Clinical Event Time-series Modeling with Periodic Events

Jeong Min Lee and **Milos Hauskrecht**
Department of Computer Science
University of Pittsburgh, Pittsburgh PA 15260 USA
jlee@cs.pitt.edu, milos@pitt.edu

Abstract

The focus of this work is on developing models that can accurately predict events in complex multivariate event-time series derived from electronic health records (EHRs). One common characteristic of many EHR-based event time series is that they are periodic and events are repeated at regular time intervals. Hence in order to define a high accuracy event prediction process, the periodicity of the event occurrence needs to be properly modeled. In this work, instead of trying to combine and model periodic patterns for many event time series in a common hidden space we propose multiple simple periodic mechanisms that help us to drive the expression of individual events in time. We show that these simple periodic mechanisms can be effectively combined with more complex neural architectures capable of modeling the dependencies among different types of events. We test our new model on the clinical event prediction problem that consists of hundreds of lab test events in EHRs derived from MIMIC-III database. We show that our model that relies on simple periodic mechanisms is able to outperform competing baseline models in the multivariate event prediction task.

The objective of this work is to build accurate multivariate event time-series models from Electronic Health Records (EHRs) of past patients that are capable of prospectively predicting future clinical events for a new patient. Our ability to accurately predict such events for a new patient holds a great promise for enhancing patient care and patient management. First, when events are related to the actions of clinicians, the successful prediction of the event can help in assisting clinicians in automatic ordering or reminders of these actions (Hauskrecht et al. 2013). Examples of the actions are orders of medications or lab tests. Second, if events we predict are related to some adverse events, their predictions can prompt the clinician to re-evaluate the patient case and take actions to mitigate the occurrence of such events. Finally, the accurate prediction of events can help us to anticipate the demands for resources that in turn may help through careful optimization plan ahead of time and alleviate resource bottlenecks.

In order to build multivariate event time series models, in this work we explore a variety of architectures for summarizing the history of patient data while at the same time preserving the information needed to accurately predict individual events. One important characteristic of EHR-based event time series is that they are often periodic and events are repeated regularly in time. Examples of such events are the administrations of various medications or regular ordering of laboratory tests. Figure 1 shows the distribution of time gaps between two consecutive clinical events for one of the lab tests with a typical time period of 24 hours. Since periodic or quasi-periodic events are quite frequent in EHRs, in order to define highly accurate event prediction model the periodicity of the events and their occurrences need to be adequately modeled. One approach to incorporate the periodic signals into the event time series models is to use general hidden state space frameworks (like the ones defined by RNNs) and let it figure out (learn) the sufficient statistics driving individual periodic signals from data. However, this approach is not feasible when the number of periodic events we want to model is large, since the hidden state space may get overloaded with the different periodic patterns and their combinations. In this work we take a different approach: we define a simple module for defining periodic events and apply it to enhance the event prediction models. The main advantage of the approach is that it is modular and can be used in combination with other patient history summarization mechanisms.

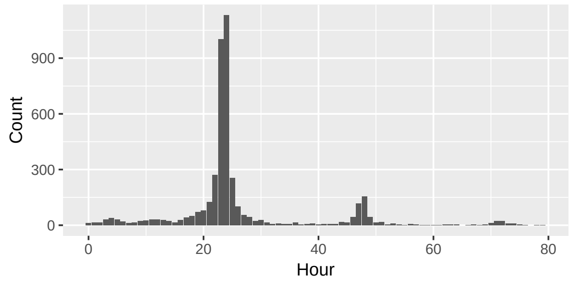

Figure 1: Histogram of time differences for two consecutive events of the cyclosporin lab test event time-series. It illustrates how events in EHRs occur with periodicity.

We experiment with our periodicity-aware module by combining it with the different patient-state summarization architectures. Briefly, recent advances in the temporal neural architectures such as recurrent neural networks (RNN) (Elman 1990) and long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) led to the emergence of new mechanisms for summarizing the past information and dependencies in complex time-series data. These mechanisms were successfully adopted to model sequence and time-series data in many areas. The large adoption, in part, due to its competitive flexibility and performance over traditional approaches which typically require sophisticated feature processing. With large amounts of data, the neural architecture learns to represent the internal representation of input as well as the way to derive the desired output. Its ability to abstract necessary information plays a key role in modeling dependencies between input and output and generating desired output from the input.

