# Recurrent Marked Temporal Point Processes: Embedding Event History to Vector

Nan Du
Georgia Tech
dunan@gatech.edu

Hanjun Dai
Georgia Tech
hanjundai@gatech.edu

Rakshit Trivedi
Georgia Tech
rstrivedi@gatech.edu

Utkarsh Upadhyay
MPI-SWS
utkarshu@mpi-sws.org

Manuel Gomez-Rodriguez
MPI-SWS
manuelgr@mpi-sws.org

Le Song
Georgia Tech
lsong@cc.gatech.edu

## ABSTRACT

Large volumes of event data are becoming increasingly available in a wide variety of applications, such as healthcare analytics, smart cities and social network analysis. The precise time interval or the exact distance between two events carries a great deal of information about the dynamics of the underlying systems. These characteristics make such data fundamentally different from independently and identically distributed data and time-series data where time and space are treated as indexes rather than random variables. Marked temporal point processes are the mathematical framework for modeling event data with covariates. However, typical point process models often make strong assumptions about the generative processes of the event data, which may or may not reflect the reality, and the specifically fixed parametric assumptions also have restricted the expressive power of the respective processes. Can we obtain a more expressive model of marked temporal point processes? How can we learn such a model from massive data?

In this paper, we propose the Recurrent Marked Temporal Point Process (RMTPP) to simultaneously model the event timings and the markers. The key idea of our approach is to view the intensity function of a temporal point process as a nonlinear function of the history, and use a recurrent neural network to automatically learn a representation of influences from the event history. We develop an efficient stochastic gradient algorithm for learning the model parameters which can readily scale up to millions of events. Using both synthetic and real world datasets, we show that, in the case where the true models have parametric specifications, RMTPP can learn the dynamics of such models without the need to know the actual parametric forms; and in the case where the true models are unknown, RMTPP can also learn the dynamics and achieve better predictive performance than other parametric alternatives based on particular prior assumptions.


*KDD '16, August 13-17, 2016, San Francisco, CA, USA*

DOI: http://dx.doi.org/10.1145/2939672.2939875


Figure 1: Given the trace of past locations and time, can we predict the location and time of the next stop?

## Keywords

Marked temporal point process; Stochastic process; Recurrent neural network;

## 1. INTRODUCTION

Event data with marker information can be produced from social activities, to financial transactions, to electronic health records, which contains rich information about what *type* of event is happening between *which entities* by *when* and *where*. For instance, people might visit various places at different moments of a day. Algorithmic trading systems buy and sell large volume of stocks within short-time frames. Patients regularly go to the clinic with a longitudinal data of diagnoses about their concerned diseases.

Although the aforementioned situations come from a broad range of domains, we are interested in a commonly encountered question: based on the observed sequence of events, *can we predict what kind of event will take place at what time in the future*? Accurately predicting the type and the timing of the next event will have many interesting applications. For mainstream personal assistants, shown in Figure 1, since people tend to visit different places specific to the temporal/spatial contexts, successfully predicting their next destinations at the most likely time will make such services more relevant and usable. In stock market, accurately forecasting when to sell or buy a particular stock means critical business success. For modern healthcare, patients may have several diseases that have complicated dependencies on each other. Accurately estimating when a clinical event might occur can effectively facilitate patient-specific care and prevention to reduce the potential future risks.

Existing studies in literature attempt to approach this problem mainly in two ways: first, classic varying-order Markov models [4] formulate the problem as a discrete-time

sequence prediction task. Based on the observed sequence of states, they can predict the most likely state the process will evolve into on the next step. As a result, one limit of the family of classic Markov models is that it assumes the process proceeds by unit time-steps, so it cannot capture the heterogeneity of the time to predict the timing of the next event. Furthermore, when the number of states is large, Markov model usually cannot capture long dependency on the history since the overall state-space will grow exponentially in the number of time steps considered. Semi-Markov model [26] can model the continuous time-interval between two successive states to some extent by assuming the intervals have very simple distributions, but it still has the same state-space explosion issue when the order grows.

Second, marked temporal point processes and intensity functions are a more general mathematical framework for modeling such event data. For example, in seismology, marked temporal point processes have originally been widely used for modeling earthquakes and aftershocks [20, 21, 31]. Each earthquake can be represented as a point in the temporal-spatial space, and seismologists have proposed different formulations to capture the randomness of these events. In the financial area, temporal point processes are active research topics of econometrics, which often leads to many simple interpretations of the complex dynamics of modern electronic markets [2, 3].

However, typical point process models, such as the Hawkes processes [20], the autoregressive conditional duration processes [12], are making specific assumptions about the functional forms of the generative processes, which may or may not reflect the reality, and thus the respective fixed simple parametric representations may restrict the expressive power of these models. How can we obtain a more expressive model of marked temporal point processes, and learn such a model from large volume of data? In this paper, we propose a novel marked temporal point process, referred to as the Recurrent Marked Temporal Point Process, to simultaneously model the event timings and markers. The key idea of our approach is to view the intensity function of a temporal point process as a nonlinear function of the history of the process, and parameterize the function using a recurrent neural network. More specifically, our work makes the following contributions:

- We propose a novel marked point process to jointly model the time and the marker information by learning a general representation of the nonlinear dependency over the history based on recurrent neural networks. Using our model, event history is embedded into a compact vector representation which can be used for predicting the next event time and marker type.
- We point out that the proposed Recurrent Marked Temporal Point Process establishes a previously unexplored connection between recurrent neural networks and point processes, which has implications beyond temporal-spatial settings by incorporating more rich contextual information and features.
- We conduct large-scale experiments on both synthetic and real-world datasets across a wide range of domains to show that our model has consistently better performance for predicting both the event type and timing compared to alternative competitors.

## 2. PROBLEM DEFINITION

The input data is a set of sequences $\mathcal{C} = \{\mathcal{S}^1, \mathcal{S}^2, \dots\}$. Each $\mathcal{S}^i = \big((t_1^i, y_1^i), (t_2^i, y_2^i), \dots\big)$ is a sequence of pairs $(t_j^i, y_j^i)$ where $t_j^i$ is the time when the event of type (or marker) $y_j^i$ has occurred to the entity $i$, and $t_j^i < t_{j+1}^i$. Depending on specific applications, the entity and the event type can have different meanings. For example, in transportation, $\mathcal{S}^i$ can be a trace of time and location pairs for a taxi $i$ where $t_j^i$ is the time when the taxi picks up or drops off customers in the neighborhood $y_j^i$. In financial transactions, $\mathcal{S}^i$ can be a sequence of time and action pairs for a particular stock $i$ where $t_j^i$ is the time when a transaction of selling ($y_j^i = 0$) or buying ($y_j^i = 1$) has occurred. In electronic health records, $\mathcal{S}^i$ is a series of clinical events for patient $i$ where $t_j^i$ is the time when the patient is diagnosed with the disease $y_j^i$. Despite that these applications emerge from a diverse range of domains, we want to build models which are able to:

- Predict the next event pair $(t_{n+1}^i, y_{n+1}^i)$ given a sequence of past events for entity $i$;
- Evaluate the likelihood of a given sequence of events;
- Simulate a new sequence of events based on the learned parameters of the model.

