_{t-1} and h_t are previous and current hidden states, and C_{t-1} and C_t are previous and current cell memories. $\{W_f, U_f, b_f\}$, $\{W_i, U_i, b_i\}$, $\{W_o, U_o, b_o\}$, and $\{W_c, U_c, b_c\}$ are the network parameters of the forget, input, output gates and the candidate memory, respectively. $\{W_d, b_d\}$ are the network parameters of the subspace decomposition. Dimensionalities of the parameters are determined by the input, output and the chosen hidden state dimensionalities. Δ_t is the elapsed time between x_{t-1} and x_t and $g(\cdot)$ is a heuristic decaying function such that the larger the value of Δ_t, less the effect of the short-term memory. Different types of monotonically non-increasing functions can be chosen for $g(\cdot)$ according to the measurement type of the time durations for a specific application domain. If we are dealing with time series data such as videos, the elapsed time is generally measured in seconds. On the other hand, if the elapsed time varies from days to years as in the healthcare domain, we need to convert the time lapse of successive elements to one type, such as days. In this case, the elapsed time might have large numerical values when there are years between two consecutive records. As a guideline, $g(\Delta_t) = 1/\Delta_t$ can be chosen for datasets with small amount of elapsed time and $g(\Delta_t) = 1/\log(e + \Delta_t)$ [25] is preferred for datasets with large elapsed times.

In the literature, studies proposing different ways to incorporate the elapsed time into the learning process can be encountered. For instance, elapsed time was used to modify the forget gate in [25]. In T-LSTM, one of the reasons behind adjusting the memory cell instead of the forget gate is to avoid any alteration of the current input's effect to the current output. The current input runs through the forget gate and the information coming from the input plays a role to decide how much memory we should keep from the previous cell. As can be seen in the expressions of *Current memory* and *Current hidden state* in Equation Current hidden state, modifying the forget gate directly might eliminate the effect of the input to the current hidden state. Another important point is that, the subspace decomposition enables us to selectively modify the short-term effects without losing the relevant information in the long-term memory. Section 4 shows that the performance of T-LSTM is improved by modifying the forget gate, which is named as Modified Forget Gate LSTM (MF-LSTM) in this paper. Two approaches are adopted from [25] for comparison. First approach, denoted by MF1-LSTM, multiplies the output of the forget gate by $g(\Delta_t)$ such as $f_t = g(\Delta_t) * f_t$. whereas MF2-LSTM utilizes a parametric time weight such as $f_t = \sigma\left(W_f x_t + U_f h_{t-1} + Q_f q_{\Delta_t} + b_f\right)$ where $q_{\Delta_t} = \left(\frac{\Delta_t}{60}, \left(\frac{\Delta_t}{180}\right)^2, \left(\frac{\Delta_t}{360}\right)^3\right)$ when Δ_t is measured in days similar to [25].

Another idea to handle the time irregularity could be imputing the data by sampling new records between two consecutive time steps to have regular time gaps and then applying LSTM on the augmented data. However, when the elapsed time is measured in days, so many new records have to be sampled for the time steps which have years in between. Secondly, the imputation approach might have a serious impact on the performance. A patient record contains detailed information and it is hard to guarantee that the imputed records reflect the reality. Therefore, a change in the architecture of the regular LSTM to handle time irregularities is suggested.

3.2 Patient Subtyping with T-LSTM Auto-Encoder

In this paper, patient subtyping is posed as an unsupervised clustering problem since we do not have any prior information about the groups inside the patient cohort. An efficient representation summarizing the structure of the temporal records of patients is required to be able to cluster temporal and complex EHR data. Auto-encoders provide an unsupervised way to directly learn a mapping from the original data [2]. LSTM auto-encoders have been used to encode sequences such as sentences [33] in the literature. Therefore, we propose to use T-LSTM auto-encoder to learn an effective single representation of the sequential records of a patient. T-LSTM auto-encoder has T-LSTM encoder and T-LSTM decoder units with different parameters which are jointly learned to minimize the reconstruction error. The proposed auto-encoder can capture the long and the short term dependencies by incorporating the elapsed time into the system and learn a single representation which can be used to reconstruct the input sequence. Therefore, the mapping learned by the T-LSTM auto-encoder maintains the temporal dynamics of the original sequence with variable time lapse.