We develop a new event time-series framework that combines neural architectures for patient state summarization in event time series with a new simple periodicity-aware mechanism to predict periodic events. Especially, the periodic event modeling mechanism can adaptively utilize event interval statistics computed a priori on the train set with ones from current time-series towards predicting the next event occurrence. We apply our model to the task of predicting hundreds of clinical events on multivariate clinical event time-series data derived from MIMIC-III dataset (Johnson et al. 2016). We show that combined information in our method leads to improved prediction performance compared to neural models singly depend on abstracted information.

Related Work

In this section, we first briefly review the existing approaches to the modeling of periodicity and then we review approaches to general and clinical event time-series data.

Modeling Periodicity

To model periodicity from discrete event time-series, various approaches have been studied. In pattern mining area, many works focus on discovering periodic patterns using various methods such as autocorrelation function (ACF) with Fast Fourier Transformation (FFT) (Berberidis et al. 2002) and chi-squared test (Ma and Hellerstein 2001). However, these methods require sequential data with relatively high sampling rates (Yuan et al. 2017) and as a result, it may not represent well EHR-derived event time-series data where most events are very sparsely occurring.

To model sparsely occurring periodic events, statistical parametric models have been developed, such as hidden semi-Markov model (Kapoor et al. 2015), Poisson mixture model with latent factors (Trouleau et al. 2016), multivariate temporal point processes with Weibull distributions (Kurashima, Althoff, and Leskovec 2018), etc.

Modeling Discrete Event-time Series

The majority of discrete event time-series models are based on Markov processes (McKenzie 2003) and Markov property which assumes that the next state depends only on the most recent state, and is independent of the past states. Standard Markov processes models assume all states of the time series are directly observed. However, the states of many real-world processes are not directly observable and Hidden Markov models (HMM) solve the problem by introducing hidden states.

HMM has been shown to reach good performance in various areas such as stock market prediction (Hassan and Nath 2005) and DNA sequence modeling (Hughey and Krogh 1996). However, the classic HMM model has drawbacks that the discrete hidden state space can restrict the transition of the model when applied to complex real-world time series. Moreover, the dimensionality of the hidden space is not known a priori, which needs to model the long term dependencies and prevent overfitting. By defining real-valued hidden state-space, Linear Dynamical systems (LDS) (Kalman 1963) and time-series models based on it (Liu and Hauskrecht 2016) resolve some of these limitations.

With end-to-end learning framework, neural temporal architectures (Elman 1990; Hochreiter and Schmidhuber 1997; Sutskever, Vinyals, and Le 2014) enable more flexible time-series modeling. Especially LSTM allows modeling long term dependencies in event time-series and they have been successfully applied to many areas such as vision (Gregor et al. 2015), speech (Graves and Jaitly 2014), and language (Sutskever, Vinyals, and Le 2014).

In clinical domain, early works focused on representing patient state based on predefined temporal templates (featurization) and used classifiers (e.g., SVM) to conduct subsequent tasks such as predicting the next events (Valko and Hauskrecht 2010) or outlier detection (Hauskrecht et al. 2016). More recent works focused on combining learning of patient state representation and prediction future state with probabilistic state-space models such as LDS and Gaussian Processes (Liu and Hauskrecht 2016; 2015). Latent state-space allows more flexible and expressive modeling of patient states. In most recent years, RNN, LSTM, and low-dimensional embedding methods (Word2Vec) are actively used to various clinical tasks such as prediction of diagnosis (Malakouti and Hauskrecht 2019a; 2019b) and future clinical events (Choi et al. 2016; Lee and Hauskrecht 2019).