## 3. RELATED WORK

Temporal point processes [6] are mathematical abstractions for many different phenomena across a wide range of domains. In seismology, marked temporal point processes have originally been widely used for modeling earthquakes and aftershocks [20, 19, 21, 31]. In computational finance, temporal point processes are very active research topics in econometrics [2, 3]. In sociology, temporal-spatial point processes have been used to model networks of criminals [35]. In human activity modeling, Poisson Process and its variants, have been used to model the inter-event durations of human activities [29, 15]. More recently, the self-excitation point process [20], has become an ongoing hot topic for modeling the latent dynamics of information diffusion [16, 9, 10, 8, 40, 14, 22], online-user engagement [13], news-feed streams [7], and context-aware recommendations [11].

A major limitation of these existing studies is that they often draw various parametric assumptions about the latent dynamics governing the generation of the observed point patterns. In contrast, in this work, we seek to propose a model that can learn a general and efficient representation of the underlying dynamics from the event history without assuming a fixed parametric forms in advance. The advantage is that the proposed model can be more flexible to be automatically adapted to the data. We compare the proposed RMTPP with many other processes of specific parametric forms in Section 6 to demonstrate the superb robustness of RMTPP to model misspecification.

## 4. MARKED TEMPORAL POINT PROCESS

Marked temporal point process is a powerful mathematical tool to model the latent mechanisms governing the observed random event patterns along time. Since the occurrence of an event may be triggered by what happened in the past, we can essentially specify models for the timing of the next event given what we have already known so far. More formally, a marked temporal point process is a random process of which the realization consists of a list of discrete events localized in time, $\{t_j, y_j\}$, with the timing $t_j \in \mathbb{R}^+$,

the marker $y_j \in \mathcal{Y}$ and $j \in \mathbb{Z}^+$ . Let the history $\mathcal{H}_t$ be the list of event time and marker pairs up to the time $t$. The length $d_{j+1} = t_{j+1} - t_j$ of the time interval between neighboring successive events $t_j$ and $t_{j+1}$ is referred to as the inter-event duration.

Given the history of past events, we can explicitly specify the conditional density function that the next event will happen at time $t$ with type $y$ as $f^*(t, y) = f(t, y|\mathcal{H}_t)$ where $f^*(t, y)$ emphasizes that this density is conditional on the history. By applying the chaining rule, we can derive the joint likelihood of observing a sequence as the following:

$$f\left(\{(t_j, y_j)\}_{j=1}^n\right) = \prod_j f(t_j, y_j|\mathcal{H}_t) = \prod_j f^*(t_j, y_j) \quad (1)$$

One can design many forms for $f^*(t_j, y_j)$. However, in practice, people typically choose very simple factorized formulations like $f(t_j, y_j|\mathcal{H}_t) = f(y_j)f(t_j|\ldots, t_{j-2}, t_{j-1})$ due to the excessive complications caused by jointly and explicitly modeling the timing and the marker information. One can think of $f(y_j)$ as a multinomial distribution when $y_j$ can only take finite number of values and is totally independent on the history. $f(t_j|\ldots, t_{j-2}, t_{j-1})$ is the conditional density of the event occurring at the time $t_j$ given the timing sequence of past events. However, note that $f^*(t_j)$ cannot capture the influence of past markers.

## 4.1 Parametrizations

The temporal information in a marked point process can be well captured by a typical temporal point process. An important way to characterize temporal point processes is via the conditional intensity function — the stochastic model for the next event time given all previous events. Within a small window $[t, t + dt)$, $\lambda^*(t)dt$ is the probability for the occurrence of a new event given the history $\mathcal{H}_t$:

$$\lambda^*(t)dt = \mathbb{P}\left\{\text{event in } [t, t + dt)|\mathcal{H}_t\right\}. \quad (2)$$

The $*$ notation reminds us that the function depends on the history. Given the conditional density function $f^*(t)$, the conditional intensity function can be specified as:

$$\lambda^*(t)dt = \frac{f^*(t)dt}{S^*(t)} = \frac{f^*(t)dt}{1 - F^*(t)}, \quad (3)$$

where $F^*(t)$ is the cumulative probability that a new event will happen before time $t$ since the last event time $t_n$, and $S^*(t) = \exp\left(-\int_{t_n}^t \lambda^*(\tau)d\tau\right)$ is the respective probability that no new event has ever happened up to time $t$ since $t_n$. As a consequence, the conditional density function can be alternatively specified by

$$f^*(t) = \lambda^*(t) \exp\left(-\int_{t_n}^t \lambda^*(\tau)d\tau\right). \quad (4)$$

Particular functional forms of the conditional intensity function $\lambda^*(t)$ are often designed to capture the phenomena of interests [1]. In the following, we review a few representative examples of typical point processes where the conditional intensity has particularly specified parametric forms.

**Poisson process [28].** The homogeneous Poisson process is the simplest point process. The inter-event times are independent and identically distributed random variables conforming to the exponential distribution. The conditional intensity function is assumed to be independent of the history $\mathcal{H}_t$ and keeping constant over time, *i.e.*, $\lambda^*(t) = \lambda_0 \geqslant 0$. For a more general inhomogeneous poisson process, the intensity is also assumed to be independent of the history $\mathcal{H}_t$ but it can be a function varying over time, *i.e.*, $\lambda^*(t) = g(t) \geqslant 0$.

**Hawkes process [20].** A Hawkes process captures the mutual excitation phenomenon among events with the conditional intensity being defined as

$$\lambda^*(t) = \gamma_0 + \alpha \sum_{t_j < t} \gamma(t, t_j), \quad (5)$$

where $\gamma(t, t_j) \geqslant 0$ is the triggering kernel capturing temporal dependencies, $\gamma_0 \geqslant 0$ is a baseline intensity independent of the history and the summation of kernel terms is history dependent and a stochastic process by itself. The kernel function can be chosen in advance, *e.g.*, $\gamma(t, t_j) = \exp(-\beta\,(t - t_j))$ or $\gamma(t, t_j) = \mathbb{I}[t > t_j]$, or directly learned from data. A distinctive feature of the Hawkes process is that the occurrence of each historical event increases the intensity by a certain amount. Since the intensity function depends on the history up to time $t$, the Hawkes process is essentially a conditional Poisson process (or doubly stochastic Poisson process [27]) in the sense that conditioned on the history $\mathcal{H}_t$, the Hawkes process is a Poisson process formed by the superposition of a background homogeneous Poisson process with the intensity $\gamma_0$ and a set of inhomogeneous Poisson processes with the intensity $\gamma(t, t_j)$. However, because the events in a past interval can affect the occurrence of the events in later intervals, the Hawkes process in general is more expressive than a Poisson process.

**Self-correcting process [25].** In contrast to the Hawkes process, the self-correcting process seeks to produce regular point patterns with the conditional intensity function

$$\lambda^*(t) = \exp\left(\mu t - \sum_{t_i < t} \alpha\right), \quad (6)$$

where $\mu > 0, \alpha > 0$. The intuition is that while the intensity increases steadily, every time when a new event appears, it is decreased by multiplying a constant $e^{-\alpha} < 1$, so the chance of new points decreases after an event has occurred recently.