In Figure 3, a single layer T-LSTM auto-encoder mechanism is given for a small sequence with three elements $[X_1, X_2, X_3]$. The hidden state and the cell memory of the T-LSTM encoder at the end of the input sequence are used as the initial hidden state and the memory content of the T-LSTM decoder. First input element and the elapsed time of the decoder are set to zero and its first output is the reconstruction ($\hat{X_3}$) of the last element of the original sequence (X_3). When the reconstruction error E_r given in Equation 1 is minimized, T-LSTM encoder is applied to the original sequence to obtain the learned representation, which is the hidden state of the encoder at the end of the sequence.

$$E_r = \sum_{i=1}^{L} \left\| X_i - \hat{X}_i \right\|_2^2, \quad (1)$$

where L is the length of the sequence, X_i is the ith element of the input sequence and $\hat{X}_i$ is the ith element of the reconstructed sequence. The hidden state at the end of the sequence carries concise information about the input such that the original sequence can be reconstructed from it. In other words, representation learned by the encoder is a summary of the input sequence [8]. The number of layers of the auto-encoder can be increased when the input dimension is high. A single layer auto-encoder requires more number of iterations to minimize the reconstruction error when the learned representation has a lower dimensionality compared to the original input. Furthermore, learning a mapping to low dimensional space requires more complexity in order to capture more details of the high dimensional input sequence. In our experiments, a two layer T-LSTM auto-encoder, where the output of the first layer is the input of the second layer, is used because of the aforementioned reasons.

Given a single representation of each patient, patients are grouped by the k-means clustering algorithm. Since we do not make any assumption about the structure of the clusters, the simplest clustering

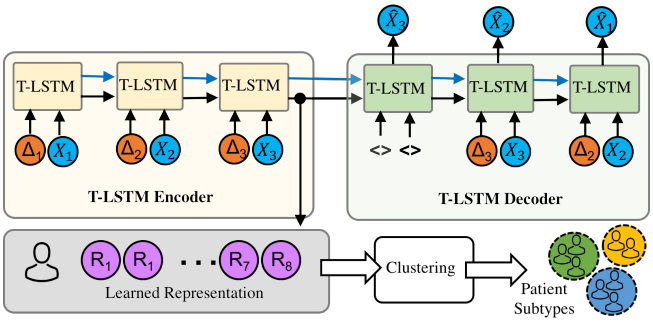


Figure 3: Clustering patients with a single-layer T-LSTM Auto-Encoder. Blue arrows denote the cell memory and the black arrows denote the hidden states. After the representations (R_i, $i = 1, 2, \cdots, 8$) are learned for the population, we can cluster the patients and obtain subtypes for each group as the prominent common medical features of the group. Number of layers should be increased in case of dimensionality reduction to be able to capture more complex structure with fewer iterations compared to single layer.

algorithm, k-means, is preferred. In Figure 3, a small illustration of clustering the patient cohort for 8 patients is shown. In this figure, learned representations are denoted by R. If R has the capability to represent the distinctive structure of patient sequence, then clustering algorithm can group patients with similar features (diagnoses, lab results, medications, conditions, and so on) together. Thus, each patient group has a subtype, which is a collection of common medical features present in the cluster. Given a new patient, learned T-LSTM encoder is used to find the representation of the patient and the subtype of the cluster which gives the minimum distance between the cluster centroid and the new patient's representation is assigned to the new patient. As a result, T-LSTM auto-encoder learns powerful single representation of temporal patient data that can be easily used to obtain the subtypes in the patient population.

4 EXPERIMENTS

In this section, experimental results on synthetic and real world datasets are reported. For synthetic data, two sets of experiments were conducted such as a classification task on a publicly available synthetic EHR dataset and a clustering task with auto-encoder setting on a randomly generated synthetic data. Comparisons between T-LSTM, MF1-LSTM, MF2-LSTM [25], LSTM, and logistic regression are made. The application of T-LSTM auto-encoder on patient subtyping is presented on a real world dataset (PPMI) and subtyping results are discussed. T-LSTM [1] was implemented in Tensorflow and mini-batch stochastic Adam optimizer was used during experiments. All the weights were learned simultaneously and in data-driven manner. Same network settings and parameters were used for all the deep methods for comparison. Therefore, fixed number of epochs were chosen during the experiments instead of using a stopping criteria. Since there are variable size sequences in longitudinal patient data, batches with same sequence sizes were generated instead of padding the original sequences with zero to make every sequence same length. In this study, we did not use the publicly available large scale ICU dataset, MIMIC [19]. MIMIC is an ICU data, therefore sequence length for the majority of patients is very small such as one or two admissions. Even though MIMIC is an important public source for healthcare research, it is not suitable for our purpose such that very short sequences do not enable us to analyze long and short term dependencies and the effect of the elapsed time irregularities.

