Journal of Biomedical Informatics 83 (2018) 167–177

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Modeling asynchronous event sequences with RNNs

Stephen Wu[a,*], Sijia Liu[a], Sunghwan Sohn[a], Sungrim Moon[a], Chung-il Wi[b], Young Juhn[b], Hongfang Liu[a]

[a] Department of Health Sciences Research, Mayo Clinic, Rochester, MN, United States
[b] Department of Pediatrics, Mayo Clinic, Rochester, MN, United States

ARTICLE INFO

Keywords:
Temporal data
Deep learning
Electronic health records
Asthma

ABSTRACT

Sequences of events have often been modeled with computational techniques, but typical preprocessing steps and problem settings do not explicitly address the ramifications of timestamped events. Clinical data, such as is found in electronic health records (EHRs), typically comes with timestamp information. In this work, we define event sequences and their properties: *synchronicity*, *evenness*, and *co-cardinality*; we then show how asynchronous, uneven, and multi-cardinal problem settings can support explicit accountings of *relative time*. Our evaluation uses the temporally sensitive clinical use case of pediatric asthma, which is a chronic disease with symptoms (and lack thereof) evolving over time. We show several approaches to explicitly incorporating relative time into a recurrent neural network (RNN) model that improve the overall classification of patients into those with *no asthma*, those with persistent *asthma*, those in long-term *remission*, and those who have experienced *relapse*. We also compare and contrast these results with those in an inpatient intensive care setting.

1. Introduction

Sequences of events have often been modeled with techniques such as Hidden Markov Models (HMMs), Conditional Random Fields (CRFs), Recurrent Neural Networks (RNNs), and their derivatives. Here, we zoom in on the typical temporal assumptions being made in these types of models, and provide evaluations on a use case of medical information (asthma diagnoses) that could intuitively benefit from more explicit accountings of event timing.

We will define event sequences and differentiate between *synchronous* v. *asynchronous*, *co-cardinal* v. *multi-cardinal*, and *even* v. *uneven* pairs of such sequences (see Section 3.1). In particular, where most methods transform event sequences to be synchronous, co-cardinal, and even, we will consider the ramifications of using event sequences that are either synchronous, co-cardinal, but *un*even; or synchronous, *multi*-cardinal, and *un*even.

To evaluate the representational power and computational tractability of these decisions, we compare several RNN architectures in the assumed setting (synchronized, co-cardinal, even) to similar networks that remove those assumptions (allowing in turn asynchronous, uneven, and multi-cardinal data). We also compare alternative representations of event timing, including the typical *elision*, the *Markov* time since previous sequence elements, and a relative time based on *landmark* events.

We chose RNN sequence models for computation in the current work, despite significant previous computational models that made explicit accounting for duration or time, notably the class of Hidden Semi-Markov Models [54] and of Irregular-Time Bayesian Networks [42]. In part, this choice is due to the fact that with RNN-based models, we can maintain the fundamental structure of the models across different treatments of event sequences: recurrent units of long short-term memory (LSTM). With the current research milieu of minimally-bounded excitement around deep learning, we also take the opportunity to demonstrate how some standard deep neural network components attempt to handle event sequences, and how those approaches differ from more explicit accounts.

Our exploration is driven by the clinical use case of pediatric asthma, whose status in a patient evolves over time, and could be classified as *no asthma*, *asthma*, *remission*, or *relapse*. Asthma is the most common chronic disease among children and one of the 5 most burdensome diseases in the US. Measuring the timing of events, as this work does computationally, is an important step in identifying the exact asthma prognosis (since clinically, the statuses above are thought of as forward-looking prognoses like persistent *asthma* vs. long-term *remission*). Previous studies showed that 10–70% of childhood asthma was outgrown by adulthood [31,8]; 50% of children achieved remission; 24% achieved long-term remission (without relapse during a 1̃0-year follow-up period) [22].

[*] Corresponding author.
E-mail address: wu.stephen.t@gmail.com (S. Wu).

https://doi.org/10.1016/j.jbi.2018.05.016
Received 26 June 2017; Received in revised form 10 May 2018; Accepted 26 May 2018
Available online 05 June 2018

We should note that this motivating problem setting (chronic disease) differs from the setting in much of the related work on asynchronous time series data (critical care). Data from intensive care units (ICUs) such as the MIMIC III [24] or Physionet [47] datasets assume constant monitoring of a relatively small number of relevant signals; the corresponding problem may be cast as one of *missing data*, and default values are predictable and suitable. In contrast, infrequent hospital visits for chronic diseases result in sparse monitoring of data. Physicians describe only the relevant condition of a patient, and default values (e.g., based on the principle of homeostasis) may not adequately characterize the "missing" values. Thus, we have framed our problem as one of event sequences rather than missing data. Because of these substantial differences, Section 4 compares the corpus for our chronic disease use case (Olmsted County Birth Cohort) with the critical care use case (Physionet 2012), and our experiments evaluate the RNN model performance on the two datasets.

After some related work (Section 2), we will discuss event sequences (Section 3.1), synchronization (Section 3.2), relative timing (Section 3.3), and LSTM-based neural networks that allow computation over event sequences (Sections 3.4 and 3.5). We describe the different experimental settings of pediatric asthma and ICU data (Section 4) and then present results (Section 5) with discussion (Section 6).

2. Related work

2.1. Time representation in clinical events

EHRs contain longitudinal patient health events associated with the temporal information. In clinical research the temporal information has been handled in different ways: complete elision, actual time of events, intervals of time, or aggregation of time [32,44,7,50]. Shahar's early work on "temporal abstraction" [46] directly addressed the processing of timestamped, longitudinal (biomedical) data. The event sequences, sequence transformations, and experimentation with neural network methods can all be considered subsets of the larger temporal abstraction problem (towards a particular goal), similar in spirit to more recent work [32,33]. The model of the current work re-casts the temporal abstraction problem in a data-centric view that requires the minimal number of assumptions concerning the data.

Temporal abstractions provide meaningful – or at least quantifiable – representations of time information, which can then be used in an applied problem. Dagliati et al. proposed a temporal mining approach to detect frequent healthcare histories in breast cancer patients [10]. Event sequences were represented as one-hot encodings, and consecutive occurrences were aggregated into single events. The temporal information is subsequently enriched with event durations and intervals for temporal pattern mining. A follow-up study on temporal association rule mining models encoded the event intervals, state (e.g. low, high) and trends (e.g. increase, decrease) into different event categories [37]. Events within an arbitrary maximum interval were merged into one event representation.

In an endeavor to better account for the temporal information of health events in EHR data, Zhao et al. explored a temporal data representation that retained sequential information based on symbolic sequence representations of time series data [55]. They observed that their random sequence model outperformed the single sequence model on detecting adverse drug events and demonstrated that time series of various lengths can be used as features for predictive models. Hripcsak et al. investigated time parameterization on laboratory tests and measured variability of rate and magnitude of changes [21]. They found that sequence time, which is simply counting measurements from some point, produced more stationary in changes and accurate predictions in a single Gaussian process modeling than using actual event time. Other researchers focused on building a standards-driven infrastructure for the efficient secondary use of EHR including temporality data. Rea et al. investigated a standardized platform using a common clinical model and consolidated health informatics standard to maintain the temporality of events in the EHR data [43].

2.2. Computational models of time-series data

Sequences of events with time information have been represented in diverse computational models. The well-known Hidden Markov Models (HMMs) statistically model the generation of a sequence of observed evidence from unobserved events [41,14], and have been applied successfully in many domains. An extension of HMMs, Hidden Semi-Markov Models (HSMMs), relax the underlying stochastic process to be a semi-Markov chain rather than Markov chain (i.e., probability of change in the hidden state depends on the amount of time elapsed) and thus broadens its usage to a wide range of applications [35,54,25].

Temporal events have also been modeled in Bayesian networks [18,36], a probabilistic graphical model representing the conditional dependencies between a set of random variables. Van der Heijden et al. developed temporal Bayesian network learning with bootstrapping methods to predict exacerbation events of chronic obstructive pulmonary disease [49]. Dynamic Bayesian networks (DBN), which relate variables to each other over adjacent time steps, were also developed to generalize the capability of Bayesian networks to time series analysis [12,11,30]. However, discrete-time Markov models, such as HMMs and DBNs, are not well designed for events occurring irregularly over time [13,34], causing inefficient computation or information loss since they usually require the specification of a constant time interval between consecutive events. To address these issues, Ramati and Shahar [42] developed irregular-time Bayesian networks generalizing DBNs to address the temporal dynamics of processes, improving adaptation to the irregularly-timed data.

More recently, recurrent neural networks (RNNs) have also been applied to model temporal data processing. Lipton et al. [27] consider temporally irregular ICU data to be missing data. Their evaluation compared imputation and other established methods for dealing with missing data, and in the end found LSTMs with simple zero-imputation and missing data indicators to be the most effective in their intra-institutional dataset. Rethinking the bigger picture of health outcomes prediction through the lens of deep learning, Pham et al.'s recent work [38] introduced DeepCare, an RNN architecture designed to address irregularities, long-term dependencies, and representation of longitudinal healthcare data. Their evaluations on the chronic diseases of diabetes and mental health showed DeepCare to predict outcomes more accurately than unmodified RNNs or Markov models. Ma et al. adopted attention-based bidirectional RNNs for the task of diagnosis prediction on diabetes patients and Medicaid claims [29]. The model embeds the medical codes to vector presentation and utilizes the hidden state RNN units with attention for diagnosis category prediction.

