Point-process based representation learning for Electronic Health Records

Hojjat Karami, *Fellow, IEEE*, Anisoara Ionescu, and David Atienza, *Member, IEEE*

***Abstract*— Irregular sampling of time series in electronic health records (EHRs) is challenging for model development. In addition, the pattern of missingness for certain clinical variables are not at random as it is determined by clinicians decision and the state of patient. Point process is a mathematical framework for handling event sequence data which is also consistent with the irregular sampling. We propose *TEEDAM*, a deep neural network that can learn patient representation from irregulary sample time series as well as informative missingness pattern of certain laboratory variables. We performed various experiments to show the effectiveness of event and state encoding for characterization of conditional intensity functions as well as downstream prediction task. Results show that in some cases learning from patterns may improve the performance of prediction task.**

***Index Terms*— electronic health records (EHRs), Point Process, irregular sampling, informative missingness,**

I. Introduction

healthcare systems are one of main area for deploying machine learning models. Thanks to recent advancements in data collection technologies, policies and computing power data are recorded in a hosptial. This data consists of multiple modalities frome tabular and time series to clinical notes and images. We focus on tabular and time series. [1]

talk about the irregular sampling in EHRs and causes./// In and ideal setup, variables are measured on a regular basis, however, in a hospital due to medical costs, patient associated risks, sampling device.

conseq of irr sampling/// Irregularly sampled time series in electronic health records can have serious consequences for data analysis and decision making. Irregular sampling occurs when the time intervals between data points are inconsistent, which can lead to biased or inaccurate results when trying to make inferences about the underlying process. This can also make it difficult to accurately compare the data over time, as well as to detect patterns or trends. Furthermore, irregular sampling can cause problems for machine learning algorithms, which may struggle to accurately capture the relationships between variables in the data. This can result in poor predictions or incorrect diagnoses, potentially leading to negative impacts on patient care. To mitigate these consequences, it is important to ensure that electronic health records are sampled at regular intervals, or that appropriate statistical techniques are applied to account for the irregularity.

potential adv of irr sampl///

imputation methods/// Imputation techniques are widely used in electronic health records (EHRs) to handle missing data. EHRs often contain incomplete or missing information due to various reasons such as measurement errors, system failures, or data entry issues. Imputation refers to the process of filling in the missing data with plausible values based on the available information. There are several imputation techniques available, each with their own strengths and weaknesses. For example, mean imputation replaces missing values with the mean of the available values in the same column, while regression imputation uses a regression model to estimate the missing values based on the relationship between the variables. Other popular imputation methods include multiple imputation, hot deck imputation, and k-nearest neighbors imputation. The choice of imputation technique depends on the type and amount of missing data, as well as the goals of the analysis. Regardless of the technique used, it is important to carefully consider the impact of imputed values on the analysis and to report the imputation method used in the results.

inherently compatible methods/// Deep learning models that are compatible with irregularly sampled time series are important in many real-world applications, such as stock market predictions, speech recognition, and medical diagnosis. These models need to be able to handle data that is not evenly spaced, as is often the case in time-sensitive applications. One popular deep learning architecture that can handle irregularly sampled time series is the Recurrent Neural Network (RNN). RNNs are well-suited for this task because they have the ability to process sequential data, taking into account not only the current input but also the previous inputs. Another architecture that can handle irregularly sampled time series is the Convolutional Neural Network (CNN), which can be used to extract features from the data and then pass the processed data to an RNN for further analysis. Additionally, Attention Mechanisms can be integrated into RNNs and CNNs to help the model focus on important features in the data. Overall, these deep learning models offer a powerful toolset for handling irregularly sampled time series and provide useful insights into this type of data.

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, “This work was supported by DigiPredict Grant BS123456.”

H. Karami is with the EPFL, Lausanne, Switzerland (e-mail: hojjat.karami@epfl.ch).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).

pp framework/// Point processes are mathematical models used to describe the distribution of events in time or space. These events can be occurrences of earthquakes, nerve impulses, customer purchases, or anything else that can be counted or measured. Neural Point Processes are a type of machine learning models that learn the patterns and relationships between events using artificial neural networks. These models can be used for tasks such as predicting future events, estimating the rate of event occurrence, or identifying correlations between events. They offer a flexible and powerful way to analyze point process data, as they can handle complex dependencies between events and incorporate prior knowledge about the process.