[1] Available at https://github.com/illidanlab/T-LSTM

Table 1: Supervised synthetic EHR experimental results, average AUC of testing on 10 different splits. Training and testing ratio was chosen as 70% and 30%, respectively.

Methods	Avg. Test AUC	Stdev.
T-LSTM	**0.91**	0.01
MF1-LSTM	0.87	0.02
MF2-LSTM	0.82	0.09
LSTM	0.85	0.02
LR	0.56	0.01

4.1 Synthetic Dataset

4.1.1 Supervised Experiment. In this section, we report experimental results for a supervised task on an artificially generated EHR data which can be found in [2]. The aforementioned data has electronic records of up to 100,000 patients with lab results, diagnoses, and start and end dates of the admissions. Each patient has a unique patient ID similar to real world EHR data. We refer to the reference study [20] for further details of the data generation process. Although the dataset is artificially generated, it contains similar characteristics as a real EHR data. In this experiment, target diagnoses was Diabetes Mellitus and the task was a binary classification problem. Input of the network was the sequence of admissions and the output was the predicted label as one-hot vector. Therefore, regular recurrent network setting was utilized for this task instead of auto-encoder. Feature of one admission was a multi-hot vector containing the diagnoses given in the corresponding admission and the vocabulary size was 529. For this purpose, 6,730 patients were sampled with an average of 4 admissions. For this task, a single layer T-LSTM, MF1-LSTM and MF2-LSTM networks were tested to compare the performance based on area under ROC curve (AUC) metric for 50 epochs. In this experiment, number of hidden and softmax layer neurons were chosen as 1028 and 512, respectively. In addition, performance of the traditional logistic regression (LR) classifier was also analyzed. In logistic regression experiments, admissions were aggregated for each patient without incorporating the elapsed time. We also tried to incorporate the elapsed time as a weight by using the same non-increasing function used in T-LSTM during the aggregation of admissions. However, this approach did not improve the performance in our case. The results are summarized in Table 1.

As it can be seen from the Table 1, T-LSTM has a better performance than the baseline approaches. The way to represent the sequential data could be improved further for logistic regression, but aggregation of the admissions for each patient did not perform well for this task. Supervised experiments show that LSTM networks can enable us to leverage the time aspect of the EHR data

[2] http://www.emrbots.org/

Table 2: Average Rand index of k-means over 10 runs. T-LSTM auto-encoder outperforms LSTM and MF1-LSTM auto-encoders. This result indicates that the time irregularities should be considered to capture the temporal dynamics of a sequence.

Method	Mean RI	Std
T-LSTM	0.96	0.05
MF1-LSTM	0.85	0.13
LSTM	0.90	0.09

better. In addition, modifying the cell memory yields a better classification performance. According to our observation, MF1-LSTM and MF2-LSTM have better and sometimes similar results as the traditional LSTM for the tasks in our experiments.

4.1.2 Unsupervised Experiment. In this experiment, we investigate the expressive power of the representation learned from the T-LSTM auto-encoder. For this purpose, a synthetic data was randomly generated and the clustering results were evaluated. Since we know the ground truth of the synthetic data, we computed the Rand index (RI), given in Equation 2 [23], of the clustering to observe the discriminative power of the learned representations. A large value of Rand index indicates that the learned representations are clustered close to the ground truth.

$$\text{RI} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{FN} + \text{TN}), \tag{2}$$

where TP, TN, FP, FN are true positive, true negative, false positive and false negative, respectively. Note that $0 \leq \text{RI} \leq 1$.

The results on a synthetic dataset containing 4 clusters generated from a mixture of normal distributions with four different means and the same covariance are reported. A data point in the synthetic dataset is a sequence of vectors and the values of the sequences are increasing with time. Some of the elements in the sequences are discarded randomly to introduce unstructured elapsed time and obtain variable sequence lengths of sizes 4, 6, 18, 22, and 30. Dimensionality of the vectors was 5 and the dimension was reduced to 2 by the T-LSTM auto-encoder to be able to plot the representations in a 2-D space. Thus, the input and the output of the T-LSTM auto-encoder were 5 dimensional input and the reconstructed sequences, respectively. The hidden state dimension of the second layer T-LSTM encoder was chosen as 2, therefore the learned representations were 2-dimensional single vectors. The learned representations were clustered by k-means, where k was set to 4. Representation learning was repeated 10 times with different initializations of k-means and the average Rand index of clustering is reported for T-LSTM, LSTM and MF1-LSTM auto-encoders in Table 2. The non-increasing heuristic function was chosen as $g(\Delta_t) = 1/\log(e + \Delta_t)$. For this experiment, we compared the performances of T-LSTM, MF1-LSTM and LSTM excluding MF2-LSTM. Since the time gap of the data used in this experiment does not relate to an actual time measurement such as days, MF2-LSTM was excluded.