Beyond this, some recent work on temporal data in RNNs directly modifies or augments the recurrent units themselves. Che et al. [6] also cast the problem as one of missing data, and use RNNs as a natural step in a longer stream of work to capture "informative missingness." They extend the gated recurrent unit (GRU) to include trainable decays (GRU-D), introducing additional input variables for masking the input and representing time intervals. In follow-up work, they also provide a health data-specific means of regularizing with a prior and present a mimic learning model that mirrors the effectiveness of their other models [5]. Their evaluations show substantial improvements in mortality prediction accuracy and other tasks on primarily ICU-related data. An alternative to GRU-D is Baytas et al.'s Time-aware LSTM [3]. This addresses the irregular time interval differences in medical records, adding a decay node on elapsed time to the cell memory of LSTM to model decreasing contribution of previous values.

Convolutional neural networks (CNNs) have also been successful at learning generalized feature extractors with shared weights, often over the spatial domain but also the temporal. Liu et al. implemented temporal-embedded CNNs to learn repeatedly occurring elements in

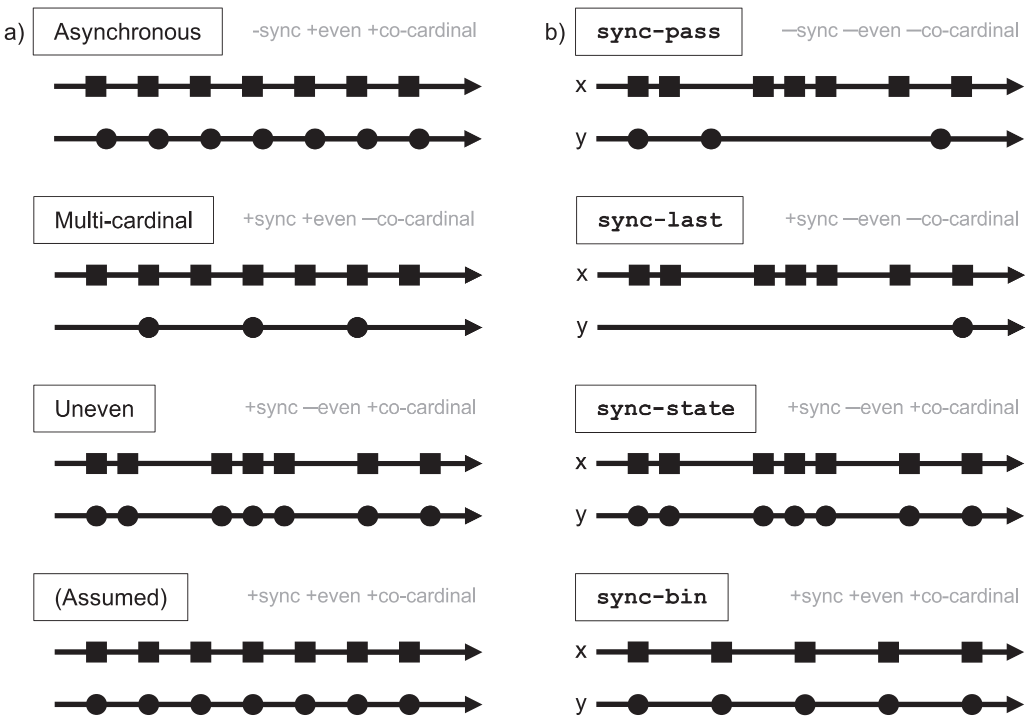

Fig. 1. Schematics of different types of event sequences; x may be events like asthma symptoms, y may be events like changes in asthma status. (a) The treatment of event sequences often implicitly models events as synchronized, co-cardinal, and even (Assumed); However, the reality is that they may be Asynchronous, Multi-Cardinal, and/or Uneven. (b) On asynchronous, uneven, multi-cardinal data, the four synchronization strategies `sync-pass`, `sync-last`, `sync-state`, and `sync-bin` transform event sequences; each distorts the original data differently and leads to a different treatment in computational models.

periodical time-series with its potential neighbors and demonstrated advantages over existing methods [28]. Cui et al. developed multi-scale CNNs that incorporate feature extraction and classification in a single platform using multi-branch layers and learnable convolutional layers [9]. This approach enables the networks to automatically extract features from different points of time-series data to better represent features. Recently, a deep CNN model has been developed to utilize the covariance structure over multiple time series to learn the group CNN structures to efficiently deal with high-dimensional multivariate time series [53].

All of these works are highly relevant to our proposed event sequences, and are interesting directions for future work. However, these works primarily focus on variants of NN models and input embeddings to modeling irregular time intervals and data missingness of EHR data, while our study addresses the fundamental properties of event sequences and explores their implementation in an RNN context. As such, any substantially different neural network architectures are out of the scope of this work. Since there is no current benchmark dataset on which these works agree, we have picked the Physionet Challenge 2012 data set, tested in Che et al.'s work, for evaluation comparison.

In our work, we make use of the basic LSTM RNN architecture [19], which is well-suited for learning from experience to classify and predict time series presented with unknown time lags between important events. One of the most successful applications using LSTM for a time series data is speech recognition [15,17,2]. It has also been applied in time series of stock prices [4] and sequence learning [45].

2.3. *Computational models of asthma*

Asthma status ascertainment is complex because it depends on multiple signs and symptoms (i.e., events) with respect to time (i.e., temporal information). Thus, data-driven approaches leveraging large EMR data with machine learning have begun to supplement or supplant traditional expert-based approaches to asthma studies. Machine learning models have been applied in various asthma studies to ascertain asthma status [39,51,50], detect airway obstruction in asthma [1], distinguish asthma phenotypes [20], predict subgroups of asthma and eczema [40], and predict asthma exacerbation [16,52].

3. Methods

3.1. *Event sequences*

We define an **event** as a time-stamped observation $e_x = (x, t_x)$, where some value x occurs at corresponding time t_x. An **event sequence** is then a sequence of pairs, $\vec{e}_x = (x_1, t_{x_1}), (x_2, t_{x_2}), \ldots, (x_n, t_{x_n})$. In this sequence, each element describes measurements of the same variable over time; we can abbreviate this n-element sequence $\vec{x} = x_1, x_2, \ldots, x_n$ (or alternatively, $\vec{x}_{1:n}$). We will often write this with a "default" event (x_0, t_{x_0}), where $t_{x_0} = -\infty$. An event sequence is **even** if $t_{x_{i+1}} - t_{x_i} = t_{x_i} - t_{x_{i-1}}$ for $i = 2, \ldots, n-1$. Conversely, an event sequence is **uneven** if there exists some i for which $t_{x_{i+1}} - t_{x_i} \neq t_{x_i} - t_{x_{i-1}}$. N event sequences are even if each event sequence is individually even. If any of the N are uneven, then the N event sequences are considered uneven.

Most traditional time-series sequences, such as videos, images, and audio signals are *even* event sequences, despite the fact that these signals arise from continuous-time random processes. Typically, only the $\vec{x}$ vector is used, and the accompanying timing vector $\vec{t}_x = t_{x_0}, t_{x_1}, \ldots, t_{x_n}$ is discarded. This is clearly a lossy transformation of data. Unfortunately, excluding $\vec{t}_x$ like this treats timings as evenly spaced, which may be less appropriate for other problem settings. Biomedical data, such as longitudinal patient histories found in EHRs, frequently violate the assumption of even spacing. We therefore assume that we begin with event timings that are unevenly spaced unless otherwise specified, and explicitly include $\vec{t}_x$ in our modeling of events.

Adding a second event sequence $\vec{e}_y$ of length m alongside $\vec{e}_x$ of length n, we may expect that the values in the sequences $\vec{x}$ and $\vec{y}$ may be different. However, because of the timing components, the two event sequences may be similar and different in more ways than these values. We define the two sequences to be **co-cardinal** if $n = m$. Whenever $n \neq m$ (i.e., lengths differ), we will call $\vec{e}_x$ and $\vec{e}_y$ **multi-cardinal**.

Furthermore, we will say that $\vec{e}_x$ and $\vec{e}_y$ are **synchronous** when $\vec{t}_x \subseteq \vec{t}_y \vee \vec{t}_y \subseteq \vec{t}_x$, i.e., that one of the timings is fully a subset of the other. When this does not hold, $\vec{e}_x$ and $\vec{e}_y$ are **asynchronous** event sequences. To test for synchronicity, then, we check if the shorter sequence is a subset of the longer sequence. If so, the superset event timings can be considered "reference" timings. Note that this does not imply co-cardinality — the event sequences are not necessarily the same length.

Fig. 1a gives examples of these often overlooked properties of event sequences. It is straightforward to extend these properties to N event sequences. If co-cardinality (synchronicity) holds for each of the $\binom{N}{2}$ subset pairs of event sequences, it holds for the N. If not, the N event sequences are multi-cardinal (asynchronous).

3.2. Synchronizing event sequences

Traditional sequence prediction and classification techniques assume that timings are both evenly spaced and synchronous. In chronic disease settings, extracted information is sparse, with most variables infrequently observed; synchronicity, evenness, and co-cardinality are in general not known. To utilize informative irregularities in time-series data, some means of *synchronization* is necessary.