**Autoregressive Conditional Duration process [12].** An alternative way of conditional intensity parametrization is to capture the dependency between inter-event durations. Let $d_i = t_i - t_{i-1}$. The expectation for $d_i$ is given by $\psi_i = E(d_i|\ldots, d_{i-2}, d_{i-1})$. The simplest form assumes that $d_i = \psi_i\epsilon_i$ where $\epsilon_i$ is independently and identically distributed exponential variables with expectation one. As a consequence, the conditional intensity has the following form:

$$\lambda^*(t) = \psi_{N(t)}^{-1}, \quad (7)$$

where $\psi_i = \gamma_0 + \sum_{j=0}^m \alpha_j d_{i-j}$ to capture the influences from the most recent $m$ durations, and $N(t)$ is the total number of events up to $t$.

## 4.2 Major Limitations

**Curse of Model Misspecification.** All these different parameterizations of the conditional intensity function seek to capture and represent certain forms of dependency on the history in different ways: Poisson process makes the assumption that the duration is stationary; Hawkes process assumes that the influences from past events are linearly additive towards the current event; Self-correcting process specifies a non-linear dependency over these past events; and autoregressive conditional duration model imposes a linear struc-

ture between successive inter-event durations. These different parameterizations *encode our prior knowledge* about the latent dynamics we try to model. In practice, however, the true model is never known. Thus, we have to try different specifications for $\lambda^*(t)$ to tune the predictive performance and most often we can expect to suffer from certain errors caused by the model misspecification.

**Marker Generation.** Furthermore, it is quite often that we have additional information (or covariates) associated with each event like the markers. For instance, the marker of a NYC taxi can be the neighborhood-name of the place where it picks up (or drops off) passengers; the marker of each financial transaction can be the action of buying (or selling); and the marker of a clinical event can be the diagnosis of the major disease. Classic temporal point processes can be extended to capture the marker information mainly in the following two ways: first, the marker is directly incorporated into the intensity function; second, each marker can be regarded as an independent dimension to have a multi-dimensional temporal point process. In terms of the former approach, we still need to specify a proper form for the conditional intensity function. Moreover, due to the extra complexity of the function induced by the markers, people normally make strong assumptions that the marker is independent on the history [33], which greatly reduces the flexibility of the model. With respect to the latter method, it is very often to have large number of markers, which results in a sparsity problem associated with each dimension where only very few events can happen.

# 5. RECURRENT MARKED TEMPORAL POINT PROCESS

Each parametric form of the conditional intensity function determines the temporal characteristics of a family of point processes. However, it will be hard to correctly decide which form to use without any sufficient prior knowledge in order to take into account both the marker and the timing information. To tackle this challenge, in this section, we propose a unified model capable of modeling a general nonlinear dependency over the history of both the event timing and the marker information.

## 5.1 Model Formulation

By carefully investigating the various forms of the conditional intensity function (5), (6), and (7), we can observe that they are inherently different representations and realizations of various kinds of dependency structures over the past events. Inspired by this critical insight, we seek to learn a general representation to *approximate* the unknown dependency structure over the history.

Recurrent Neural Network (RNN) is a feedforward neural network structure where additional edges, referred to as the recurrent edges, are added such that the outputs from the hidden units at the current time step are fed into them again as the future inputs at the next time step. In consequence, the same feedforward neural network structure is replicated at each time step, and the recurrent edges connect the hidden units of the network replicates at adjacent time steps together along time, that is, the hidden units with recurrent edges not only receive the input from the current data sample but also from the hidden units in the last time step. This feedback mechanism creates an internal state of the network to memorize the influence of each past data sample. In theory, finite-sized recurrent neural networks with sigmoidal activation units can simulate a universal Turing machine [36], which is able to perform an extremely rich family of computations. In practice, RNN has been shown to be a powerful tool for general purpose sequence modeling. For instance, in Natural Language Processing, recurrent neural network has state-of-the-arts predictive performance for sequence-to-sequence translations [24], image captioning [38], handwriting recognition [18]. It has also been used for discrete-time series data prediction [30, 39, 34](treat time as discrete indices) for a long time.

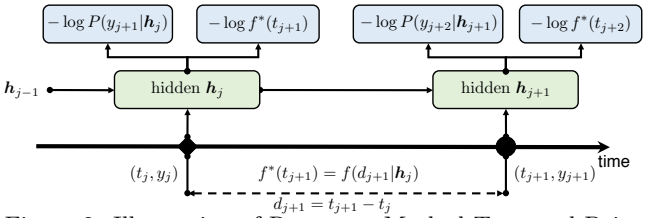


Figure 2: Illustration of Recurrent Marked Temporal Point Process. For each event with the timing $t_j$ and the marker $y_j$, we treat the pair $(t_j, y_j)$ as the input to a recurrent neural network where the embedding $\boldsymbol{h}_j$ up to the time $t_j$ learns a general representation of a nonlinear dependency over both the timing and the marker information from past events. Note that the solid diamond and the circle on the timeline indicate two events of different types $y_j \neq y_{j+1}$.

Our key idea is to let the RNN (or its modern variant LSTM [23], GRU [5], *etc.*) model the nonlinear dependency over both of the markers and the timings from past events. As shown in Figure 2, for the event occurring at the time $t_j$ of type $y_j$, the pair $(t_j, y_j)$ is fed as the input into a recurrent neural network unfolded up to the $j+1$-th event. The embedding $\boldsymbol{h}_{j-1}$ represents the memory of the influence from the timings and the markers of past events. The neural network updates $\boldsymbol{h}_{j-1}$ to $\boldsymbol{h}_j$ by taking into account the effect of the current event $(t_j, y_j)$. Since now $\boldsymbol{h}_j$ represents the influence of the history up to the $j$-th event, the conditional density for the next event timing can be naturally represented as

$$f^*(t_{j+1}) = f(t_{j+1}|\mathcal{H}_t) = f(t_{j+1}|\boldsymbol{h}_j) = f(d_{j+1}|\boldsymbol{h}_j), \quad (8)$$

where $d_{j+1} = t_{j+1} - t_j$. As a consequence, we can depend on $\boldsymbol{h}_j$ to make predictions to the timing $\hat{t}_{j+1}$ and the type $\hat{y}_{j+1}$ of the next event.

The advantage of this formulation is that we explicitly embed the event history into a latent vector space, and by the elegant relation (4), we are now able to capture a general form of the conditional intensity function $\lambda^*(t)$ without the need of specifying a fixed parametric specification for the dependency structure over the history. Figure 3 presents the overall architect of the proposed RMTPP. Given a sequence of events $\mathcal{S} = \big((t_j, y_j)_{j=1}^n\big)$, we design an RNN which computes a sequence of hidden units $\{\boldsymbol{h}_j\}$ by iterating the following components.