Table 2 shows that the T-LSTM outperforms the baselines and T-LSTM auto-encoder can learn the underlying structure of the input sequence with varying elapsed times such that the representations obtained by T-LSTM encoder could be clustered. In this example, performance of MF1-LSTM was obtained better than LSTM on average. A visual example of one of the trials is also shown in Figure 4 where the 2-dimensional representations obtained by the three approaches were plotted.

In Figure 4 different colors denote ground truth assignments of different clusters. Representations learned by T-LSTM yields more compact groups in the 2-D space leading to a more accurate clustering result compared to the standard LSTM and MF1-LSTM. As it can be observed from the Figures 4c and 4b, directly multiplying the forget gate with the time coefficient does not always enables a modification which leverages the time irregularity in our experiments. Even though MF1-LSTM produced a higher Rand index, there are examples, such as Figure 4, where LSTM actually learns a better representation than MF1-LSTM. The change in the objective values of T-LSTM, MF1-LSTM and LSTM with respect to the number of epochs are also compared in Figure 5 for the trial illustrated in Figure 4. It is observed that the modifications related to the time irregularity does not affect the convergence of the original LSTM network in a negative way.

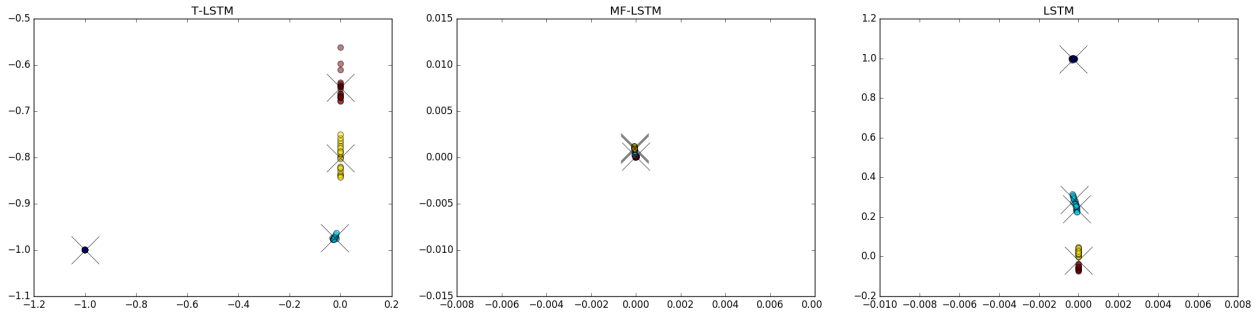
4.2 Parkinson's Progression Markers Initiative (PPMI) Data

In this section, we present experimental results for a real world dataset. Parkinson's Progression Markers Initiative (PPMI) is an observational clinical and longitudinal study comprising of evaluations of people with Parkinson's disease (PD), those people with high risk, and those who are healthy [12]. PPMI aims to identify biomarkers of the progression of Parkinson's disease. PPMI data is a publicly available dataset which contains clinical and behavioral assessments, imaging data, and biospecimens, therefore PPMI is a unique archive of PD [12]. As with many EHRs, PPMI is a longitudinal dataset with unstructured elapsed time. Therefore, T-LSTM is a suitable approach for prediction and clustering of PPMI dataset.

In our experiments, we used the pre-processed PPMI data of 654 patients given in [4]. Che *et al.* [4] collected patients with Idiopathic PD or non PD, imputed missing values, used one-hot feature form for categorical values, and encoded data abnormalities as 1 and 0. As a result, dataset we used has 15,636 records of 654 patients with an average of 25 sequences (minimum sequence length is 3). Authors of [4] also categorized data as features and targets, where the features are related to patient characteristics and the targets correspond to the progression of PD. A total of 319 features consist of motor symptoms/complications, cognitive functioning, autonomic symptoms, psychotic symptoms, sleep problems, depressive symptoms, and hospital anxiety and depression scale. A total of 82 targets are related to motor sign, motor symptom, cognition, and other non-motor factors [4]. Summary of the PPMI data used in this paper can be found in Table 3.