Here, we define several synchronization strategies for any two (or any N) event sequences, $\overrightarrow{e}_x$ and $\overrightarrow{e}_y$, with no other requirements on the event sequences' properties. Each strategy leaves the N event sequences with different evenness, cardinality, and synchronicity properties.

A synchronization strategy transforms the timings and/or values of the sequences. Below, we use operators inspired by (but not consistent with) semiring notation. The binary operator $\otimes$ indicates the concatenation of two sequences. The binary operator $\oplus$ below will typically indicate a sum for real or integer variables, a disjunction $\vee$ for boolean variables, or the last value in a sequence of categorical variables. These synchronization strategies are transformations producing different outputs $\overrightarrow{e}^{*}_{x}$ and $\overrightarrow{e}^{*}_{y}$, as illustrated in Fig. 1b.

3.2.1. Synchronize-pass

The `sync-pass` process is a no-op, returning unmodified sequences $\overrightarrow{e}^{*}_{x} = \overrightarrow{e}_x$ and $\overrightarrow{e}^{*}_{y} = \overrightarrow{e}_y$. In general, then, these sequences may be unevenly spaced, asynchronous, and multi-cardinal. There are some downstream tasks with event sequences that do not require synchronization, including the downstream task of a different synchronization procedure.

3.2.2. Synchronize-last

The `sync-last` transformation produces a one-element $\overrightarrow{e}_y$ sequence corresponding to the final element of $\overrightarrow{e}_x$. If used in a case where $\overrightarrow{e}_y$ is sequence of outcome variables, `sync-last` can be used in preparing a sequence classification task.

$$\begin{aligned} &\overrightarrow{e}^{*}_{y} = (y_i,\ t_{x_n}), \quad \text{where } i = \underset{\iota}{\operatorname{argmin}}\ t_{x_n} - t_{y_\iota} \quad \text{for } t_{y_\iota} < t_{x_n} \\ &\overrightarrow{e}_x = \overrightarrow{e}_x \end{aligned} \tag{1}$$

Note that the edge case (no ι for which $t_{y_\iota} < t_{x_n}$) can be avoided if we have prepended the "default" value for $e_{y_0} = (\text{default}, -\infty)$. The output of `sync-last` is an unmodified $\overrightarrow{e}_x$ and a single-element $\overrightarrow{e}_y$ whose timing is consistent with e_{x_n}; it is synchronous but not necessarily even or co-cardinal.

3.2.3. Synchronize-state

The `sync-state` transformation assigns the $\overrightarrow{e}_x$ timings to $\overrightarrow{e}^{*}_{y'}$, namely, $\overrightarrow{t}_y = \overrightarrow{t}_x$. $\overrightarrow{t}_y$ then serves as the "reference" timing for both $\overrightarrow{e_x}$ and $\overrightarrow{e_y}$. It is intended for when $\overrightarrow{e}_y$ is an event sequence in which each event describes the beginning (or equivalently, the changing) of a *state*. For example, in the asthma data used for our evaluation, a patient may have a sequence $y_1 = \textit{asthma}$, $y_2 = \textit{remission}$, $y_3 = \textit{relapse}$, $y_4 = \textit{remission}$, where the respective timings mean that each progressive asthma-related state has begun.

$$\begin{aligned} &\overrightarrow{e}^{*}_{y} = \bigotimes_{j=0}^{m} \left(\bigoplus_{i=0}^{n-1} (y_i,\ t_{xj}) \right), \quad \text{where } t_{y_i} < t_{xj} \leqslant t_{y_i+1} \\ &\overrightarrow{e}^{*}_{x} = \overrightarrow{e}_x \end{aligned} \tag{2}$$

We see that `sync-state` assigns the $\overrightarrow{t}_x$ timings to the event sequence $\overrightarrow{e}^{*}_{y}$. To assign the values for $\overrightarrow{y}$, these timings are partitioned by the inequalities, depending on their relation to the $\overrightarrow{t}_y$ timings. The output of `sync-state` leaves $\overrightarrow{e}_x$ intact, but transforms $\overrightarrow{e}_y$ into an unevenly spaced, synchronous, co-cardinal $\overrightarrow{e}^{*}_{y}$.

3.2.4. Synchronize-bin

The `sync-bin` transformation expands both $\overrightarrow{e}_x$ and $\overrightarrow{e}_y$ with padded (null or zero) values. We observe that $\overrightarrow{t}_x$ and $\overrightarrow{t}_y$, while conceived of as continuous-time variables, can typically be binned or sampled into discrete time windows. For example, patient information can be binned by month (e.g., for a longitudinal population study), day (e.g., for medication side effects), or hour (e.g., for intensive care visits). The size of a bin will affect the total number of bins B and the sparsity of the resulting representation.

We define an evenly spaced bin-timing vector $\overrightarrow{t}_b = t_{b_0}, \ldots, t_{b_B}$, where $t_{b_0} \leqslant \min(t_{x_0}, t_{y_0})$ and $t_{b_B} > \max(t_{y_n}, t_{y_m})$.

$$\begin{aligned} &\overrightarrow{e}^{*}_{y} = \bigotimes_{i=0}^{B-1} \left(\bigoplus_{j} y_j,\ t_{b_i} \right), \quad \text{where } t_{b_i} \leqslant t_{y_j} < t_{b_{i+1}} \\ &\overrightarrow{e}^{*}_{x} = \bigotimes_{i=0}^{B-1} \left(\bigoplus_{j} x_j,\ t_{b_i} \right), \quad \text{where } t_{b_i} \leqslant t_{x_j} < t_{b_{i+1}} \end{aligned} \tag{3}$$

Essentially, the sequence information of both $\overrightarrow{e}_x$ and $\overrightarrow{e}_y$ is aggregated within each bin by the $\oplus$ operator. There is an important edge case, which may describe the majority of bins in medical data: there may be no fitting j values, i.e., no contents in a bin. Some legitimate default value must be specified (e.g., null category, false binary, 0 integer) in order for `sync-bin` to be a well-formed sequence. These are tantamount to "inferences" at previously unspecified time points, which may or may not be tantamount to distorting the "evidence" in event sequences.

Unlike the strategies above, this can easily be extended to an arbitrary number of asynchronous event sequences, and results in output $\overrightarrow{e}^{*}_{x}$ and $\overrightarrow{e}^{*}_{y}$ that are not only synchronous, but also evenly spaced and of co-cardinal lengths. Thus, the `sync-bin` transformation allows for traditional sequence analysis techniques to be applied, once default values are accounted for.

3.2.5. Temporal aggregation

As an aside, the temporal aggregation approaches from Wu et al.'s temporal aggregation [50] can be cast as transformations of sequences that are similar to (not subsumed by) synchronization. We constrain our discussion to feature vectors $\overrightarrow{e}_{x1}$, $\overrightarrow{e}_{x2}$, ... and a single state vector $\overrightarrow{e}_y$, where the e_{x_i} and e_y are discrete.

In a classification problem $\mathbf{P}(\overrightarrow{y} \mid \overrightarrow{e}_{x1}, \overrightarrow{e}_{x2}, \ldots)$, temporal aggregation attempts to "roll up" the information of each $\overrightarrow{e}_{x_i}$ into a single statistic $x^{\circ}{}_{i}$. Namely, we solve for $\mathbf{P}(\overrightarrow{y} \mid x^{\circ}{}_{1}, x^{\circ}{}_{2}, \ldots)$ instead of the original event sequences.

The aggregated statistic x° may itself be as simple as a step function or a summation; or it may be more complicated – for example, calculated using probability distributions:

$$x^{\circ} = \sum_{i=0}^{n} \lambda_i \cdot \mathbf{P}(t_y - t_{x_i} \mid x_i, y) \cdot \mathbf{P}(x_i, y) \tag{4}$$

To make this kind of approach [50] feasible, `sync-state` needed to be applied to remove the dependence on $\overrightarrow{t}_y$. Also, `sync-last` needed to be applied when the classification was for the y value at a single time point. Thus, while temporal aggregation summarizes $\overrightarrow{e}_{x_i}$, it does not fully solve the issue of $\overrightarrow{e}_y$ and thus does not obviate the need for the synchronization strategies in this article.

3.3. Relative time

As noted previously, it is normal to approximate or ignore the timing information defined in Section 3.1, and there is good reason for it. An absolute time value like “9/3/2006” is meaningless without context. Furthermore, it does not add information concerning the event unless that time value can be interpreted in light of other event timings, e.g., a previous event occurring at “12/14/2005,” or even a list of events “12/16/2000, 3/24/2003, 12/14/2005.”

Even without knowing the event *values* at our example dates, we can still characterize the temporal unfolding of this event sequence as different (with respect to evenness) from a sequence such as “3/14/2001, 3/14/2002, 3/14/2003, 3/14/2004,” and this is different (in scale) from “9/16/2012, 9/23/2012, 9/30/2012, 10/7/2012.”

A simple accounting of relative time (which we will write $\widetilde{t}$) might be $\widetilde{t} = t_y - t_x$ for an event from sequence x and another from sequence y. More generally, we define $d(\cdot)$ to be a distance function between the two events e_y and e_x (or their timings). Many time parameterizations [21] are possible, e.g., polynomial time warping $d(t_y, t_x) = (t_y - t_x)^{\frac{1}{3}}$, sequence time (the difference in indices when e_x and e_y are synchronized and co-cardinal), or warping based on probability density functions [50]. However, for simplicity, we will assume that distance is a simple subtraction between the timings: $d(e_y, e_x) = d(t_y, t_x) = t_y - t_x$.