Methodology

In this work, we develop a multivariate discrete event time-series prediction framework that models periodicity of multivariate event streams and utilizes it towards the prediction task. Our framework processes multivariate event streams with two channels that model different aspect of clinical event time-series: one channel is periodicity prediction module that learns a periodicity of each event from event sequence of current individual patient's sequence and another channel is LSTM-based neural prediction module that learns to model longer-term dependencies of events. To compensate cold-start problem on gathering periodic statistics from newly observed patient data, we compile prior probability distribution of event time-gaps from train set. The prior time-gap statistics are then combined with patient-specific event stream periodicity statistics towards predicting the next event occurrence. In the following, we first formalize the prediction problem and then we introduce the periodicity prediction module and neural event prediction module. Finally, we describe the final prediction generation by combining the output of the two modules.

The Prediction Problem

The focus on this work is to predict the next occurrences of multivariate clinical time-series for a patient given his/her clinical event history during hospitalization. Particularly, we focus on predicting the occurrence of lab test events in the next time-window given the history of the events types of medication administration, lab test, procedure, and physiological signals. For the timings of events, we use discrete-time representation. That is, clinical event occurrences of a patient are discretized into a fixed-sized time-window (e.g., 12 hours) and all events occurred during each time-window are represented as a multi-hot vector size of the cardinality of all event types $|E|$. Hence, all clinical events of a patient during hospitalization are represented as a sequence of multi-hot vectors.

Let a binary vector $x_t \in \{0,1\}^{|E|}$ represents occurrences of events of a patient at t-th time window. For the multivariate target events, we have another binary vector $y_t \in \{0,1\}^{|E'|}$ where $|E'|$ is the number of lab test events. Given past T input vector sequences $(x_1, \ldots, x_t)$, our objective is to accurately predict next target vector y_{t+1} that represents clinical events of lab tests that will occur to the patient within the next time window.

Periodicity-based Event Time-Series Prediction

Our framework models periodicity of individual event stream and utilizes it for predicting the next occurrence of each event in the sequence. In order to extract periodic signals from a sequence of events of type e, we define two statistics and update them each time we observe a new event: *Latest Time Gap* (θ) is a time gap between two recent occurrences of a sequence for event e: $\theta^e_t = \chi^e_t - \chi^e_{t-1}$ where χ^e_t and χ^e_{t-1} are timings of the two most adjacent occurrences closest from the current time t. *Elapsed Time* (ϱ) amounts the time elapsed from the latest occurrence χ^e_t of the event: $\varrho^e_t = t - \chi^e_t$. Based on θ^e_t and ϱ^e_t, we create a periodicity-based predictive signal α^e_t that indicates whether the next event will occur during prediction time window (size: W) defied by ϱ^e_t:

$$\alpha^e_{t+1} = \begin{cases} 1 & \text{if } \varrho^e_t < \theta^e_t < \varrho^e_t + W \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Prior Gap Distribution One issue with the aforementioned approach is that it cannot make a prediction until it observes the first two occurrences (χ_1 and χ_2). Another is that when the recent time gap statistic on the current patient event sequence differs from the typical gap (e.g., due to randomness in data collection), the model can output an incorrect α signal. To compensate for these issues, we propose to compile and use the time gap statistic from the complete train set.

Specifically, we define the prior probability distribution (Ψ_e) of time gaps for event type e as follows: we first create a histogram of time gaps by counting the number of occurrences (Y-axis) according to each time-gap (X-axis) from event streams of e across all patients in the train set, as shown in Figure 1. Then, we normalize each count in the histogram to turn the histogram into a probability distribution. Given a specific time-gap τ, Ψ_e can be considered as a function that outputs a probability of observing an event at next time gap τ by returning a value at a bin corresponding to τ in the normalized histogram: $Pr(\tau) = \Psi_e(\tau)$.

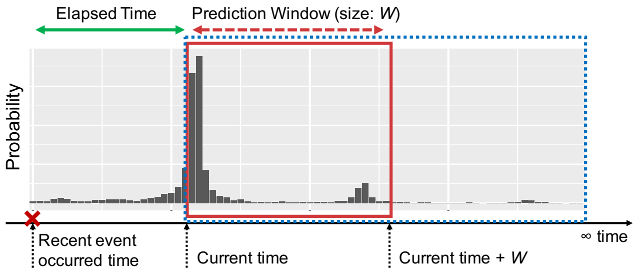

Figure 2: The inter-event gap histogram in Figure 1 is placed on the time axis for prediction. Left edge of the histogram is fitted at the timing of the recent event occurrence. The prior time-gap-based prediction signal β is computed as a probability mass ψ_e^{window} in prediction window (red square) over a probability mass ψ_e^{onward} from current time to onward future (blue dotted square).