As it can be seen in Table 3, the elapsed time is measured as months. From 1 month to nearly 2 years gap between successive records of patients is encountered in the dataset. Several experiments were conducted on PPMI data to show the performance of the proposed subtyping approach.

4.2.1 Target Sequence Prediction. In this experiment, T-LSTM is used to predict the target sequence of each patient. For this purpose, we divided the data into different train (70%)-test (30%) splits and report the mean square error (MSE) between the original target sequence and the predicted target sequence. Average MSEs of 10



(a) T-LSTM Auto-Encoder **(b) MF1-LSTM Auto-Encoder** **(c) LSTM Auto-Encoder**

Figure 4: Illustration of the clustering results. Different colors denote ground truth assignments of different clusters. T-LSTM auto-encoder learns a mapping for the sequences such that 4 separate groups of points can be represented in the 2-D space.

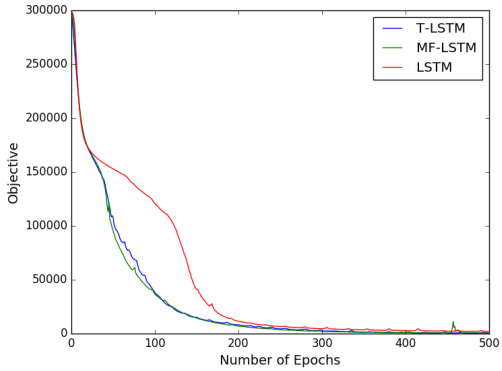


Figure 5: Change in the objective values of T-LSTM, MF1-LSTM and LSTM with respect to 500 epochs. It is observed that the modifications related to the time irregularity does not deteriorate the convergence of the original LSTM network.

Table 3: Details of PPMI data used in this study. Elapsed time encountered in the data is measured in months and it varies between 1 month to nearly 2 years. Here, the elapsed time interval is not the time interval of PPMI data recording, but elapsed times seen in records of individual patients.

Number of Patients	654
Elapsed Time Interval	[1, 26]
Average Sequence Length	25
Feature Dimensionality	319
Target Dimensionality	82

different train-test splits for T-LSTM, LSTM, MF1-LSTM and MF2-LSTM are given in Table 4. Same step size and the number of epochs were used for all the three methods. The non-increasing heuristic function of the elapsed time was chosen as $g(\Delta_t) = 1/\log(e + \Delta_t)$ for PPMI data.

We also investigated target features on which T-LSTM performed the best. The commonly encountered target features where the T-LSTM provided lower MSE than LSTM, MF1-LSTM and MF2-LSTM

Table 4: Average mean square error (MSE) for 10 different train-test splits for T-LSTM, LSTM, MF1-LSTM, and MF2-LSTM. T-LSTM yielded a better result than the standard LSTM in the presence of the unstructured time gaps. Elapsed time was multiplied by 30 while applying MF2-LSTM since the time lapse is measured in months.

MSE	T-LSTM	MF1-LSTM	MF2-LSTM	LSTM
Mean	0.50	0.53	0.51	0.51
Std	0.018	0.017	0.012	0.017

are reported in Table 5. The main observation about the target features in Table 5 is that they are related to the effects of Parkinson's disease on the muscle control such as finger tapping, rigidity, and hand movements. In addition, T-LSTM predicted the target value of Bradykinesia, which encompasses several of the problems related to movement, and MoCA (Montreal Cognitive Assessment) Total Score, which assesses different types of cognitive abilities with lower error than other methods. This result shows that the reported target features are sensitive to elapsed time irregularities and discounting the short-term effects by the subspace decomposition of memory cell helps to alleviate this sensitivity.

4.2.2 Patient Subtyping of PPMI Data. In this experiment, T-LSTM auto-encoder was used to obtain subtypes of the patients in the PPMI dataset. The T-LSTM encoder was used to learn a representation from the input feature sequence of each patient and the T-LSTM decoder generated the target sequence. Parameters of the auto-encoder were learned to minimize the squared error between the original target sequence and the predicted target sequence. The learned representations were used to cluster the patients by the k-means algorithm as discussed before.

Since we do not know the ground truth for the clustering, we conducted a statistical analysis to assess the subtyping performance. For this purpose, clustering results were statistically analyzed at the time of 6 years follow-up in the PPMI study. Features including demographics, motor severity measures such as Unified Parkinson's Disease Rating Scale (MDSUPDRS), Hoehn and Yahr staging (H&Y),

Table 5: Some common target features from PPMI dataset on which T-LSTM performed better than LSTM and MF1-LSTM during 10 trials. These target features are mainly related to the effects of Parkinson's disease on muscle control.