Fig. 2 gives an example of 3 possible definitions of relative time, which we further describe below.

3.3.1. Markov relative time

Disease histories may exhibit bursts of activity. Thus, the rate of change (and timing of more than just the immediately previous event) may be important, and can be captured with *Markov* relative time. Making a Markov assumption, a 1–*Markov* relative timing is to define $\widetilde{t}_{x_i} = d(t_{x_i}, t_{x_{i-1}})$. This captures the intuition that the state $e_{x_{i-1}}$ had a duration of $\widetilde{t}_{x_i, x_{i-1}}$, and has been applied to models such as HSMMs. A 2–*Markov* relative timing is $\widetilde{t}_{x_i} = d(t_{x-i}, t_{x_{i-2}})$.

We allow for *Markov* relative time to take a *#h* history parameter, indicating how many previous events to consider. Thus, *Markov-#h* relative time is defined as:

$$\widetilde{t}_{x_i} = [d(t_{x_i}, t_{x_{i-1}}), d(t_{x_i}, t_{x_{i-2}}), \ldots, d(t_{x_i}, t_{x_{i-\#h}})] \qquad (5)$$

3.3.2. Landmark-any relative time

While Markov relative time captures the “velocity” of events, it is agnostic to what actually happened at each point in time. In *Landmark* relative time, the timing of a whole sequence $\overrightarrow{e}_x$ can be specified at each point in relation to any other landmark event. In a clinical setting, this centers the timeline around a significant event such as a birth date or the most recent occurrence of a particular symptom.

The *landmark-any* strategy chooses its landmark event to be the most recent non-zero value for any of the features in the feature vector. In Fig. 2, the 6/28 and 7/5 dates have no non-zero values; the relative time calculation at 7/29, for example, looks back to 6/23.

Landmark-any is similar to the *Markov* method, allowing for a *#h* history parameter. The main difference is that *landmark-any* will “skip” feature vectors that are completely empty (common in our asthma ascertainment evaluation setting).

3.3.3. Landmark-own relative time

A weakness of *Landmark-any* is that it assumes some dependence between each of the features, at least as far as relative timing is concerned. The opposite extreme is to consider each feature as completely independent of the others with respect to timing.

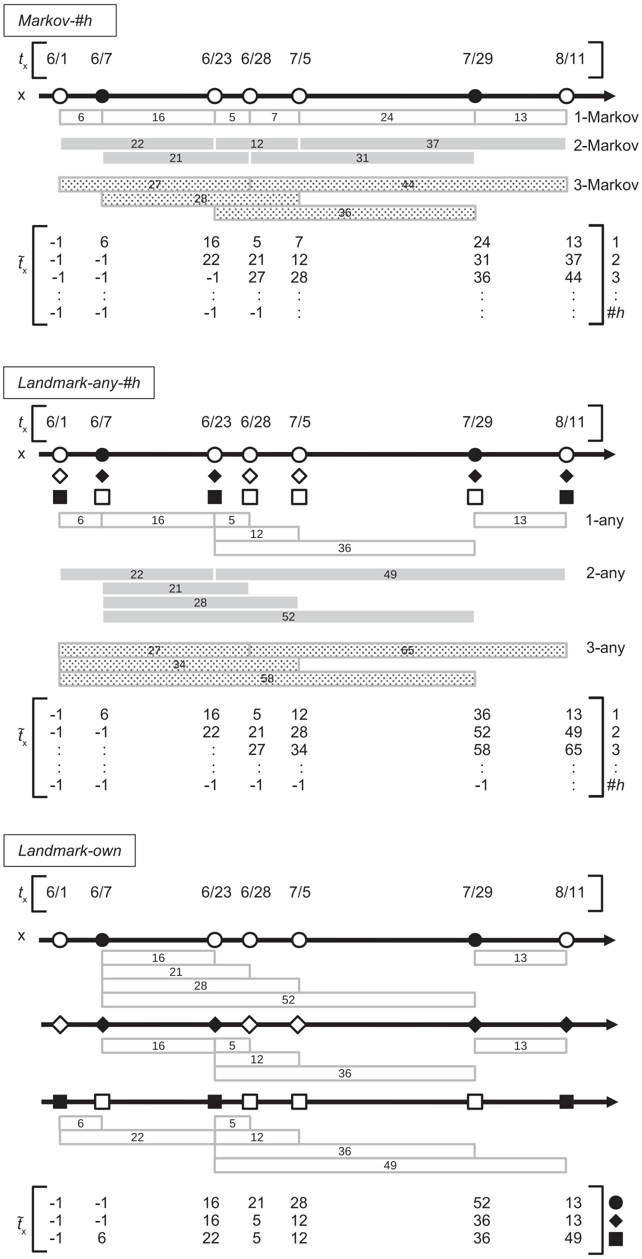

Fig. 2. Approaches to relative time calculation. Filled shapes are `true` features, while empty shapes are `false`; bars with numbers represent relative time calculations; the $\widetilde{t}_x$ matrices hold those relative time calculations. In the *Markov-#h* strategy, relative timing does not depend on the features' values, only on the existence of a feature. Relative time in the *Landmark-any-#h* strategy considers positive (shaded) values of *any* feature; in the *Landmark-own* strategy, only positive values of the *same* feature are considered.

The *landmark-own* strategy chooses its landmark event to be each feature's most recent non-zero value. As Fig. 2 makes clear, the dimensionality of the timing vector $\widetilde{t}_x$ is equal to the number of features, rather than a parameter *#h*. It therefore diverges from *Markov* and *landmark-any*, enabling the relative time to be highlighted for individual features.

3.3.4. Elision

In experiments, we contrast the above methods with *elision*, the removal of the timing component. This is the often the default used in preprocessing event sequences.

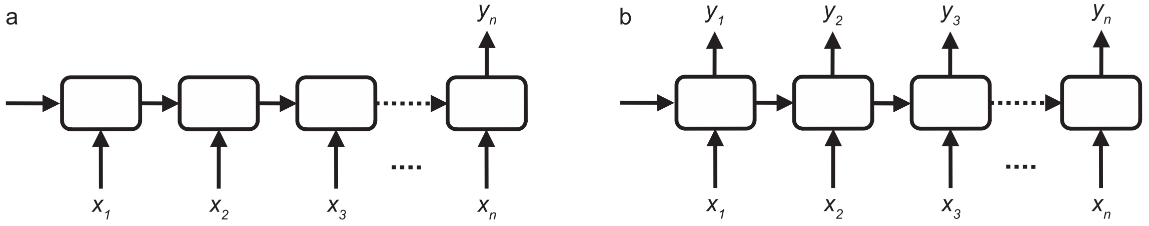

Fig. 3. Modeling sequences with Recurrent Neural Networks (RNNs) allows for very similar structure for both the tasks of: (a) sequence classification, with one output per sequence; and (b) sequence labeling, with one output per time point. Our evaluations are carried out on the sequence classification task.

3.4. RNNs with event sequences

We designed RNN models to solve the tasks of sequence classification (one class per sequence) and sequence labeling (one class per sample in a sequence). Fig. 3a and b illustrates the relatively minor conceptual difference between the classification and labeling tasks using RNNs. However, our evaluations only consider sequence classification; we leave sequence labeling to future work.

As a baseline approach, we make prominent use of LSTMs (see Hochreiter & Schmidhuber's original article [19] for a fuller description), after finding insignificant differences (unreported) with alternatives (e.g., gated recurrent units) that similarly solve the vanishing gradient problem. In preliminary tests we did not find anything beyond single-layered forward-directional LSTMs to be helpful, either; because our input data $\vec{e}_x$ consists of NLP-extracted features rather than actual embedded text, it is a relatively sparse signal compared to the typical input of RNNs.

A fully-connected layer follows the LSTM on the output, reducing the dimensionality of the LSTM output (32, for our default) to the one-hot-encoded dimensionality of the output (4, for each of *no asthma*, *asthma*, *remission*, *relapse*). For labeling, this fully-connected layer is applied at every time step (Fig. 3a), whereas for classification it occurs only at the final time step (Fig. 3b). A softmax function then reduces provides a normalized probability-like score for the value of y_i. Training minimized categorical cross-entropy loss.

When incorporating explicit relative time, we simply concatenate inputs as in Fig. 4a and b. Relative time was encoded in multiple ways, as described in Section 3.3.

3.5. Incorporating relative time

Two means of incorporating relative time into an RNN model are shown in Fig. 5.

For an event sequence $\vec{e}_x = (x_1, t_{x1}), (x_2, t_{x2}), \ldots$, we can create a relative time version: $(x_1, \tilde{t}_{x1}), (x_2, \tilde{t}_{x2}), \ldots$ using one of the strategies from Section 3.3.

Relative Time Concatenation. The Relative Time Concatenation strategy expresses the perspective that explicit timing information is useful to the LSTM itself. Relative timing events may not be properly scaled for the neural network, so we pass relative time events through a sigmoid activation function. To keep this method simple, we do not include linear weights. We then concatenate this scaled version of $\tilde{t}_{xi}$ to the input, effectively increasing the number of features that are available to the LSTM layer.