Our aims??/// In this work, we propose a DNN for joint modeling of NTPP and a DAM. Our first aim is to investigate the effectiveness of state encoding for CIF characterization. Second, we are interested in seeing the utility of event encoding a downstream task

contribs?///

II. Background

Temporal point process: Consider an event sequence data $\mathcal{D} = \{\mathcal{S}_i\}_{i=1}^{N}$ where each sample is represented as an event sequence $\mathcal{S}_i = \{(t_i, e_i)\}_{j=1}^{L}$, where L is the total number of events, t_j is event's timestamp, and $e_j \in \mathbb{R}^M$ is the binary represntation of event marks (multi-class or multi-label). The history of events is denoted as $\mathcal{H}_t = \{(t_j, e_j) : t_j < t\}$.

The core idea of the point process framework is the definition of conditional intensity functions (CIFs) which is the probability of the occurence of an event of type m in an infinitesimal time window $[t, t + dt)$:

$$\lambda_m^*(t) = \lim \frac{P(\text{event of type m in } [t, t + \Delta t) | \mathcal{H}_t)}{\Delta t} \tag{1}$$

Multivariate Hawkes process is the traditional approach to characterize CIFs:

$$\lambda_m^*(t) = \mu_m + \sum_{(t', e') \in \mathcal{H}_t} \phi(t - t') \tag{2}$$

Neural temporal point process: Encoder-decoder architectures have proven to be effective in many applications. The main idea of a neural temporal point process (NTPP) is to first encode the history of events until t_j using a neural network architecture $h_j = Enc(\mathcal{H}_{j+1}; \theta)$. Then it tries to estimate $\lambda_m^*(t) = Dec(h_j; \phi)$ for $t \in (t_j, t_{j+1}]$.

Initial works used recurrent neural networks either in a discrete way [16], a continuous way [39], or a multi-scale multi-channel architecture [24]; then self-attention was also used by [54, 57]. [57] develop a Structured Transformer Hawkes (STH) model that studies the relationship between point processes. It assumes the modeling of one point process might contribute to the modeling of another. Each vertex in the STH's graph is associated with a point process. Other efforts have been made to model the cumulative intensity function [42] and conditional probability density [46]. One issue that has been ignored in these works is that these NPP models take all types of historic events to predict the future ones, neglecting that some event types may not contribute to the prediction of another type. This study aims to correct this defect by learning to omit those non-contributing event types when predicting the target type via NPPs.

Parameter Estimation: Based on conditional intensity function, it is straightforward to derive conditional probability density function $p_m^*(t)$ in $(t_j, t_{j+1}]$:

$$p_m^*(t) = \lambda_m^*(t) \exp\left[-\sum_{m=1}^{M} \int_{t_j}^{t_{j+1}} \lambda_m^*(t') dt' \right] \tag{3}$$

Intuitively, the integral term indicates that there is no event of any type in the interval.

In the multi-class setting, the log-likelihood (LL) of a point process for a single event sequnce $\mathcal{S}_i$ is defined as:

$$\begin{aligned} \log p_{mc}(\mathcal{S}_i) &= \sum_{j=1}^{L} \sum_{m=1}^{M} 1(e_j = m) \log p_m^*(t_j) \\ &= \sum_{j=1}^{L} \sum_{m=1}^{M} 1(e_j = m) \log \lambda_m^*(t_j) \\ &\quad \sum_{m=1}^{M} \left(\int_0^T \lambda_m^*(s)\, ds \right) \end{aligned} \tag{4}$$

However, in many cases such as EHRs it is common to have co-occuring events. [ntpp] proposed to add:

$$\begin{aligned} \log p_{ml}(\mathcal{S}_i) &= \log p_{mc}(\mathcal{S}_i) \\ &\quad + \sum_{m=1}^{M} (1 - 1(e_j = m)) \log (1 - p_m^*(t_j)) \end{aligned} \tag{5}$$

It should be noted that the real advantage of point process is modeling non-event likelihoods in the form of integrals. If we neglect the integrals, we would achieve the cross-entropy and binary cross entropy loss in the multi-class and multi-label settings respectively for the prediction of next mark given history of events.

Another approach is the marked case which assumes that marks and timestamps are conditionally independent given $\mathcal{H}_t$.

$$\begin{aligned} \log p_{marked}(\mathcal{S}_i) &= \sum_{j=1}^{L} \sum_{m=1}^{M} 1(e_j = m) \log p^*(e_j = m) \\ &\quad + \sum_{j=1}^{L} \lambda^*(t_j) - \int_{t_0}^{t_L} \lambda^*(t') dt' \end{aligned} \tag{6}$$

Deep learning for irregular sampling: An irregularly sampled data can be denoted as $\mathcal{D} = \{\mathcal{U}_i\}_{i=1}^{N}$ where N is number of samples and each sample is represented as series of tuple $\mathcal{U}_i = (t_p, k_p, v_p)$ where t_p, k_p, v_p represents the time, modality and value of p-th datapoint respectively.