Code	Name
NP3BRADY	Global spontaneity of movement
NP3RIGRU	Rigidity - RUE(Right Upper Extremity)
NP3FTAPR	Finger Tapping Right Hand
NP3TTAPR	Toe tapping - Right foot
NP3PRSPR	Pronation-Supination - Right Hand
NP3HMOVR	Hand movements - Right Hand
NP3RIGN	Rigidity - Neck
NP2DRES	Dressing
PN3RIGRL	Rigidity - RLE (Right Lower Extremity)
DFBRADYP	Bradykinesia present and typical for PD
NP3RTARU	Rest tremor amplitude - RUE
NP3PTRMR	Postural tremor - Right Hand
MCATOT	MoCA Total Score

non-motor manifestations such as depression, anxiety, cognitive status, sleep disorders, imaging assessment such as DaTScan, as well as cerebrospinal fluid (CSF) biomarkers were taken into account. In order to interpret the clustering results in terms of subtyping, we compared the clusters using Chi-square test for the categorical features, F-test for the normal continuous features, Kruskal-Wallis test for the non-normal continuous features, and Fisherfis exact test for the high sparsity features. According to the previous Parkinson's disease studies, if the p-values of the aforementioned features are less than 0.05, a significant group effect is considered for the associated features [15]. Thus, if a method can obtain higher number of features with small p-values, it means that method provides a more sensible patient subtyping result.

Since we do not know the ground truth groups of the patient population, we tried several k values for the k-means algorithm. We often observed that there were two main clusters, therefore we reported the clustering results for $k = 2$. We conducted several tests with different parameters. According to our observation, LSTM yielded very few features with p-values less than 0.05 and most of the patients were generally grouped into one cluster. In Table 6, features of small p-values and cluster means of the features are presented for T-LSTM, MF1-LSTM and MF2-LSTM. As it can be seen from the table, T-LSTM has more discriminative features than MF1-LSTM and MF2-LSTM.

In Table 6, high cluster mean indicates that the symptoms of the corresponding feature are more severe for that cluster and the PD patients have lower cluster mean for DaTScan feature. Note that one of the observed features of T-LSTM in Table 6 is MoCA which was predicted better by T-LSTM in the target sequence prediction experiment. Finally, we illustrate the patient subtyping results of T-LSTM with heat map illustration in Figure 6. In this figure, shade of red color represents the cluster mean which is higher than the total mean of the patients and the shades of blue color show lower mean values for the corresponding feature with the p-value< 0.05. Subtypes and features which are significant for each subtype can be observed from the heat map. For instance, DaTSCAN features were found to be significant for subtype I, whereas subtype II was defined by BJLO (Benton Judgement Line Orientation) and MoCA

Table 6: Results of the statistical analysis for T-LSTM, MF1-LSTM and MF2-LSTM. DaTScan1 corresponds to DaTScan SBR-CAUDATE RIGHT, DaTScan2 is DaTScan SBR-CAUDATE LEFT, and DaTScan4 is DaTScan SBR-PUTAMEN LEFT.

Feature	P-Value	Cluster1 Mean	Cluster2 Mean
T-LSTM			
BJLO	9.51×10^{-8}	16.5	24.7
MoCA	0.001	40.0	41.2
DaTScan1	0.042	2.29	2.07
DaTScan2	0.027	2.31	2.08
DaTScan4	0.001	1.4	1.1
MF1-LSTM			
CSF-Total tau	0.007	87.9	46.72
MoCA	2.16×10^{-17}	47.5	41.05
SDM	0.005	58.5	41.5
MF2-LSTM			
HVLT-Retention	0.03	0.84	0.83
SDM	0.007	36.61	41.68

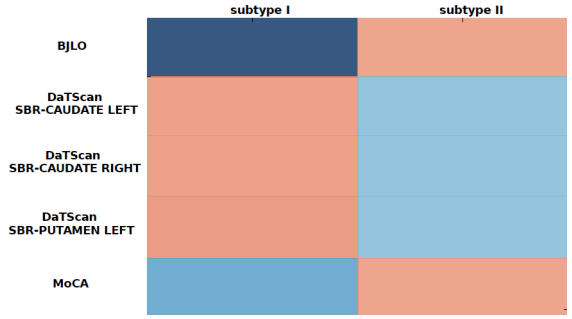


Figure 6: Heat map illustration of the patient subtyping results of T-LSTM for two clusters. Shade of red represents the cluster mean which is higher than the total mean of the patients and the shades of blue show lower mean values for the corresponding feature with p-value< 0.05.

features. Note that the dataset contains healthy subjects as well. It is known that PD patients have lower DaTScan SBR values than healthy subjects [29]. Hence, we can conclude from Figure 6 that subtype II can be considered as PD patients. We can also observe from Figure 6 that cluster means of BJLO and MoCA are very low (darker shades of blue) for subtype I compared to subtype II.