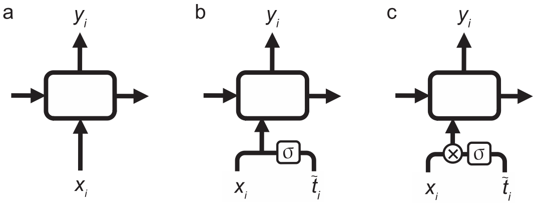

Fig. 5. Models for incorporating relative time into an RNN. (a) Elision; (b) Relative Time Concatenation; (c) Relative Time Gate. From the perspective of the recurrent unit (LSTM), Relative Time Concatenation effectively increases the size of the input vector, whereas the Relative Time Gate maintains the same size input. The dimensionality of $\tilde{t}_i$ and of the σ unit differ for *Markov*, *landmark-own*, and *landmark-any* relative time, as defined in Section 3.3.

Relative Time Gate. The Relative Time Gate strategy attempts to further stipulate that relative timing information primarily mediates the importance weight of new input x_i. Practically speaking, the relative timing vector $\tilde{t}_{xi}$ may be as small as a single scalar, or it may be multiple times the size of x_i; thus, we incorporate a fully-connected NN layer with sigmoid activation. This fully-connected layer outputs a vector of the same dimensionality as the input x_i, and the two are pointwise-multiplied. This does not increase the number of features available to the LSTM; it requires the DNN to learn an additional matrix.

4. Experimental setup

4.1. Asthma data

For this study, we used a subset of the Olmsted County Birth Cohort, which includes children born at Mayo Clinic Rochester, MN between 1997 and 2007 (n = 8525; median age at asthma onset and the last follow-up date were 1.6 and 10.7 years, respectively). Our subset includes the first 4013 patients born between 1997 and 2002. Patients are assumed to be born with *no asthma*. For patients who are diagnosed with *asthma* [22], asthma *remission* was defined by the absence of signs/symptoms of asthma, asthma-related medications, or health care services for three consecutive years; *relapse* was defined as occurrence of any of these events after achieving remission.

Children who belong to the three categories of asthma prognosis of this cohort defined by a rule-based NLP system (i.e., persistent *asthma* (15.18%), long-term *remission* (10.64%), and *relapse* after remission

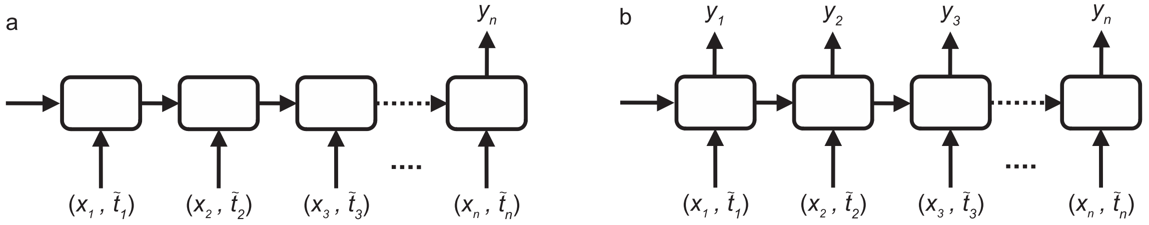

Fig. 4. RNN sequence modeling with explicit relative time, for (a) sequence classification, and (b) sequence labeling. Our evaluations are carried out on the sequence classification task.

Table 1
Corpus statistics for input variables in the Olmsted County Birth Cohort 1997–2002 subset, versus all input variables in the Physionet Challenge 2012 dataset. The Olmsted County Birth Cohort uses asthma-related variables extracted from text, whereas the Physionet variables were directly recorded in critical care settings.

	OC Birth Cohort	Physionet 2012
# sequences in corpus	4013 patients	4000 patients
# input variables	60 from text	33 measured
min–max event duration	[0, 6017] days	[0, 26] hours
$\mu \pm \sigma$ event duration	53.6 ± 131.4 days	0.632 ± 0.627 hours
min–max event density	[0.0010, 2.8000] ev/day	[0.0811, 4.3191] ev/hour
$\mu \pm \sigma$ event density	0.0198 ± 0.0488 ev/day	1.5985 ± 0.4838 ev/hour

(7.08%)) and those with *no asthma* (67.11%) served as a gold standard for the current study, as our previous showed reasonable concordance with manual chart review (92%) and manual chart review for this large cohort is infeasible.

We identified asthma prognosis with an NLP feature extraction component followed by a set of logic-based rules, similar to Wu et al.'s baseline approach [50]. These clinically motivated [22] rules included temporally significant specifiers; for example, the beginning of *remission* was counted as 3 years after the last recorded asthma symptom. There were a total of 60 binary features: 30 positive and 30 negative asthma-related Named Entities (NEs), such as "Viral Infection" and "NOT Viral Infection" indicating the existence of a mention or negated mention of that NE in clinical notes. Each extracted feature had a timestamp; features are not explicitly temporally aligned, but relative time implicitly aligns events with respect to other events.

We should note that the "gold standard" data in this Olmsted County Birth Cohort dataset is automatically produced by Sohn et al.'s rule-based detection algorithm [48], since the size of this dataset and the cost of clinician–annotators makes it infeasible to manually judge the data; thus the upper limit accuracy of an automatic method is 100%.

4.2. Comparison with ICU data

Since related work on asynchronous time series has focused on ICU settings, we evaluate the methods on the Physionet Challenge 2012 dataset. The dataset and mortality prediction task are described elsewhere [47], so Table 1 focuses on the ways in which this data differs from the setting of pediatric asthma.

First, we should note that the generation of input variables is different for the two settings. The Birth Cohort data had a per-document summary of asthma-related variables that were extracted from text; while detecting missing values would be an interesting NLP problem, it is out of the scope of this work. Physionet data was collected in the ICU.

What we see clearly in the corpus statistics is that the Birth Cohort data is on a significantly different time scale than the Physionet data. To account for this, it is possible to scale the units in Table 1 so that the two datasets have similar event durations and densities: for example, using 90-day increments (*quarters*) on the Birth Cohort would yield typical event durations of 0.5956 ± 1.46 quarters, and event densities of 1.782 ± 4.392 ev/quarter – comparable averages to the 0.632 ± 0.627 and 1.5985 ± 0.4838 values of the Physionet 2012 data.

However, it is obvious that the data characteristics are not just a difference in time scale; the standard deviation of event density is much higher for asthma data, suggesting a temporal distribution of events that is much more irregular than for ICU data. Thus, what time scale best represents the data and how to model the more highly irregular data are open questions. The event sequences proposed in this work are one attempt at solving these problems.

For our experiments, Physionet Challenge 2012 outcome variables (for the mortality prediction task) were transformed into timestamped events. We chose hours (for ICU) as the unit of time, rather than days (for asthma). Also, we represented Physionet input variables as binary events (1 if present, 0 if not). This allowed us to keep our systems consistent across the use cases of asthma and ICU settings. While this means our Physionet results do not make full use of the real-valued inputs in related work using Physionet [6] (any information in the values of the input variables is necessarily lost), it gives an analogous setting to our binarized asthma features.

4.3. Pre-evaluation parameters

We conducted preliminary experimentation on the classification task, comparing variant models against a baseline of a single forward LSTM layer with 32 units with random undersampling. Based on this, we made the following choices for our evaluation:

- LSTM as the recurrent unit. There is minimal difference between LSTMs and gated recurrent units (GRUs), which both seek to solve the vanishing gradient problem. A GRU unit had a *p*-value of $p = 0.9827$ compared to the LSTM, and thus there was no significant improvement.
- Single forward LSTM layer. No increased classification accuracy was seen with multi-layer (increased absolute performance at $p = 0.9063$, not significant) or bidirectional LSTMs (increased performance at $p = 0.8985$, not significant). This may be due to the simplicity of the 60 binary extracted features at each time point.
- Random undersampling. Because all patients start as having *no asthma* but a subset develop *asthma*, a sub-subset achieve *remission*, and a sub-subsubset experience *relapse*, the classification and labeling problems are both inherently class-imbalanced. We found random undersampling (taking the same number of samples as the smallest class) to be more effective on our data than nearest-neighbor-based (decreased F_1 at $p = 0.3535$, not significant) and other well-known approaches to addressing class-imbalance.
- RNN training for 50 epochs. Learning curves slowed down (but continued to show improvement) after 50 epochs. See Appendix A.1.
- 10-fold cross-validation. Performance averages (across folds) were relatively stable after 5 folds and more so after 10.
- Binary inputs. Asthma input features were already binary indicators, signaling the presence or absence of NLP-extracted events. Physionet input features were binarized to match, without showing performance degradation in preliminary tests (see Appendix A.2).

Despite these implementation decisions, the underlying goal of the experiments is not mainly to design the most accurate asthma status algorithm, but to characterize the performance impacts of modeling asynchronous event sequences.

5. Evaluation

5.1. Binary vs. multiclass sequence classification

The problem setting implicit in the asthma dataset of Section 4.1 is a multiclass classification problem, rather than just a binary one. Thus, in Table 2 we characterize how much more difficult of a problem the multiclass setting is, in the common setting where any explicit timing information (all $\tilde{t}_{x_i}$ and $\tilde{t}_{y_j}$) is removed.

We will use Table 2's multi-class results as our baseline (marked with a †) in Tables 3 and 4. This 10-fold cross-validated baseline has moderate variability ($\mu \pm \sigma$ for F_1 is 0.4976 ± 0.0440) across folds, and includes the preliminary experimentation and modeling decisions in Section 4.3.