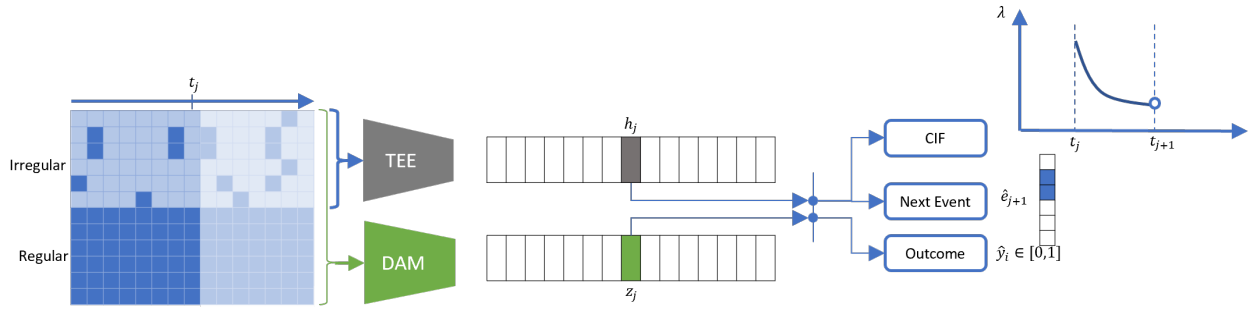

Fig. 1. Magnetization as a function of applied field. It is good practice to explain the significance of the figure in the caption.

III. Related Works

IV. Proposed Model

The key advantage of our proposed model is to combine a transformer-based event encoder (TEE) with a deep attention module (DAM) that can handle an irregularly sampled time series for a any downstream prediciton task. In this case the data is represented as $\mathcal{D} = \{\mathcal{S}_i, \mathcal{U}_i\}_{i=1}^{N}$. General schematic of TEDAM is depicted in FIG.

A. Event Encoder

We use a similar transformer architecture [thp] for encoding events with minor modifications for event embedding.

In the first step, we embed all event marks $E_{emb} = E \times W_{emb}$ where $E_i \in \mathbb{R}^{L \times M}$ is the binary encoding matrix of all event marks (multi-label or multi-class), and $W_{emb} \in \mathbb{R}^{M \times d_{emb}}$ is the trainable embedding matrix.

In the second step, timestamps should be encoded and added to the event embedding, however, we propose to concatenate time encodings that can lead to better characterization of conditional intensity functions. Finally, the input of the transformer encoder will be $X_{emb} = [E_{emb}, T_{emb}] \in \mathbb{R}^{L \times (d_{emb} + t_{emb})}$.

Here, we use the standard transformer encoder similar to [vaswani] with masking matrix to prevent the model from looking into the future. we obtain the encoded matrix $H = (h_1, ..., h_j, ..., h_L)$ where h_j containes the all available information until occurence of j-th event.

B. State Encoder

Similar to [setF], we use an attention-based aggregation approach for encoding all additional information. Each side information (t_k, v_k, m_k) can be represented by $s_k = (z(t_k), v_k, m_k)$. we define attention $a(\mathcal{U}_k, s_k)$

We define $\mathcal{U}_p$ to be the set of the first p available information. The goal is to calculate $a(\mathcal{U}_p, s_k), k \leq p$ that is the relevance of k-th observation s_k to the first p observed values $\mathcal{U}_p$. This is achieved by computing an embedding of the set elements using a smaller set functions f', and projecting the concatenation of the set representation and the individual set element into d-dimensional space:

$$f'(\mathcal{U}_p) = g'\left(\frac{1}{|p|}\sum_{u_k \in \mathcal{U}_p} h'(u_k; \theta'); \rho'\right) \tag{7}$$

Then we can compute key values using key matrix $W^k \in \mathbb{R}^{(d_g + d_s) \times d_{prod}}$

$$K_p = [f'(\mathcal{U}_p), u_p]^T W^K \tag{8}$$

using a query vector $w^q \in \mathbb{R}^{d_{prod}}$

$$a(\mathcal{U}_p, u_p) = softmax(\frac{K_p . w^q}{\sqrt{d}})$$

Finally, we compute a weighted aggregation of set elements:

$$z_p = f(\mathcal{U}_p) = g\left(\sum_{s_k \in \mathcal{U}_p} a(\mathcal{U}_p, s_k) h(s_k; \theta); \rho\right)$$

we regard $z_p \in \mathbb{R}^{d_\rho}$ as the representation of state data until arrival of p-th datapoint.