**Input Layer.** At the $j$-th event, the input layer first projects the sparse one-hot vector representation of the marker $y_j$ into a latent space. We add an embedding layer with the weight matrix $\boldsymbol{W}_{em}$ to achieve a more compact and efficient representation $\boldsymbol{y}_j = \boldsymbol{W}_{em}^\top y_j + \boldsymbol{b}_{em}$, where $\boldsymbol{b}_{em}$ is the bias. We learn $\boldsymbol{W}_{em}$ and $\boldsymbol{b}_{em}$ while we train the network. In addition, for the timing input $t_j$, we can extract the associ-

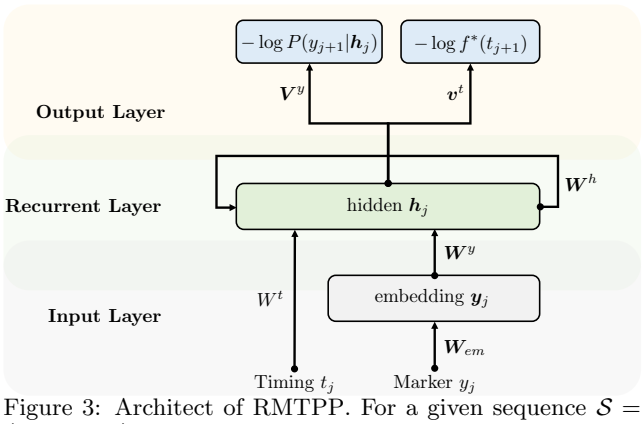



Figure 3: Architect of RMTPP. For a given sequence $\mathcal{S} = \big((t_j, y_j)_{j=1}^n\big)$, at the $j$-th event, the marker $y_j$ is first embedded into a latent space. Then, the embedded vector and the temporal features are fed into the recurrent layer. The recurrent layer learns a representation that summaries the nonlinear dependency over the previous events. Based on the learned representation $\boldsymbol{h}_j$, it outputs the prediction for the next marker $\hat{y}_{j+1}$ and timing $\hat{t}_{j+1}$ to calculate the respective loss functions.

ated temporal features $\boldsymbol{t}_j$ (*e.g.*, like the inter-event duration $d_j = t_j - t_{j-1}$).

**Hidden Layer.** We update the hidden vector after receiving the current input and the memory $\boldsymbol{h}_{j-1}$ from the past. In RNN, we have

$$\boldsymbol{h}_j = \max\left\{\boldsymbol{W}^y \boldsymbol{y}_j + \boldsymbol{W}^t \boldsymbol{t}_j + \boldsymbol{W}^h \boldsymbol{h}_{j-1} + \boldsymbol{b}_h, 0\right\}. \quad (9)$$

**Marker Generation.** Given the learned representation $\boldsymbol{h}_j$, we model the marker generation with a multinomial distribution by

$$P(y_{j+1} = k|\boldsymbol{h}_j) = \frac{\exp\left(\boldsymbol{V}^y_{k,:}\boldsymbol{h}_j + b^y_k\right)}{\sum_{k=1}^K \exp\left(\boldsymbol{V}^y_{k,:}\boldsymbol{h}_j + b^y_k\right)}, \quad (10)$$

where $K$ is the number of markers, and $\boldsymbol{V}^y_{k,:}$ is the $k$-th row of matrix $\boldsymbol{V}^y$.

**Conditional Intensity.** Based on $\boldsymbol{h}_j$, we can now formulate the conditional intensity function by

$$\lambda^*(t) = \exp\Big(\underbrace{\boldsymbol{v}^{t\top} \cdot \boldsymbol{h}_j}_{\substack{\text{past}\\ \text{influence}}} + \underbrace{w^t(t - t_j)}_{\substack{\text{current}\\ \text{influence}}} + \underbrace{b^t}_{\substack{\text{base}\\ \text{intensity}}}\Big), \quad (11)$$

where $\boldsymbol{v}^t$ is a column vector, and $w^t$, $b^t$ are scalars. More specifically,

- The first term $\boldsymbol{v}^{t\top} \cdot \boldsymbol{h}_j$ represents the accumulative influence from the marker and the timing information of the past events. Compared to the fixed parametric formulations of (5), (6), and (7) for the past influence, we now have a highly non-linear general specification of the dependency over the history.
- The second term emphasizes the influence of the current event $j$.
- The last term gives a base intensity level for the occurrence of the next event.
- The exponential function outside acts as a non-linear transformation and guarantees that the intensity is positive.

By invoking the elegant relation between the conditional intensity function and the conditional density function in (4), we can derive the likelihood that the next event will occur at the time $t$ given the history by the following equation:

$$\begin{aligned} f^*(t) &= \lambda^*(t) \exp\left(-\int_{t_j}^t \lambda^*(\tau) d\tau\right) \\ &= \exp\Bigg\{\boldsymbol{v}^{t\top} \cdot \boldsymbol{h}_j + w^t(t - t_j) + b^t + \frac{1}{w}\exp(\boldsymbol{v}^{t\top} \cdot \boldsymbol{h}_j + b^t) \\ &\quad - \frac{1}{w}\exp(\boldsymbol{v}^{t\top} \cdot \boldsymbol{h}_j + w^t(t - t_j) + b^t)\Bigg\}. \end{aligned} \quad (12)$$

Then, we can estimate the timing for the next event using the expectation

$$\hat{t}_{j+1} = \int_{t_j}^{\infty} t \cdot f^*(t) dt. \quad (13)$$

In general, the integration in (13) does not have analytic solutions, so we can apply commonly used numerical integration techniques [32] for one-dimensional functions to compute (13) instead.

**Remark.** Based on the hidden unit of RNN, we are able to learn a unified representation of the dependency over the history. In consequence, the direct formulation (11) of the conditional intensity function $\lambda^*(t_{j+1})$ captures both of the information from past event timings and event markers. On the other hand, since the prediction for the marker also depends nonlinearly on the past timing information, this may improve the performance of the classification task as well when both of these two information are correlated with each other. In fact, experiments on synthetic and real world datasets in the following experimental section do verify such mutual boosting phenomenon.

## 5.2 Parameter Learning

Given a collection of sequences $\mathcal{C} = \{\mathcal{S}^i\}$, where $\mathcal{S}^i = \big((t^i_j, y^i_j)_{j=1}^{n_i}\big)$, we can learn the model by maximizing the joint log-likelihood of observing $\mathcal{C}$.

$$\ell(\{\mathcal{S}^i\}) = \sum_i \sum_j \Big(\log P(y^i_{j+1}|\boldsymbol{h}_j) + \log f(d^i_{j+1}|\boldsymbol{h}_j)\Big), \quad (14)$$

We exploit the Back Propagation Through Time (BPTT) for training RMTPP. Given the size of BPTT as $b$, we unroll our model in Figure 3 by $b$ steps. In each training iteration, we take $b$ consecutive samples $\{(t^i_k, y^i_k)_{k=j}^{j+b}\}$ from a single sequence, apply the feed-forward operation through the network, and update the parameters with respect to the loss function. After we unroll the model for $b$ steps through time, all the parameters are shared across these copies, and will be updated sequentially in the back propagation stage. In our algorithm framework[1], we need both sparse (the marker $y_j$) and dense features at time $t_j$. Meanwhile, the output is also mixed of discrete markers and real-value time, which is then fed into different loss functions including the cross-entropy of the next predicted marker and the negative log-likelihood of the next predicted event timing. Therefore, we build an efficient and flexible platform[2] particularly optimized for training general directed acyclic structured computational graph (DAG). The backend is supported via CUDA and MKL for GPU and CPU platform, respectively. In the end, we apply stochastic gradient descent (SGD) with mini-batch and several other techniques of training neural networks [37].

[1]https://github.com/dunan/NeuralPointProcess
[2]https://github.com/Hanjun-Dai/graphnn

# 6. EXPERIMENT

We evaluate RMTPP in large-scale synthetic and real world data. We compare it to several discrete-time and continuous-time sequential models showing that RMTPP is more robust to model misspecificationmisspecification than these alternatives.