It is evident that the multiclass results are significantly lower than the binary {*asthma*, *no asthma*} classification. The multiclass setting averages performance over 4 classes instead of 2, and these include

Table 2
Classifying binary outcomes v. multiclass outcomes on the asthma data set. Other work on asthma data sets has addressed a binary classification problem {*no asthma*, *asthma*}, while this work addresses the more challenging multiclass classification problem {*no asthma*, *asthma*, *remission*, *relapse*}. Scores are macro-averages unless otherwise noted.

	Precision	Recall	F_1	$\min(P, R)$
Asthma binary	0.8454	0.8565	0.8508	0.8454
Asthma progression†	0.5004	0.4962	0.4976	0.4962

some significant class imbalance. Micro- metrics (not reported here) significantly outperform macro- metrics, since in spite of random undersampling, the amount of training data available is much smaller with the less-frequent *remission* and *relapse* events. Also, the fact that binary performance is similar to that of a similar problem setting [50] is confirmation that the baseline multiclass model is a comparable system starting point.

5.2. Comparing synchronization techniques

Next, we compared sequence classifiers trained on `sync-pass` + `sync-last`, `sync-state` + `sync-last`, and `sync-bin` + `sync-last` data. The `sync-last` in each case makes this a sequence classification task, keeping only the final state both for training and testing. The intention here is to test how the synchronization of input variables affects outcomes. These are commonly utilized, but infrequently studied, preprocessing steps for a sequence classification task.

Table 3
Comparison of `sync-pass`, `sync-bin`, and `sync-state` synchronization strategies. For each listed strategy, a subsequent `sync-last` makes this a classification task. All relative timing information is removed (`elision`), but the `sync-bin` strategy uses a scale of 30 days per bin. The best performance in each column is highlighted in bold.

Synchronization	Precision	Recall	F_1	$\min(P, R)$	AUC
`sync-pass`†	0.5004	0.4962	0.4976	0.4962	0.7182
`sync-state`	**0.5299**	**0.5337**	**0.5314**	**0.5299**	**0.7641**
`sync-bin`	0.5011	0.5139	0.5071	0.5011	0.7352

Table 4
Classification performance of *Markov*, *landmark-own*, and *landmark-any* relative time (Section 3.3), included via concatenation or a relative timing gate (Section 3.5), with different # history context values (Section 3.3). Evaluation metrics include macro-averaged precision (P), recall (R), and F-measure (F); also, area under the precision–recall curve (AUC). The best performance in each column is highlighted in bold.

$\tilde{\tau}$ definition	$\tilde{\tau}$ in DNN	$\#h$	P	R	F_1	$\min(P, R)$
baseline†		0	0.5004	0.4962	0.4976	0.4962
Markov	concat	1	0.5447	0.5630	0.5536	0.5447
Markov	concat	2	0.5073	0.5257	0.5163	0.5073
Markov	concat	3	0.5319	0.5327	0.5319	0.5319
Markov	gate	1	0.5008	0.5097	0.5049	0.5008
Markov	gate	2	0.5217	0.5137	0.5173	0.5137
Markov	gate	3	0.5014	0.5000	0.5004	0.5000
landmark-own	concat	1	**0.5477**	**0.6002**	**0.5726**	**0.5477**
landmark-own	gate	1	0.4934	0.4965	0.4945	0.4934
landmark-any	concat	1	0.5385	0.5569	0.5473	0.5385
landmark-any	concat	2	0.5290	0.5356	0.5321	0.5290
landmark-any	concat	3	0.5283	0.5255	0.5264	0.5255
landmark-any	gate	1	0.4895	0.4954	0.4920	0.4895
landmark-any	gate	2	0.5294	0.5019	0.5148	0.5019
landmark-any	gate	3	0.5188	0.5244	0.5211	0.5188

While most related work performs sampling of time series data, similar to the `sync-bin` strategy, Table 3 shows slightly better performance for `sync-state` (outside of the `sync-pass` 95% confidence interval) on the low-event-density asthma data set.

5.3. Comparing representations of relative time

Next, we considered including adding relative timing information into the model, with different relative timing representations {*Markov*, *landmark*} and means of incorporating into the RNN {elision, concatenation, multiplication} as described in Sections 3.3 and 3.5.

In Table 4, we first note that the Relative Time Gate does not perform as well as Relative Time Concatenation overall. However, whereas concatenating $\#h > 1$ values seems to confuse the model, the gated versions peak at a higher $\#h$ value.

We see that the best performance (by F_1 score, 3rd column) is achieved by concatenating *landmark-own* relative time. Taking into account the inherent variability of this 10-fold cross-validated measurement, 0.5726 ± 0.0350 is still significantly better than the baseline. The other 1-element histories with relative time concatenation also show stronger performance relative to the longer histories.

5.4. Classification on ICU data

We have previously mentioned related work with asynchronous time series data, primarily driven by constant-monitoring critical care settings like ICUs. Here, we evaluate our models on the Physionet Challenge 2012, drawing attention to two additional metrics: the minimum of precision and recall, which is the official metric for Physionet 2012; and the AUC, which is commonly used in the literature on RNNs for time series data. In Table 5, we report results on ICU data that are analagous to both Tables 3 and 4.

We note that the `sync-bin` is a very successful classification baseline with respect to F_1 score, unlike in the asthma setting. However, this does not necessarily hold for AUC. Namely, with `sync-bin` (followed by `sync-last`, to make this a classification problem), the network has learned an appropriate threshold for the Physionet 2012 mortality prediction task; however, it has not necessarily modeled the outcome prediction more generally, since the AUC does not have similar gains.

Interestingly, for the non-binned methods, it is the Relative Time Gate incorporating *landmark-own* relative timing that seems to achieve

Table 5
Incorporating relative time into ICU data. The ICU setting reported consistently lower results for precision than recall, and thus the official Physionet 2012 metric of $\min(P, R)$ is equivalent to the precision P for all rows. The best performance in each column is highlighted in bold.

$\tilde{\tau}$ definition	$\tilde{\tau}$ in DNN	$\#h$	P	R	F_1	AUC
`sync-pass` baseline		0	0.3864	0.4309	0.4066	0.6390
`sync-state` baseline		0	0.3908	0.4649	0.4243	0.6158
`sync-bin` baseline		0	0.4163	**0.5909**	**0.4881**	0.6337
Markov	concat	1	0.3955	0.4650	0.4270	0.6331
Markov	concat	2	0.3955	0.4598	0.4240	0.6445
Markov	concat	3	0.3875	0.4554	0.4182	0.6087
Markov	gate	1	0.3843	0.4391	0.4095	0.6119
Markov	gate	2	0.3938	0.4572	0.4230	0.6384
Markov	gate	3	0.3839	0.4416	0.4102	0.6034
landmark-own	concat	1	0.4106	0.5005	0.4509	0.6201
landmark-own	gate	1	**0.4343**	0.5256	0.4737	**0.6464**
landmark-any	concat	1	0.3771	0.4169	0.3955	0.6266
landmark-any	concat	2	0.3945	0.4760	0.4310	0.6347
landmark-any	concat	3	0.3931	0.4531	0.4204	0.6460
landmark-any	gate	1	0.3870	0.4412	0.4118	0.6237
landmark-any	gate	2	0.3825	0.4383	0.4080	0.5888
landmark-any	gate	3	0.3871	0.4424	0.4122	0.6184

the highest performance by both F_1 (0.4737 ± 0.0302) and AUC (0.6464 ± 0.0389). This is what we would expect, as it is most similar to the more successful settings given in Che et al.: each variable and its duration are treated independently. Other strategies either do not improve on baselines, or do so only marginally.

The original Physionet 2012 competition was judged on $\min(P, R)$, which for Table 5 is the same as the Precision column. Our proposed approach (*landmark-own* with $\tilde{t}$ gate in Table 5) scored $\min(P, R) = 0.4343$. It thus outperformed the baseline algorithm, SAPS-1 [26], which produced $\min(P, R) = 0.3125$; however, it underperformed the top-ranked algorithm in the challenge, which used Bayesian ensemble learning and achieved $\min(P, R) = 0.5353$ [23]. Che et al.'s recent work [6] used the area under the curve (AUC) as its metric, reporting results with their formulation of missing data. For example, they produced $\text{AUC} = 0.7423$ for Logistic Regression with forward imputation, and saw a marked improvement up to $\text{AUC} = 0.8424$ when using their own GRU-D RNN unit [6]. These techniques are optimized for the ICU data in the Physionet Challenge, outperforming our *landmark-own* with $\tilde{t}$ gate.

We will directly compare the evaluations on asthma vs. ICU data in the Discussion section.

6. Discussion

6.1. Asthma sequence classification

The inclusion of relative time has benefits in most of the tests in Tables 2 and 4, showing that, at minimum, there is some signal that is lost in the typical case of timing *elision*. However, it is still quite evident that the included timing information is not nearly robust enough to recreate the rules that generated the gold standard data (e.g., the "no-symptoms-for-3-years = remission" rule).