5 CONCLUSION

In this paper, we propose a novel LSTM unit, called time-aware LSTM (T-LSTM) which can deal with irregular elapsed times between the successive elements of sequential data. Examples include medical records which are complex temporal data with varying sequence lengths and elapsed times, and video sequence with missing frames. T-LSTM does not have any assumption about the elapsed time measure such that the time gap does not have to be measured in days or years and thus it can be adopted by other domains dealing with different types of sequences. T-LSTM adjusts the previous memory content of an LSTM unit by a decaying function of the elapsed time in a way that longer the time lapse, less the influence of the previous memory content on the current output. The proposed T-LSTM was tested for supervised and unsupervised tasks

on synthetic data and real world datasets. Patient subtyping, which can be defined as clustering sequential patient records, was analyzed on a publicly available real world dataset called Parkinson's Progression Markers Initiative (PPMI). For the subtyping purpose, T-LSTM auto-encoder was used to learn powerful representations for the temporal patient data, and the learned representations were used to cluster the patient population. In future work, we plan to apply the proposed approach to several other real datasets to observe the behaviour of our method for patient populations with different characteristics.

ACKNOWLEDGMENTS

Data used in the preparation of this article were obtained from the Parkinson's Progression Markers Initiative (PPMI) database (http://www.ppmi-info.org/data). For up-to-date information on the study, visit http://www.ppmi-info.org. PPMI a public-private partnership is funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners, including abbvie, Avid, Biogen, Bristol-Mayers Squibb, Covance, GE, Genentech, GlaxoSmithKline, Lilly, Lundbeck, Merk, Meso Scale Discovery, Pfizer, Piramal, Roche, Sanofi, Servier, TEVA, UCB and Golub Capital. This research is supported in part by the Office of Naval Research (ONR) under grants number N00014-17-1-2265 (to JZ and AKJ), N00014-14-1-0631 (to JZ and AKJ) and National Science Foundation under grants IIS-1565596 (to JZ), IIS-1615597 (to JZ) and IIS-1650723 (to FW).

REFERENCES

[1] 2017. Health IT Dashboard. Retrieved at https://dashboard.healthit.gov/quickstats/quickstats.php. (2017).

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2014. Representation Learning: A Review and New Perspectives. *arXiv:1206.5538v3[cs.LG]* (2014). https://arxiv.org/abs/1206.5538

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks* 5, 2 (March 1994), 157–166.

[4] Chao Che, Cao Xiao, Jian Liang, Bo Jin, Jiayu Zhou, and Fei Wang. 2017. An RNN Architecture with Dynamic Temporal Matching for Personalized Predictions of Parkinson's Disease. In *Proceedings of the 2017 SIAM International Conference on Data Mining*.

[5] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. 2015. Deep Computational Phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 507–516.

[6] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2016. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *arXiv preprint arXiv:1606.01865* (2016).

[7] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoderfi?!Decoder for Statistical Machine Translation. *arXiv:1406.1078v3[cs.CL]* (2014). https://arxiv.org/abs/1406.1078

[8] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, and Dzmitry Bahdanau et al. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078v3[cs.CL]* (2014). https://arxiv.org/pdf/1406.1078v3

[9] Edward Choi, Mohammad Taha Bahadori, Andy Schuetzy, Walter F. Stewarty, and Jimeng Sun. 2016. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. *arXiv:1511.05942v11 [cs.LG]* (2016). https://arxiv.org/pdf/1511.05942v11.pdf

[10] Edward Choi, Mohammad Taha Bahadori, Andy Schuetzy, Walter F. Stewarty, and Jimeng Sun. 2016. RETAIN: Interpretable Predictive Model in Healthcare using Reverse Time Attention Mechanism. *arXiv:1608.05745v3 [cs.LG]* (2016). https://arxiv.org/pdf/1608.05745v3.pdf

[11] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. 2016. Multi-layer Representation Learning for Medical Concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD16*. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/2939672.2939823

[12] Ivo D. Dinov, Ben Heavner, Ming Tang, Gustavo Glusman, Kyle Chard, and Mike Darcy et al. 2016. Predictive Big Data Analytics: A Study of Parkinson's Disease Using Large, Complex, Heterogeneous, Incongruent, Multi-Source and Incomplete Observations. *PLoS ONE* 11, (8):e0157077 (August 2016).