## 6.1 Baselines

To evaluate the predictive performance of forecasting markers, we compare with the following discrete-time models:

- **Majority Prediction**. This is also known as the 0-order **M**arkov **C**hain (**MC-0**), where at each time step, we always predict the most popular marker regardless of the history. Most often, predicting the most popular type is a strong heuristic.
- **Markov Chain**. We compare with Markov models with varying orders from one to three, denoted as **MC-1**, **MC-2**, and **MC-3**, respectively.

To show the effectiveness of predicting time, we compare with the following continuous-time models:

- **ACD**. We fit a second-order autoregressive conditional duration process with the intensity function given in (7).
- **Homogeneous Poisson Process**. The intensity function is a constant, which produces an estimate of the average inter-event gaps.
- **Hawkes Process**. We fit a self-excitation Hawkes process with the intensity function in (5).
- **Self-correcting Process**. We fit a self-correcting process with the intensity function in (6).

Finally, we compare with the **C**ontinuous-**T**ime **M**arkov **C**hain (**CTMC**) model, which learns continuous transition rates between two states (or markers). This model predicts the next state with the earliest transition time, so it can predict both the marker and the timing for the next event jointly.

## 6.2 Synthetic Data

To show the robustness of RMTPP, we propose the following generative processes[3]:

**Autoregressive Conditional Duration.** The conditional density function for the next duration $d_n$ conforms to an exponential distribution with the expectation determined by the past $m$ subsequent durations in the following form, which is denoted as ACD.

$$f(d_n|\mathcal{H}_{n-1}) = \alpha_n \exp(-\alpha_n \tau_n), \alpha_n = \left(\mu_0 + \gamma \sum_{i=1}^{m} d_{n-i}\right)^{-1}, \tag{15}$$

where $\mu_0$ is the base duration to generate the first event starting from zero, $d_1 \sim (\mu_0)^{-1} \exp(-d_1/\mu_0)$. We set $m = 2$, $\mu_0 = 0.5$ and $\gamma = 0.25$.

**Hawkes Process.** The conditional intensity function is given by $\lambda(t) = \lambda_0 + \alpha \sum_{t_i<t} \exp\left(-\frac{t-t_i}{\sigma}\right)$ where $\lambda_0 = 0.2$, $\alpha = 0.8$ and $\sigma = 1.0$.

**Self-Correcting Process.** The conditional intensity function is given by $\lambda(t) = \exp\left(\mu t - \sum_{t_i<t} \alpha\right)$ where $\mu = 1$ and $\alpha = 0.2$.

**State-Space Continuous-Time Model.** To model the influence from both markers and time, we further propose the State-Time Mixture model with the following steps:

1. For each time $t_{n-1}$, we take the mod of $t_{n-1}$ by a period of $P = 24$. If the residual is greater than 12, the process is defined to be in the time state $r_{n-1} = 0$; otherwise, it is in the time state $r_{n-1} = 1$.
2. Based on the combination of both the time state $\{r_{n-j}\}_{j=1}^{m}$ and the marker state $\{y_{n-j}\}_{j=1}^{m}$ of the previous $m$ events, the process will have the marker $k$ for the next step in the probability $P(y_n = k| \{y_{n-j}\}_{j=1}^{m}, \{r_{n-j}\}_{j=1}^{m})$.
3. Similarly, based on the combination of $\{y_{n-j}\}_{j=1}^{m}$ and $\{r_{n-j}\}_{j=1}^{m}$ from the previous $m$ events, the duration $d_n = t_n - t_{n-1}$ has a Poisson distribution with the expectation determined by $\{r_{n-j}\}_{j=1}^{m}$ and $\{y_{n-j}\}_{j=1}^{m}$ jointly. Here, we use the Poisson distribution to mimic the elapsed time units (*e.g.*, hours, minutes).

In our experiments, without loss of generality, we set the total number of markers to two, $m = 3$ and randomly initialize the transition probabilities between states.

**Experimental Results.** Figure 4 presents the predictive performance of RMTPP fitted to different types of time-series data, where we simulate 1,000,000 events and use 90% for training and the rest 10% for testing for each case. We first compare the predictive performance of RMTPP with the optimal estimator in the left column where the optimal estimator knows the true conditional intensity function. We treat the expectation of the time interval between the current and the next event as our estimation. Grey curves are the observed inter-event durations from 100 successive events in the testing data. Blue curves are the respective expectations given by the optimal estimator. Red curves are the predictions from RMTPP. We can observe that even though RMTPP has no prior knowledge about the true functional form of each process, its predictive performance is almost consistent with the respective optimal estimator.

The middle column of Figure 4 compares the learned conditional intensity functions (red curves) with the true ones (blue curves). It clearly demonstrates that RMTPP is able to adaptively and accurately capture the unknown heterogeneous temporal dynamics of different time-series data. In particular, because the order of dependency over the history is fixed for ACD, RMTPP almost exactly learns the conditional intensity function with comparable BPTT steps. The Hawkes and the self-correcting processes are more challenging in that the conditional intensity function depends on the entire history. Because the events are far from being uniformly distributed, the influence from individual past event to the occurrence of new future events can vary widely. From this perspective, these processes essentially have random varying order dependency on the history compared to ACD. However, with properly chosen BPTT steps, RMTPP can accurately capture the general shape and each single change point of the true intensity function. In particular, for the Hawkes case, the abruptly increased intensity from time index 60 to 100 results in 40 events in a very tiny time interval, but still, the predictions of RMTPP can capture the trend of the true data.

The right column of Figure 4 reports the overall RMSE of different processes between the predictions and the true testing data. We can observe that RMTPP has very strong competitive performance and better robustness against model misspecification to capture the heterogeneity of the latent temporal dynamics of different time-series data compared to other parametric alternatives.