Using compiled prior probabilities in Ψ_e, the prior-based predictive signal β^e_{t+1} is computed as a ratio of two probability masses:

$$\beta^e_{t+1} = \psi_e^{\text{window}} / \psi_e^{\text{onward}} \tag{2}$$

where the probability masses are computed as a sum of probabilities in an area defined by specific time ranges shown in Figure 2:

$$\psi_e^{\text{window}} = \sum_{\tau=t}^{t+W} \Psi_e(\tau) \quad \psi_e^{\text{onward}} = \sum_{\tau=t}^{t+\infty} \Psi_e(\tau) \tag{3}$$

Neural Event Prediction

To capture the predictive signals from past history, we use LSTM-based neural architectures. In a nutshell, we first project events observed in a time-window t into a lower-dimensional embedded space and pool event embeddings into a single real-valued vector $\tilde{x}_t$. Then, we use LSTM to abstract information on the time-window into hidden state spaces h_t with encoded information from previous steps in h_{t-1}. In the following, we describe the details of our approach.

Event Embedding and Pooling. To generate LSTM inputs we rely on the embedding matrix $W_{emb} \in \mathbb{R}^{d_e \times |E|}$. Briefly, we first represent all input events that occur in a time-window t to a multi-hot x_t and then project this vector into a lower dimensional representation $\tilde{x}_t$:

$$\tilde{x}_t = W_{emb} \cdot x_t \tag{4}$$

where d_e is the dimension of the embedding. This process amounts to aggregate individual event embeddings in a time-window into a single vector using sum pooling.

LSTM-based Sequence Process. With the help of gating mechanisms and hidden states, LSTM models have been successfully used to model time series with various dependencies including those with longer horizons. At a glance, at each time step t, given input $\tilde{x}_t$ and the previous hidden state h_{t-1}, LSTM updates the hidden state h_t and cell states C_t:

$$h_t, C_t = \text{LSTM}(\tilde{x}_t, h_{t-1}, C_{t-1}) \tag{5}$$

For details of the LSTM parameterization, please refer to (Hochreiter and Schmidhuber 1997).

Predicting Next Events

The prediction of events in the next window ($\hat{y}_{t+1}$) is computed by combining predictive signals $\alpha^e_{t+1}, \beta^e_{t+1}$ from periodicity module and h_t from LSTM. Specifically, for each event e, we first combine these signals into a vector γ by concatenation:

$$\gamma^e = [h_{t+1}; \alpha^e_{t+1}; \beta^e_{t+1}] \tag{6}$$

Then, γ is projected to a real-valued scalar z_e through a linear transformation with a vector $W^e \in \mathbb{R}^{1 \times d_h + 2}$ and a bias $b^e \in \mathbb{R}$. We apply the same procedure to all target events $e \in E'$ and compose a vector z:

$$\begin{aligned} z^e &= W^e \cdot \gamma^e + b^e \\ z &= [z^1; \dots ; z^{|E'|}] \end{aligned} \tag{7}$$

We compute the final prediction with a logistic sigmoid function $\sigma(x) = 1/(1 + \exp(-x))$:

$$\hat{y}_{t+1} = \sigma(z + b_\kappa) \tag{8}$$

where b_κ is a recent-bias term (Lee and Hauskrecht 2019) which brings information about recently occurred events. It is known that the bias can improve the quality of EHR-based event sequence prediction. Briefly, b_κ is computed as a linear projection of recently occurred events to a target event space with a matrix $W_r \in \mathbb{R}^{|E| \times |E'|}$ and a bias term b_r:

$$b_\kappa = W_r \cdot x_t + b_r \tag{9}$$

Parameter Learning

The parameters of the model are learned by stochastic gradient descent based adaptive optimizer (Adam). For loss function, we use the binary cross-entropy between true label vector y_{t+1} and prediction vector $\hat{y}_{t+1}$ over all sequences in train set.