[13] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. 2016. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. *arXiv:1411.4389v4[cs.CV]* (2016). https://arxiv.org/pdf/1411.4389.pdf

[14] Cristobal Esteban, Oliver Staeck, Yinchong Yang, and Volker Tresp. 2016. Predicting Clinical Events by Combining Static and Dynamic Information Using Recurrent Neural Networks. *arXiv:1602.02685v1 [cs.LG]* (2016). https://arxiv.org/pdf/1602.02685v1.pdf

[15] Seyed-Mohammad Fereshtehnejad, Silvia Ros-Romenets, Julius B. M. Anang, and Ronald B. Postuma. 2015. New Clinical Subtypes of Parkinson Disease and Their Longitudinal Progression: A Prospective Cohort Comparison With Other Phenotypes. *JAMA Neurol* 72, 8 (2015), 863–873.

[16] Alex Graves, Abdel rahman Mohamed, and Geoffrey Hinton. 2013. Speech Recognition with Deep Recurrent Neural Networks. *arXiv:1303.5778[cs.NE]* (2013). https://arxiv.org/abs/1303.5778

[17] Joyce C Ho, Joydeep Ghosh, and Jimeng Sun. 2014. Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 115–124.

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[19] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, A Freely Accessible Critical Care Database. *Scientific Data* 3 (2016).

[20] Uri Kartoun. 2016. A Methodology to Generate Virtual Patient Repositories. *arXiv:1608.00570 [cs.CY]* (2016). https://arxiv.org/ftp/arxiv/papers/1608/1608.00570.pdf

[21] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[22] Zachary C. Lipton, David C. Kale, Charles Elkan, and Randall" Wetzell. 2016. Learning to Diagnose with LSTM Recurrent Neural Networks. *arXiv:1511.03677v6 [cs.LG]* (2016). https://arxiv.org/pdf/1511.03677v6.pdf

[23] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. 2008. *Introduction to Information Retrieval.* Cambridge University Press.

[24] Benjamin M. Marlin, David C. Kale, Robinder G. Khemani, and Randall C. Wetzel. 2012. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*. ACM, 389–398.

[25] Trang Pham, Truyen Tran, Dinh Phung, and Svetha Vankatesh. 2016. DeepCare: A Deep Dynamic Memory Model for Predictive Medicine. *arxiv:1602.00357v1 [stat.ML]* (February 2016). https://arxiv.org/pdf/1602.00357v1.pdf

[26] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. 2016. Unsupervised Learning of Video Representations using LSTM. *arXiv:1502.04681v3[cs.LG]* (2016). https://arxiv.org/abs/1502.04681

[27] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1711–1721.

[28] Ting Xiang, Debajyoti Ray, Terry Lohrenz, Peter Dayan, and P Read Montague. 2012. Computational Phenotyping of Two-person Interactions Reveals Differential Neural Response to Depth-of-thought. *PLoS Comput Biol* 8, 12 (2012), e1002841.

[29] Yu Zhang, I-Wei Wu, Duygu Tosun, Eric Foster, and Norbert Schuff. 2016. Progression of Regional Microstructural Degeneration in Parkinsonfis Disease: A Multicenter Diffusion Tensor Imaging Study. *PLOS ONE* (2016).

[30] Jiayu Zhou, Zhaosong Lu, Jimeng Sun, Lei Yuan, Fei Wang, and Jieping Ye. 2013. Feafiner: biomarker identification from medical data through feature generalization and selection. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1034–1042.

[31] Jiayu Zhou, Fei Wang, Jianying Hu, and Jieping Ye. 2014. From Micro to Macro: Data Driven Phenotyping by Densification of Longitudinal Electronic Medical Records. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.

[32] Jiayu Zhou, Lei Yuan, Jun Liu, and Jieping Ye. 2011. A multi-task learning formulation for predicting disease progression. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 814–822.

[33] Xiaoqiang Zhou, Baotian Hu, Qingcai Chen, and Xiaolong Wang. 2015. An Auto-Encoder for Learning Conversation Representation Using LSTM. In *Proceedings of the 22nd International Conference on Neural Information Processing, ICONIP 2015*. 310–317. http://dx.doi.org/10.1007/978-3-319-26532-2_34