In addition to time, the state-space continuous-time model

[3]https://github.com/dunan/MultiVariatePointProcess

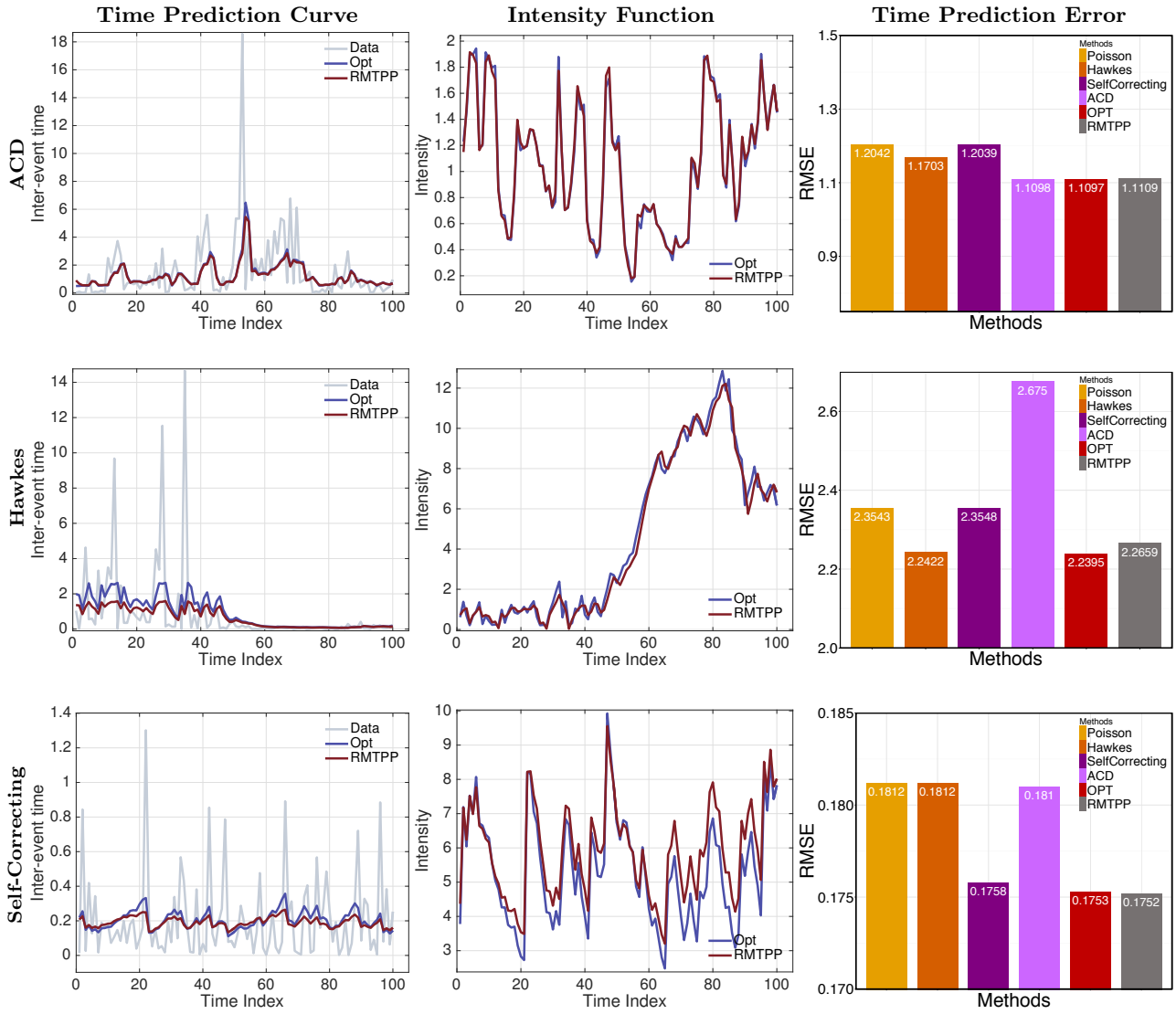
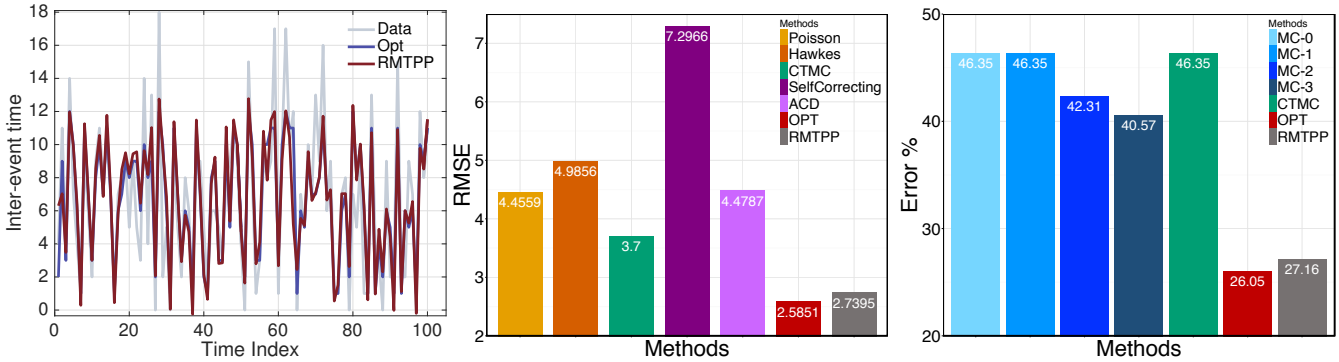



Figure 4: Time predictions on the testing data produced from different processes. Left column is the predicted inter-event time. Blue curve is the optimal estimator, and red curve is RMTPP without any prior knowledge about the true dynamics of each case. Middle column shows the learned intensity functions vs. the respective true ones. Right column gives the overall testing RMSE of predicting the timings from different processes.




(a) Time Prediction Curve (b) Time Prediction Error (c) Marker Prediction Error

Figure 5: Performance evaluation of predicting timings and markers on the state-space continuous-time model.

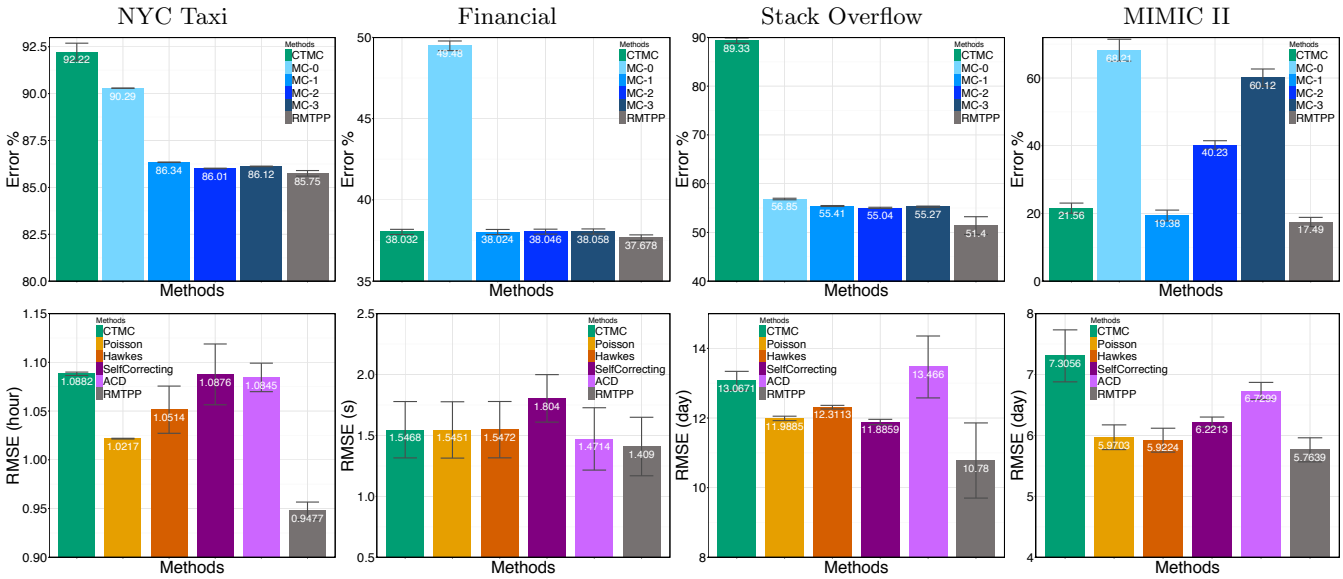
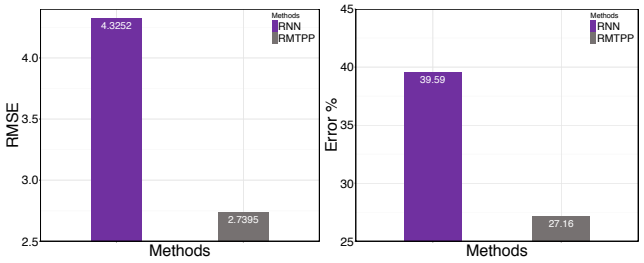



Figure 7: Performance evaluation for predicting both marker and timing of the next event. The top row presents the classification error of predicting th event markers, and the bottom row gives the RMSE of predicting the event timings.