Experiments

In this section, we provide the details experimental evaluation of our new event prediction model and its comparison to multiple baseline models. We start by describing the data preprocessing steps.

Data and Preprocessing

Data Source and Cohort Selection. We use MIMIC-III dataset to evaluate the performances of our model. It is publicly available and contains real-world EHRs of intensive care unit patients. We extract 5137 patients from the database with following criteria: (1) adult, whose age between 19 and 99 (2) length of stay is between 48 and 480 hours (3) whose clinical records are in the Meta Vision, one of EHR systems that generated patient records for MIMIC-III dataset. Except for these criteria, we do not filter out any patient in order to test our model across the general patient pool regardless of disease, symptoms, or conditions. We randomly split 5137 patients into training and test sets with a ratio of 8:2.

Feature Preparation. Then we generate multivariate discrete event time-series by segmenting all sequences with three window sizes ($W = 6, 12, 24$ hours). At each step of a window segment, a multi-hot vector input $x_t \in (0,1)^{|E|}$ is formed by aggregating all types of events occurred in the window. Similarly, the prediction target $y_{t+1} \in (0,1)^{|E'|}$ is formed as a multi-hot vector of lab test events occurred in the next window segment. For the category of events in the input window, we use four clinical event types: medication administration, lab results, procedure, and physiological result. For medication, lab, and procedure event categories, we filter out those events observed in less than 500 different patients. Further, for each of 10 splits, we filter out those events that not observed in both train and test sets. As a result, we obtain 64 medication events, 44 procedure events, 155 lab test events, and 19 physiological signal events. They correspond to ($|E|$=) 282 input events x_t and ($|E'|$=) 155 target events y_t for multivariate discrete event prediction task.

Baseline Models

We compare our model with baseline models that can also predict events in multivariate vector given history sequence:

- **State-space Markov Model (Markov)** defines next target event occurrence y_{t+1} as a Markov transition of current event occurrence x_t. We parameterize the transition by a linear transformation with sigmoid output function: $\hat{y}_{t+1} = \sigma(W_s \cdot x_t + b_s), W_s \in \mathbb{R}^{|E'| \times |E|}$
- **Logistic Regression with Binary Input (LR-BN)**: As the Markov model only utilizes the current information, it cannot make use of information from the past. To solve the problem, we aggregate all event occurrences in a history sequence and represent it as an indicator vector. Then, contents in the vector are projected to the prediction by using the same parameterization of the Markov model.
- **LSTM**: We take hidden states h_t and project it to event space to make a prediction: $\hat{y}_{t+1} = \sigma(W_l \cdot h_t + b_l), W_l \in \mathbb{R}^{d_h \times |E'|}$. As LSTM processes sequences with hidden states, it can utilize information from distant past.
- **LSTM with Recent Bias (LSTM-RB)**: As shown in (Lee and Hauskrecht 2019), when the next event prediction from LSTM is shifted with recent bias from the projected

Table 1: Overall prediction results (AUPRC) for different window segmentation settings. The metrics are averaged for 10 different random subsampling splits.

	W=6	W=12	W=24
LR-BN	0.1511	0.2194	0.3000
Markov	0.1749	0.2658	0.3324
LSTM	0.2427	0.2797	0.3161
LSTM-RB	0.2512	0.2928	0.3367
Proposed	0.2593	0.2981	0.3378

recent event occurrence x_t, the predictability is significantly increased. The recent bias vector b_κ in Equation 9 is added to hidden states h_t of the baseline LSTM: $\hat{y}_{t+1} = \sigma(W_l \cdot h_t + b_l + b_\kappa)$.

Evaluation Metrics

The area under the precision-recall curve (AUPRC) is used to evaluate the quality of predictions of the models. Under a highly imbalanced dataset, AUPRC is known for presenting a more accurate profile on measuring performances of models (Saito and Rehmsmeier 2015). Due to the nature of EHR-derived time-series data, many events occur sparsely among all possible time windows. For example, the rate of occurrence among all possible time window is low such as 5% for medication category, 7% for procedure category, and 12% for lab test category.