Finally, we need to combine event embeddings $H_{L \times d_e}$ and state embeddings $Z_{P \times d_\rho}$, however, the length of each does not match. As a result, we consider the reduced version of state matrix as below:

$$Z_i^{red} = Z_p \text{ where } p = argmax(z \leq i)$$

Without loss of generality, we can consider multiple heads by adding an additional dimension to keys and queries.

C. All formulas

D. Event Decoder

Once we obtain a representation of a patient using embedded events and states, we can try to parameterize conditional intensity functions (CIFs) of the events.

In neural point process literature, many approaches have been propose to decode either conditional or cumulative intensity function. We will use a decoder similar to [sahp] as it can model both exciting and inhibiting effects for modeling CIFs.

$$\begin{aligned}
\mu_{m,i+1} &= gelu(h_{i+1} W_{m,\mu} + z_{i+1} W_{m,\mu}), \\
\eta_{m,i+1} &= gelu(h_{i+1} W_{m,\eta} + z_{i+1} W_{m,\eta}), \\
\gamma_{m,i+1} &= gelu(h_{i+1} W_{m,\gamma} + z_{i+1} W_{m,\gamma}),
\end{aligned}$$

Finally, we can express the intensity function as follows:

$$\lambda_m(t) = \text{softplus}(\mu_{m,i+1} + (\eta_{m,i+1} - \mu_{m,i+1}) \exp(-\gamma_{m,i+1}(t - t_i))),$$

for $t \in (t_i, t_{i+1}]$,where the $softplus$ is used to constrain thei intensity function to be positive.

E. Loss Function

We define a multi-objective loss function $\mathcal{L} = \mathcal{L}_{CIF} + \mathcal{L}_{mark} + \mathcal{L}_{state}$.

V. Experiments

We perform various experiments to show the effectiveness of each component in out model.

Datasets

To show the utility of time concatenation and marked loss, we consider three datasets:

Synthea(*Syn*). We used the Synthea simulator (Walonoski et al., 2018) which generates patient-level EHRs using human expert curated Markov processes. Here, we reused the already processed version of this data by [ntpp].

ReTweets (*RT*). The Retweets dataset contains sequences of tweets, where each sequence contains an origin tweet (i.e., some user initiates a tweet), and some follow-up tweets. We record the time and the user tag of each tweet. Further, users are grouped into three categories based on the number of their followers: “small”, “medium”, and “large”

Stackoverflow (*SO*). is a question-answering website. The website rewards users with badges to promote engagement in the community, and the same badge can be rewarded multiple times to the same user. We collect data in a two-year period, and we treat each user’s reward history as a sequence. Each event in the sequence signifies receipt of a particular medal.

Furthermore, we consider two EHRs provided by physionet challenge to investigate the advantage of irregular sample and point process modeling in the same time.

Physionet 2012 Mortality Prediction Challenge (*P12*). The 2012 Physionet challenge dataset (Goldberger et al., 2000),contains 12, 000 ICU stays each of which lasts at least 48 h. For each stay, a set of general descriptors (such as gender or age) are collected at admission time. Depending on the course of the stay and patient status, up to 37 time series variables were measured (e.g. blood pressure, lactate, and respiration rate). While some modalities might be measured in regular time intervals (e.g. hourly or daily), some are only collected when required; moreover, not all variables are available for each stay.

Physionet 2019 Sepsis Early Prediction Challenge (*P19*). This dataset contains clinical data of about 40k patients in ICU. Clinical data consist of demographics, vital signs and laboratory values as well as sepsis label in a one-hour time grid. Our objective is to predict the timestamp of next lab sampling events as well as measured variables (event marks) given the patient history.

Scenarios

To show the utility of time concatenation and marked-shp we report the metrics for SO, RT and SYN and compare it with 3 baselines: SAHP, THP, and GRU-CP.

To show the utility of additional information for CIF modeling, we report NLL/events for P12 and P19 in the following conditions: TE, TE+DAM, TE+noise

Finally, we investigate whether point process characterization can be useful in a downstream task or not. As each dataset consists of different hospitals and the pattern of irregular sampling might differ from hosptial to hospital, we assume different settings: single-center, multi-center and external evaluation.