(a) Time Prediction Error (b) Marker Prediction Error

Figure 6: Predictive performance comparison with RNN which is trained for predicting the next timing only in (a), and for predicting the next marker only in (b).

also includes the marker information. Figure 5 compares the error rates of different processes in predicting both event timings and markers. Compared to the other baselines, RMTPP is again consistent with the optimal estimator without any prior knowledge about the true underlying generative process.

Finally, since the occurrences of future events depend on both of the past marker and timing information, we would like to investigate whether learning a unified representation of the joint information can further improve future predictions. Therefore, we train an RNN by only using the temporal and the marker information, respectively. Figure 6 gives the comparisons between RMTPP and RNN where in panel (a), RNN has the 4.3522 RMSE while RMTPP achieves a 2.7395 RMSE, and in panel (b), RNN reports 39.59% classification error while RMTPP reaches to the 27.16% level. Clearly, they verify that the joint modeling of both information can boost the performance of predicting future events.

## 6.3 Real Data

We evaluate the predictive performance of RMTPP on real world datasets from a diverse range of domains.

**New York City Taxi Dataset.** The NYC taxi dataset[4] contains ∼173 million trip records of individual Taxi for consecutive 12 months in 2013. The location information is available in the form of latitude/longitude coordinates. Each record also contains the temporal information of pick-up (drop-off) passengers associated with every trip. We have used NYC Neighborhood Names GIS dataset[5] to map the coordinates to neighborhood names. For those coordinates of which the location name is not directly available in the GIS dataset, we use geodesic distance to map them to the nearest neighborhood name. With this process we obtained 299 unique locations as our markers. An event is a pickup record for a taxi. Further, we have divided each single sequence of a taxi into multiple fine-grained subsequences where two consecutive events are within 12 hours. We obtained 670,753 sequences in total out of which 536,603 were used for training and 134,150 were used for testing. We predict the location and the time of the next pickup event.

**Financial Transaction Dataset.** We have collected a raw limited order book data from NYSE of the high-frequency transactions for a stock in one day. It contains 0.7 million transaction records, each of which records the time (in millisecond) and the possible action (B = buy, S = sell). We treat the type of actions as markers. The input data is a single long sequence with 624,149 events for training and 69,350 events for testing. The task is to predict which action will be taken next at what time.

**Electrical Medical Records.** MIMIC II medical dataset is a collection of de-identified clinical visit records of Intensive Care Unit patients for seven years. We have filtered out 650 patients and 204 diseases. Each event records the time when a patient had a visit to the hospital. We have used the sequences of 585 patients to train, and the rest for test. The goal is to predict which major disease will happen to a given patient at what time in the future.

**Stack OverFlow Dataset.** Stack Overflow[6] is a question-

[4]http://www.andresmh.com/nyctaxitrips/

[5]https://data.cityofnewyork.us/City-Government/Neighborhood-Names-GIS/99bc-9p23

[6]https://archive.org/details/stackexchange

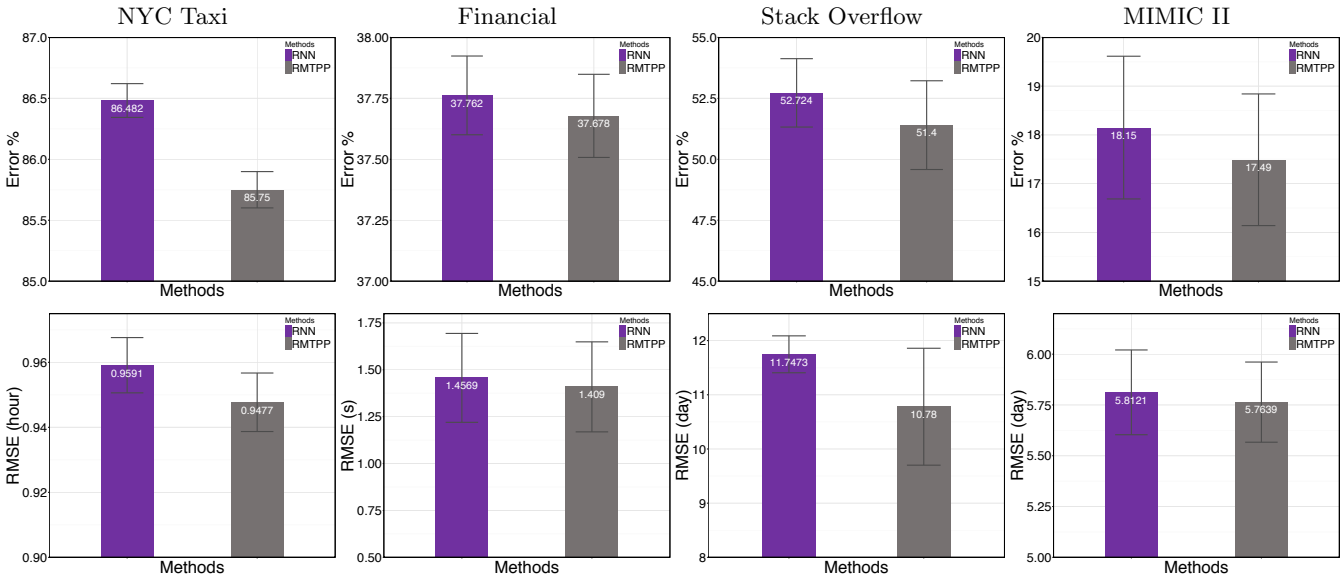
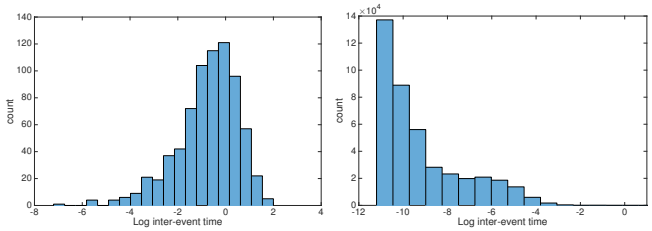



Figure 8: Predictive performance comparison with RNN which are trained only using the markers in the left column and only using the temporal information in the right column.

answering website which exploits badges to encourage user engagement and guide behaviors [17]. There are 81 types of non-topical (i.e., non-tag affiliated) badges which can be awarded either only once (e.g. Altruist, Inquisitive, etc.) or multiple times (e.g. Stellar Question, Guru, Great Answer, etc.) to a user. By ignoring the badges which can be awarded only once, we first select users who have earned at least 40 badges between 2012-01-01 and 2014-01-01 and then those badges which have been awarded at least 100 times to the users selected in the first step. We have removed the users who have been instantaneously awarded multiple badges due to technical issues of the servers. In the end, we have ∼6 thousand users with a total of ∼480 thousand events where each badge is treated as a marker.