In order to conduct the experiments in a robust way, we obtain 10 different train-test sets from the random shuffling of patients. The reported AUPRC values (for the different methods) are averaged over all target events and over test sets for 10 different train/test splits.

Implementation Details

For the experiments, we use fixed embedding size $d_e = 64$, learning rate $= 0.005$, and minibatch size $= 128$. To prevent over-fitting, L_2 weight decay regularization is applied to all models including baselines. Size of hidden state $d_h = (64, 128, 256, 512)$ and the regularization weight are determined by 5-fold internal cross validation.

Experiment Results

Table 1 summarizes the performance of proposed model and baseline models for lab test results events prediction ($|E'| = 155$) on different window segmentation settings. The results show that our model outperforms all baselines. More specifically, compared to LSTM-RB, the best-performing baseline model, our model shows 5.4% improvement for 6-hour window, 3.2% and 1.8% improvements for 6 and 12 hour windows respectively. Compared to averaged AUPRC of all baseline models, our model shows 26%, 12%, and 5% improvements for 6, 12, and 24 hour windows. The overall performance across all models is higher with a larger window segmentation setting. This is mainly due to a higher prior for events for a longer prediction window.

To confirm the benefits of the periodicity signals α, β, we break down the prediction results based on the number of previous occurrences. Concretely, for each event type, we divide all occurrences into three groups: **First occur** group includes cases without any previous occurrence (from the beginning till the first occurrence). **Second occur** group includes cases with only one previous occurrence (after the first occurrence until the second occurrence). **Later occur** group includes cases with two and more previous occurrences (after the second occurrence till the last occurrence). AUPRC is first computed separately for each event in each group and then averaged across events for the report. Since the periodicity signals cannot be generated before observing the first occurrence, there should be a minimal performance gap between our model and LSTM-RB in the first occur group. As β is generated after the first occurrence and α is generated after observing the first two occurrences, we expect to see some performance gap in the second occur group and more in the later occur group. As shown in Figure 3, the results reflect the expectation. For example, in W=6 setting, our model performs 1.21% better than LSTM-RB in the first occur group. In the second and later occur groups, the gap is increased to 3.3% and 3.4% respectively. For W=12 setting, our model performs 1.4% and 2.2% better than LSTM-RB in the second and later groups but it performs 1.6% worse in the first occur group.

Conclusion

We have proposed a novel clinical time-series model that aims to accurately predict events in complex multivariate event-time series with periodic event patterns. To do it, we develop a simple periodic mechanism to drive the expression of individual events in time. We show that this mechanism (when applied to many different events) can be effectively combined with more complex neural architectures capable of modeling the dependencies among different types of events. We test our new model on the clinical event prediction problem that consists of hundreds of lab test events in EHRs derived from MIMIC-III database. We show that our model that relies on simple periodic mechanisms is able to outperform more general state-space frameworks.

References

Berberidis, C.; Aref, W. G.; Atallah, M.; Vlahavas, I.; Elmagarmid, A. K.; et al. 2002. Multiple and partial periodicity mining in time series databases. In *ECAI*, volume 2, 370–374.

Choi, E.; Bahadori, M. T.; Sun, J.; Kulas, J.; Schuetz, A.; and Stewart, W. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, 3504–3512.

Elman, J. L. 1990. Finding structure in time. *Cognitive science* 14(2):179–211.

Graves, A., and Jaitly, N. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, 1764–1772.

Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D. J.; and Wierstra, D. 2015. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*.