Baselines

we compare our model against existing models: THP, SAHP, GRU-CP.

Metrics

We report the weighted AUPRC, AUROC of next predicted event as well as root mean sqaure error (RMSE) of next measurement interval. For evaluating the goodness of fit for the parameterized point process, we report normalized negative likelihood normalized by number of ocurred event (NLL/events). Furthermore, we can also evaluate the learned representation of each patient to predict the sepsis label in a binary classification task.

Training Details

To be completed.

VI. Results and Discussion

In this section, we present our results regarding the advantage of state and event encoding.

A. Effect of minor improvements

[TB1] shows the performance metrics of different datasets.

In general, time concatenation has lead to better F1-score/AUROC as well as LL/events compared to adding time. This idea can be used for future research in NTPPs for better characterization of CIFs.

Another interesting fact is that in some datasets (A, B) next mark prediction has competetive performance compared to the point process loss. However, in many cases, point process loss significantly improves the performance metrics which indicates that modeling non-event likelihood could be beneficial. In addition, we propose to use next-mark prediction case as a baseline.

Another interesting results is that independent assumption in the marked loss may lead to worse performance compared to MC/ML loss.

TABLE I
ADD CAPTION

		TEEDAM									
		AE (next mark)		PP(single+mark)		PP (MC/ML)					
Dataset	Metric	concat	sum	concat	sum	concat	sum	Latent	SAHP	THP	GRU-CP
SO	LL/#events	ND	ND	-0.56	-0.57			-1.54	-1.86	-1.84	NR
	F1-score	27.86	28.79	29.10	28.95			28.34(0.19)	24.12	23.89	26
ReTweet (MC)	LL/#events	ND	ND					-3.89	-4.56	-4.57	NR
	F1-score							58.29	53.92	53.86	NR
Synthea	LL/#events	ND	ND					ND	ND	ND	NR
	AUROC							ND	ND	ND	0.85(.014)
ReTweet (ML)	LL/#events	ND	ND	1.86	1.4			ND	ND	ND	NR
	AUROC	63.3	61.01	68.7	66.2	73.87		ND	ND	ND	0.611(0.001)

B. Negative Likelihood with state encoding

[TB2] shows the result for estimation of negative likelihood for the two datasets.

In general we can see that using state information results in lower NLL, however, it cannot neccessarily increase the AUROC for next time prediction.

Another observation is that more data will lead to higher LL.

TABLE II
ADD CAPTION

		Model		
Dataset	setting	TE	TE+DAM	TE+noise
P12	sc	-1.99	-1.77	-1.92
	mc1	-0.99	-1.14	-0.89
	mc2	-1.5	-1.33	-1.47
P19	sc	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
	mc1	-1.81	-1.62	-1.69
	mc2	-2.77	-2.38	-2.67

can u provide one example patient?

C. Downstream task

[TB3] indicates the result for Mortality/sepsis prediction in different settings and hospital centers.

In some cases, incorporating event embeddings can lead to better performance.

D. Learned represention s

Fig 1 visualizes the tsne plot for the two scenarios.

E. Model interpretability

one advantage of proposed method is use of attention mechanisms in both event and state encoder. Fig 1 shows the attention mechanism

F. Likelihood estimation

Although CIF does not improve mark prediction, it has led to better representation of patient for downstream task such as sepsis prediction.

In addition, we can interpret some of learned CIF patterns.
explain the effect of time concatenation in SO dataset
tsne of learned representation. 4 modes:

- (DA,TE)-¿(Mark, CIF)

attention of DA for sepsis prediction
attention matrix of events for SO dataset

VII. Conclusion

References

[1] Simiao Zuo et al. “Transformer Hawkes Process”. In: *Proceedings of the 37th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, Nov. 21, 2020, pp. 11692–11702. URL: https://proceedings.mlr.press/v119/zuo20a.html (visited on 05/06/2022).

TABLE III
ADD CAPTION

Dataset	Setting	Center	F1		AUPRC		AUROC	
			DAM	TE+DAM	DAM	TE+DAM	DAM	TE+DAM
P12	sc	1	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
		2	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
		3	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
	mc1	1	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
		2	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
		3	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
	mc2	-	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
	seft	-	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
P19	sc	1	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
		2	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
		3	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
	mc1	1	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
		2	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
		3	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
	mc2	-	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)
	seft	-	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)	0.55 (0.02)