Indeed, while the *Markov* and *landmark-own* strategies have no representation for this type of rule, we expected the *landmark-any* strategy to be able to learn this type of rule on such a sparse input, but it was not able to do so reliably. Our initial hypotheses about performance drove us to consider class imbalance, training epochs, and other pre-evaluation parameters that we reported in Section 4.3 (namely, these were not truly *pre*-evaluation, but we found insufficient evidence that these were the main areas that contributed to poor performance). It is likely, then, that the *landmark-any* temporal representation does not have a fine-grained enough focus to learn the appropriate rules.

Table 4 shows that there is clearly potential for explicit relative time to help in the task of classification. There is fairly clear indication that including relative time via a concatenative strategy is more beneficial than a gated weighting strategy on asthma data. We hypothesize that the Relative Time Gate places too strong a restriction on the effect of the timing information; it only creates a weighting for the *current* time frame based on the history, rather than also weighting the previous evidence. Future work may include a model that employs the relative time as a secondary, concatenated input to the LSTM Forget Gate.

Another observation from Table 4 was that the concatenative models with $\#h = 1$ were the most successful. While we might surmise that the neural network is simply able to handle a single input better than the multiple inputs of $\#h \geq 2$, we should note that, in our case, *landmark-own* with $\#h = 1$ actually entails a 60-element vector of relative timings. Thus, we hypothesize that the data itself finds a strong signal of relative time in $\#h = 1$, but that signal may become more noisy for larger $\#h$. We may consider alternative neural network structures that more explicitly encode which features' relative timings are important.

6.2. ICU mortality prediction v. asthma status

Comparing Tables 3–5, we see that `sync-pass` and `sync-state` had higher F_1 scores in asthma data, but `sync-bin` performed better in ICU data. This demonstrates how the typical `sync-bin` assumption used in time-series data can lead to poorer performance. As we saw in Section 4.2, the event density in asthma data is much more variable than in the ICU data; we hypothesize that this "burstiness" in asthma input variables is what makes `sync-bin` less successful.

Further, if we compare Table 4 with Table 5, we see that *landmark-own* relative time is effective in both cases. However, incorporating $\tilde{t}$ via concatenation is better with asthma data, whereas the Relative Time Gate performs better with ICU data. With ICU input variables, the mechanisms of homeostasis would suggest that after observing a variable, the values decay towards some default value; thus, the decay rate is captured to some degree in the Relative Time Gate. In contrast, with asthma input variables, we cannot assume homeostasis or default values to be in play; then, the large variability in time between events is itself important information that must be handled separately.

These observations comparing Table 5 with the asthma setting lend some support to our assertion that there is a need for the event sequence representations we have introduced. Although the systems we evaluated have not yet been engineered for excellent performance, we can say from the foregoing results that an ICU-motivated handling of time-series data may not perform optimally in a chronic disease setting. Thus, pragmatically, what model to use to get the best performance is still an open question that depends on what kind of data is being tested.

6.3. Limitations

Our work is preliminary and had a number of limitations on the scope and content, from both an engineering perspective and from a clinical standpoint. First, our evaluation did not optimize neural networks for the best performance; for example, the Physionet 2012 data had real-valued inputs that were lossily binarized so that the models across datasets were essentially the same, and we could make the comparisons above. Furthermore, we did not tackle the difficult problem of determining the granularity of time representation for different "event densities," as introduced in Section 4.2, but only replaced the unit of days with the unit of hours.

In terms of scope, our evaluation only considered the problem setting of classification; future work will tackle the labeling task or asynchronous prediction/generation. From a practical perspective, we aim to compare these strategies to typical neural network methods that address the setting of asynchronously distributed of evidence, such as convolutional, max-pooling, or locally connected layers. These are left to future work. We may also consider modeling event sequence probabilities with temporal aggregation or different statistical distributions (e.g., negative binomial or Poisson process), and comparison with other temporal abstraction techniques.

Clinically, different types of events are not equally valuable for predicting or classifying asthma prognosis. For example, asthma exacerbation carries more weight than poorly defined shortness of breath during or after exercise. While this is partially accounted for in the learned weights of the neural networks, the model does not seem to learn a human-like weighting (or does not have an appropriate set of features to weight). Beyond this, the relative importance of features changes over the course of a child's life. For example, wheezing during early childhood (at<3years of age) carries less weight than wheezing episodes after a later age (>6years of age). These clinical valuations are not modeled directly in either the event sequence preprocessing or in the neural network approach we have presented.

7. Conclusion

We have defined *event sequences* and their associated properties of synchronicity, evenness, and co-cardinality, then seen how explicit timing information can be incorporated into neural network-based models. We found that the inclusion of relative timing can meaningfully improve performance, especially when concatenating one history element of relative time; but, it is also a noisy signal, and further work needs to be done to engineer systems using this paradigm.

Conflict of interest

None.

Acknowledgments

This study was supported by NIH grants R21AI116839 and R01LM011934.

Appendix A. Pre-evaluation experimentation parameters

This Appendix expands on the preliminary tests by which we set the pre-evaluation parameters of Section 4.3.

A.1. Epoch learning curve

Fig. 6 shows the macro-F1 metric as training epochs progress. After an initial period of increased performance, the marginal gain is insignificant.

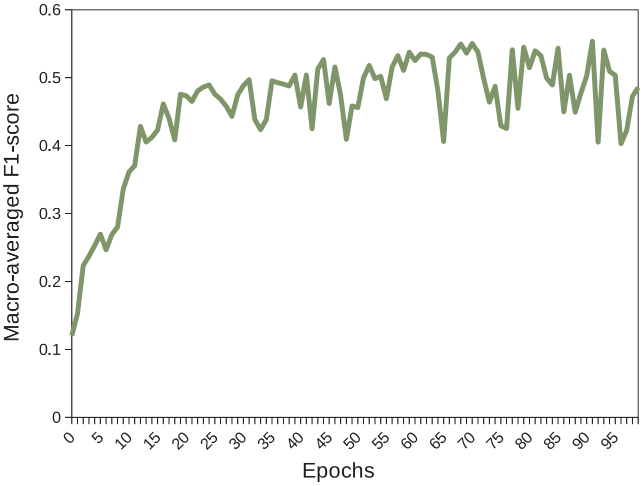

Fig. 6. The learning curve for a single run of a baseline model (single forward LSTM with 32 units, random undersampling, and binary inputs on the asthma data).

A.2. Real-valued Physionet features

In comparison with Tables 5 and 6 results are uniformly lower. This counterintuitive result shows that the lossy binarization procedure that we applied to the ICU data (to correspond with the test on asthma data) did not unnecessarily hurt performance.

Table 6
ICU results (Physionet) with real-valued data. The best performance in each column is highlighted in bold.

$\tilde{t}$ definition	$\tilde{t}$ in DNN	#h	P	R	F1	AUC
Markov	concat	1	0.3287	0.3537	0.3399	0.5693
Markov	concat	2	0.3259	0.3610	0.3416	0.5880
Markov	concat	3	0.3086	0.3318	0.3186	0.5745
Markov	concat	4	0.3433	0.3811	0.3595	0.5910
Markov	gate	1	0.3150	0.3252	0.3188	0.5757
Markov	gate	2	0.3458	0.3794	0.3614	0.6029
Markov	gate	3	0.3439	0.3915	0.3654	0.6084
Markov	gate	4	0.3337	0.3808	0.3554	0.5883
landmark-own	concat	1	0.3249	0.3680	0.3446	0.5871
landmark-own	gate	1	0.3524	**0.4370**	**0.3896**	0.5894
landmark-any	concat	1	0.3389	0.3628	0.3496	0.5687
landmark-any	concat	2	0.3131	0.3372	0.3233	0.5717
landmark-any	concat	3	0.3392	0.3697	0.353	0.5886
landmark-any	concat	4	0.3185	0.3493	0.3323	0.5786
landmark-any	gate	1	0.3276	0.3578	0.3418	0.6040
landmark-any	gate	2	0.3346	0.3690	0.3506	**0.6115**
landmark-any	gate	3	**0.3569**	0.3914	0.3729	0.5911
landmark-any	gate	4	0.3340	0.3653	0.3481	0.5681

References

[1] J.L. Amaral, A.J. Lopes, J. Veiga, A.C. Faria, P.L. Melo, High-accuracy detection of airway obstruction in asthma using machine learning algorithms and forced oscillation measurements, Comput. Methods Programs Biomed. 144 (2017) 113–125.

[2] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, Deep Speech 2: End-to-end Speech Recognition in English and Mandarin, 2015. Available from: arXiv preprint < arXiv:1512.02595 > .

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

[4] F.J. Breidt, N. Crato, P. De Lima, The detection and estimation of long memory in stochastic volatility, J. Econometr. 83 (1) (1998) 325–348.

[5] Z. Che, Y. Liu, Deep learning solutions to computational phenotyping in health care, 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2017, pp. 1100–1109.

[6] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent Neural Networks for Multivariate Time Series with Missing Values, 2016. Available from: arXiv preprint < arXiv:1606.01865 > .

[7] Y. Chen, R.J. Carroll, E.R.M. Hinz, A. Shah, A.E. Eyler, J.C. Denny, H. Xu, Applying active learning to high-throughput phenotyping algorithms for electronic health records data, J. Am. Med. Inform. Assoc. 20 (e2) (2013) e253–e259.

[8] R.A. Covar, R. Strunk, R.S. Zeiger, L.A. Wilson, A.H. Liu, S. Weiss, J. Tonascia, J.D. Spahn, S.J. Szefler, C.A.M.P.R. Group, et al., Predictors of remitting, periodic, and persistent childhood asthma, J. Allergy Clin. Immunol. 125 (2) (2010) 359–366.