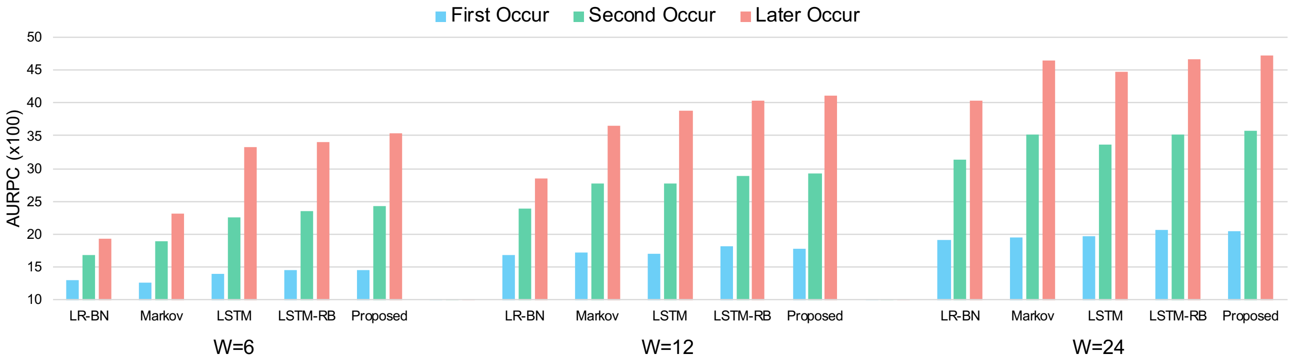

Figure 3: Prediction results decomposed by the number of previously occurred events of the same type.

Hassan, M. R., and Nath, B. 2005. Stock market forecasting using hidden markov model: a new approach. In *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*, 192–196. IEEE.

Hauskrecht, M.; Batal, I.; Valko, M.; Visweswaran, S.; Cooper, G. F.; and Clermont, G. 2013. Outlier detection for patient monitoring and alerting. *Journal of biomedical informatics* 46(1):47–55.

Hauskrecht, M.; Batal, I.; Hong, C.; Nguyen, Q.; Cooper, G. F.; Visweswaran, S.; and Clermont, G. 2016. Outlier-based detection of unusual patient-management actions: an icu study. *Journal of biomedical informatics* 64.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hughey, R., and Krogh, A. 1996. Hidden markov models for sequence analysis: extension and analysis of the basic method. *Bioinformatics* 12(2):95–107.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Li-wei, H. L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data* 3:160035.

Kalman, R. E. 1963. Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control* 1(2):152–192.

Kapoor, K.; Subbian, K.; Srivastava, J.; and Schrater, P. 2015. Just in time recommendations: Modeling the dynamics of boredom in activity streams. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 233–242. ACM.

Kurashima, T.; Althoff, T.; and Leskovec, J. 2018. Modeling interdependent and periodic real-world action sequences. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 803–812.

Lee, J. M., and Hauskrecht, M. 2019. Recent context-aware lstm for clinical event time-series prediction. In *Conference on Artificial Intelligence in Medicine*, 13–23. Springer.

Liu, Z., and Hauskrecht, M. 2015. Clinical time series prediction: Toward a hierarchical dynamical system framework. *Artificial Intelligence in Medicine* 65(1):5–18.

Liu, Z., and Hauskrecht, M. 2016. Learning linear dynamical systems from multivariate time series: A matrix factorization based framework. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, 810–818. SIAM.

Ma, S., and Hellerstein, J. L. 2001. Mining partially periodic event patterns with unknown periods. In *Proceedings 17th International Conference on Data Engineering*, 205–214. IEEE.

Malakouti, S., and Hauskrecht, M. 2019a. Hierarchical adaptive multi-task learning framework for patient diagnoses and diagnostic category classification. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 701–706.

Malakouti, S., and Hauskrecht, M. 2019b. Predicting patient's diagnoses and diagnostic categories from clinical-events in ehr data. In *Conference on Artificial Intelligence in Medicine*, 125–130. Springer.

McKenzie, E. 2003. Ch. 16. discrete variate time series. *Handbook of Statistics* 21:573–606.

Saito, T., and Rehmsmeier, M. 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one* 10(3):e0118432.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Trouleau, W.; Ashkan, A.; Ding, W.; and Eriksson, B. 2016. Just one more: Modeling binge watching behavior. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Valko, M., and Hauskrecht, M. 2010. Feature importance analysis for patient management decisions. *Studies in health technology and informatics* 160(Pt 2):861.

Yuan, Q.; Shang, J.; Cao, X.; Zhang, C.; Geng, X.; and Han, J. 2017. Detecting multiple periods and periodic patterns in event time sequences. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 617–626. ACM.