**Experimental Results.** We compare and report the predictive performance of different models on the testing data of each dataset in Figure 7. The hyper-parameters of RMTPP across all these datasets are tuned as following: learning rate in $\{0.1, 0.01, 0.001\}$; hidden layer size in $\{64, 128, 256, 512, 1024\}$; momentum $= 0.9$ and L2 penalty $= 0.001$; and batch-size in $\{16, 32, 64\}$. Figure 7 compares the predictive performance of forecasting markers and timings for the next event of different processes across the four real datasets. RMTPP outperforms the other alternatives with lower errors for predicting both timings and markers. Because the MIMIC-II dataset has many short sequences and is the smallest out of the four datasets, increasing the order of Markov chain will decrease its classification performance.

We also compare RMTPP with RNN trained only with the marker and with the timing information separately in Figure 8. We can observe that RMTPP trained by incorporating both the past marker and timing information performs consistently better than RNN trained with either one source of the information alone. Finally, Figure 9 shows the empirical distribution for the inter-event times on the Stack Overflow and the financial transaction data. Compared to the other temporal processes of fixed parametric forms, even though the real datasets might have quite different characteristics, RMTPP is more flexible to capture such heterogeneity in general.




(a) Stack Overflow (b) Financial Transaction

Figure 9: Empirical distribution for the inter-event times. The x-axis is in log-scale.

## 7. DISCUSSIONS

We present the Recurrent Marked Temporal Point Process, which builds a connection between recurrent neural networks and point processes. The recurrent neural network supports different architects, including the classic RNN and modern LSTM, GRU, *etc.* Besides, in addition to the inter-event temporal features, our model can be readily generalized to incorporate other contextual information. For instance, in addition to training a global model, we can also take the potential user-profile features into account for personalization. Furthermore, based on the structural information of social networks, our model can be generalized in such a way that the prediction of one user sequence depends not only on her own history but also on the other users' history to capture their interactions in a networked setting.

To conclude, RMTPP inherits the advantages from both the recurrent neural networks and the temporal point processes to predict both the marker and the timing of the future events without any prior knowledge about the hidden functional forms of the latent temporal dynamics. Experiments on both synthetic and real world datasets demonstrate that RMTPP is robust to model mis-specifications and has consistently better performance compared to the other alternatives.

## Acknowledgements

This project was supported in part by NSF/NIH BIGDATA 1R01GM108341, ONR N00014-15-1-2340, NSF IIS-1218749, and NSF CAREER IIS-1350983.

## 8. REFERENCES

[1] O. Aalen, O. Borgan, and H. Gjessing. *Survival and event history analysis: a process point of view*. Springer, 2008.

[2] E. Bacry, A. Iuga, M. Lasnier, and C.-A. Lehalle. Market impacts and the life cycle of investors orders. 2014.

[3] E. Bacry, T. Jaisson, and J.-F. Muzy. Estimation of slowly decreasing hawkes kernels: Application to high frequency order book modelling. 2015.

[4] R. Begleiter, R. El-Yaniv, and G. Yona. On prediction using variable order markov models. *J. Artif. Intell. Res. (JAIR)*, 22:385–421, 2004.

[5] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[6] D. Daley and D. Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*, volume 2. Springer, 2007.

[7] N. Du, M. Farajtabar, A. Ahmed, A. J. Smola, and L. Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *KDD*. ACM, 2015.

[8] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *NIPS*, 2013.

[9] N. Du, L. Song, A. J. Smola, and M. Yuan. Learning networks of heterogeneous influence. In *NIPS*, 2012.

[10] N. Du, L. Song, H. Woo, and H. Zha. Uncover topic-sensitive information diffusion networks. In *Artificial Intelligence and Statistics (AISTATS)*, 2013.

[11] N. Du, Y. Wang, N. He, and L. Song. Time-sensitive recommendation from recurrent user activities. In *NIPS*, 2015.

[12] R. F. Engle and J. R. Russell. Autoregressive conditional duration: A new model for irregularly spaced transaction data. *Econometrica*, 66(5):1127–1162, Sep 1998.

[13] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song. Shaping social activity by incentivizing users. In *NIPS*, 2014.

[14] M. Farajtabar, M. Gomez-Rodriguez, N. Du, M. Zamani, H. Zha, and L. Song. Back to the past: Source identification in diffusion networks from partially observed cascades. In *AISTAT*, 2015.

[15] A. Ferraz Costa, Y. Yamaguchi, A. Juci Machado Traina, C. Traina, Jr., and C. Faloutsos. Rsc: Mining and modeling temporal activity in social media. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 269–278, 2015.

[16] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the International Conference on Machine Learning*, 2011.

[17] S. Grant and B. Betts. Encouraging user behaviour with achievements: an empirical study. In *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*, pages 65–68. IEEE, 2013.

[18] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, , and J. Schmidhuber. A novel connectionist system for uncon- strained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 855–868, 2009.

[19] A. G. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society Series B*, 33:438–443, 1971.

[20] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[21] A. G. Hawkes and D. Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pages 493–503, 1974.

[22] X. He, T. Rekatsinas, J. Foulds, L. Getoor, and Y. Liu. Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In *ICML*, pages 871–880, 2015.

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

[24] O. V. Ilya Sutskever and Q. V. Le. Sequence to sequence learning with neural networks. 2014.

[25] V. Isham and M. Westcott. A self-correcting pint process. *Advances in Applied Probability*, 37:629–646, 1979.

[26] J. Janssen and N. Limnios. *Semi-Markov Models and Applications*. Kluwer Academic, 1999.

[27] J. Kingman. On doubly stochastic poisson processes. *Mathematical Proceedings of the Cambridge Philosophical Society*, pages 923–930, 1964.

[28] J. F. C. Kingman. *Poisson processes*, volume 3. Oxford university press, 1992.

[29] R. D. Malmgren, D. B. Stouffer, A. E. Motter, and L. A. N. Amaral. A Poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences*, 105(47):18153–18158, 2008.

[30] H. Min, X. Jiahui, X. Shiguo, and Y. Fuliang. Prediction of chaotic time series based on the recurrent predictor neural network. *IEEE Transactions on Signal Processing*, 52:3409–3416, 2004.

[31] Y. Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998.

[32] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C. The Art of Scientific Computation*. Cambridge University Press, Cambridge, UK, 1994.

[33] J. G. Rasmussen. Temporal point processes: the conditional intensity function. http://people.math.aau.dk/~{}jgr/teaching/punktproc11/tpp.pdf, 2009.

[34] C. Rohitash and Z. Mengjie. Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction. *Neurocomputing*, 86:116–123, 2012.

[35] M. Short, G. Mohler, P. Brantingham, and G. Tita. Gang rivalry dynamics via coupled point process networks. *Discrete and Continuous Dynamical Systems Series B*, 19:1459–1477, 2014.

[36] H. T. Siegelmann and E. D. Sontag. Turing computability with neural nets. *Applied Mathematics Letters*, 4:77–80, 1991.

[37] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. On the importance of initialization and momentum in deep learning. In S. Dasgupta and D. Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 1139–1147. JMLR Workshop and Conference Proceedings, May 2013.

[38] O. Vinyals, A. Toshev, S. Bengio, , and D. Erhan. Show and tell: A neural image caption generator. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[39] C. Xindi, Z. Nian, V. Ganesh K., and W. I. Donald C. Time series prediction with recurrent neural networks trained by a hybrid psoâĂŞea algorithm. *Neurocomputing*, 70:2342–2353, 2007.

[40] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1513–1522, 2015.