[9] Z. Cui, W. Chen, Y. Chen, Multi-scale Convolutional Neural Networks for Time Series Classification, 2016. Available from: arXiv preprint < arXiv:1603.06995 > .

[10] A. Dagliati, L. Sacchi, A. Zambelli, V. Tibollo, L. Pavesi, J.H. Holmes, R. Bellazzi, Temporal electronic phenotyping by mining careflows of breast cancer patients, J. Biomed. Inform. 66 (2017) 136–147.

[11] P. Dagum, A. Galper, E. Horvitz, Dynamic network models for forecasting, Proceedings of the Eighth International Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc, 1992, pp. 41–48.

[12] P. Dagum, A. Galper, E.J. Horvitz, Temporal Probabilistic Reasoning: Dynamic Network Models for Forecasting, Knowledge Systems Laboratory, Medical Computer Science, Stanford University, 1991.

[13] T. Dean, K. Kanazawa, A model for reasoning about persistence and causation, Comput. Intell. 5 (2) (1989) 142–150.

[14] Y. Ephraim, N. Merhav, Hidden markov processes, IEEE Trans. Inform. Theory 48 (6) (2002) 1518–1569.

[15] S. Fernndez, A. Graves, J. Schmidhuber, An application of recurrent neural networks to discriminative keyword spotting, Artif. Neural Networks-ICANN 2007 (2007) 220–229.

[16] J. Finkelstein, A. Gangopadhyay, Using machine learning to predict asthma exacerbations, AMIA. Annual Symposium Proceedings/AMIA Symposium. AMIA Symposium, 2007, p. 955.

[17] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2013, pp. 6645–6649.

[18] D. Heckerman, D. Geiger, D.M. Chickering, Learning bayesian networks: the combination of knowledge and statistical data, Mach. Learn. 20 (3) (1995) 197–243.

[19] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[20] R. Howard, M. Rattray, M. Prosperi, A. Custovic, Distinguishing asthma phenotypes using machine learning approaches, Curr. Allergy Asthma Rep. 15 (7) (2015) 1–10.

[21] G. Hripcsak, D.J. Albers, A. Perotte, Parameterizing time in electronic health record studies, J. Am. Med. Inform. Assoc. (2015) 794–804.

[22] A. Javed, K.H. Yoo, K. Agarwal, R.M. Jacobson, X. Li, Y.J. Juhn, Characteristics of children with asthma who achieved remission of asthma, J. Asthma 50 (5) (2013) 472–479.

[23] A.E. Johnson, N. Dunkley, L. Mayaud, A. Tsanas, A.A. Kramer, G.D. Clifford, Patient specific predictions in the intensive care unit using a bayesian ensemble, Computing in Cardiology (CinC), 2012, IEEE, 2012, pp. 249–252.

[24] A.E. Johnson, T.J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, Mimic-iii, a freely accessible critical care database, Sci. Data 3 (2016).

[25] M.J. Johnson, A.S. Willsky, Bayesian nonparametric hidden semi-markov models, J. Mach. Learn. Res. 14 (Feb) (2013) 673–701.

[26] J.-R. Le Gall, P. Loirat, A. Alperovitch, P. Glaser, C. Granthil, D. Mathieu, P. Mercier, R. Thomas, D. Villers, A simplified acute physiology score for icu patients, Crit. Care Med. 12 (11) (1984) 975–977.

[27] Z.C. Lipton, D. Kale, R. Wetzel, Directly modeling missing data in sequences with rnns: improved classification of clinical time series, Machine Learning for Healthcare Conference, 2016, pp. 253–270.

[28] J. Liu, K. Zhao, B. Kusy, J.-r. Wen, R. Jurdak, Temporal Embedding in Convolutional Neural Networks for Robust Learning of Abstract Snippets, 2015. Available from: arXiv preprint < arXiv:1502.05113 > .

[29] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, J. Gao, Dipole: diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks, Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 1903–1911.

[30] V. Mihajlovic, M. Petkovic, Dynamic Bayesian Networks: A State of the Art, University of Twente Document Repository, 2001.

[31] W.J. Morgan, D.A. Stern, D.L. Sherrill, S. Guerra, C.J. Holberg, T.W. Guilbert, L.M. Taussig, A.L. Wright, F.D. Martinez, Outcome of asthma and wheezing in the first 6 years of life: follow-up through adolescence, Am. J. Respirat. Crit. Care Med. 172 (10) (2005) 1253–1258.

[32] R. Moskovitch, Y. Shahar, Medical temporal-knowledge discovery via temporal abstraction, AMIA Annual Symposium Proceedings, 2009.

[33] R. Moskovitch, Y. Shahar, Classification-driven temporal discretization of multivariate time series, Data Min. Knowl. Disc. 29 (4) (2015) 871–913.

[34] K.P. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning (Thesis), University of California, Berkeley, 2002.

[35] K.P. Murphy, Hidden Semi-Markov Models (hsmms). Unpublished Notes 2, 2002b.

[36] T.D. Nielsen, F.V. Jensen, Bayesian Networks and Decision Graphs, Springer Science & Business Media, 2009.

[37] K. Orphanou, A. Dagliati, L. Sacchi, A. Stassopoulou, E. Keravnou, R. Bellazzi, Incorporating repeating temporal association rules in naïve bayes classifiers for coronary heart disease diagnosis, J. Biomed. Inform. (2018).

[38] T. Pham, T. Tran, D. Phung, S. Venkatesh, Predicting healthcare trajectories from medical records: a deep learning approach, J. Biomed. Inform. 69 (2017) 218–229.

[39] B. Prasad, P.K. Prasad, Y. Sagar, A comparative study of machine learning algorithms as expert systems in medical diagnosis (asthma), Adv. Comput. Sci. Inform. Technol. (2011) 570–576.

[40] M.C. Prosperi, S. Marinho, A. Simpson, A. Custovic, I.E. Buchan, Predicting phenotypes of asthma and eczema with machine learning, BMC Med. Genom. 7 (1) (2014) S7.

[41] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.

[42] M. Ramati, Y. Shahar, Irregular-time bayesian networks, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2010, pp. 484–491.

[43] S. Rea, J. Pathak, G. Savova, T.A. Oniki, L. Westberg, C.E. Beebe, C. Tao, C.G. Parker, P.J. Haug, S.M. Huff, Building a robust, scalable and standards-driven infrastructure for secondary use of ehr data: the sharpn project, J. Biomed. Inform. 45 (4) (2012) 763–771.

[44] S. Saria, A.K. Rajani, J. Gould, D. Koller, A.A. Penn, Integration of early physiological responses predicts later illness severity in preterm infants, Sci. Transl. Med. 2 (48) (2010) 48ra65–48ra65.

[45] J. Schmidhuber, D. Wierstra, F. Gomez, Evolino: hybrid neuroevolution/optimal linear search for sequence learning, Proceedings of the 19th International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc, 2005, pp. 853–858.

[46] Y. Shahar, A framework for knowledge-based temporal abstraction, Artif. Intell. 90 (1–2) (1997) 79–133.

[47] I. Silva, G. Moody, D.J. Scott, L.A. Celi, R.G. Mark, Predicting in-hospital mortality of icu patients: the physionet/computing in cardiology challenge 2012, Computing in Cardiology (CinC), 2012, IEEE, 2012, pp. 245–248.

[48] S. Sohn, C.I. Wi, S.T. Wu, H. Liu, E. Ryu, E. Krusemark, A. Seabright, G.A. Voge, Y.J. Juhn, Ascertainment of asthma prognosis using natural language processing from electronic medical records, J. Allergy Clin. Immunol. (2018) pii: S0091-6749(18)30218-5, 2018..

[49] M. van der Heijden, P.J. Lucas, B. Lijnse, Y.F. Heijdra, T.R. Schermer, An autonomous mobile system for the management of copd, J. Biomed. Inform. 46 (3) (2013) 458–469.

[50] S.T. Wu, Y.J. Juhn, S. Sohn, H. Liu, Patient-level temporal aggregation for text-based asthma status ascertainment, J. Am. Med. Inform. Assoc. 21 (5) (2014) 876–884.

[51] S.T. Wu, S. Sohn, K. Ravikumar, K. Wagholikar, S.R. Jonnalagadda, H. Liu, Y.J. Juhn, Automated chart review for asthma cohort identification using natural language processing: an exploratory study, Ann. Allergy, Asthma Immunol. 111 (5) (2013) 364–369.

[52] M. Xu, K.G. Tantisira, A. Wu, A.A. Litonjua, J.-h. Chu, B.E. Himes, A. Damask, S.T. Weiss, Genome wide association study to predict severe asthma exacerbations in children using random forests classifiers, BMC Med. Genet. 12 (1) (2011) 90.

[53] S. Yi, J. Ju, M.-K. Yoon, J. Choi, Grouped Convolutional Neural Networks for Multivariate Time Series, 2017. Available from: arXiv preprint < arXiv:1703.09938 > .

[54] S.-Z. Yu, Hidden semi-markov models, Artif. Intell. 174 (2) (2010) 215–243.

[55] J. Zhao, P. Papapetrou, L. Asker, H. Bostrm, Learning from heterogeneous temporal data in electronic health records, J. Biomed. Inform. 65 (2017) 105–119.