Modeling Inter-Dependence Between Time and Mark in Multivariate Temporal Point Processes

Govind Waghmare
Mastercard, AI Garage
Gurugram, India
govind.waghmare@mastercard.com

Ankur Debnath
Mastercard, AI Garage
Gurugram, India
ankur.debnath@mastercard.com

Siddhartha Asthana
Mastercard, AI Garage
Gurugram, India
siddhartha.asthana@mastercard.com

Aakarsh Malhotra
Mastercard, AI Garage
Gurugram, India
aakarsh.malhotra@mastercard.com

ABSTRACT

Temporal Point Processes (TPP) are probabilistic generative frameworks. They model discrete event sequences localized in continuous time. Generally, real-life events reveal descriptive information, known as marks. Marked TPPs model time and marks of the event together for practical relevance. Conditioned on past events, marked TPPs aim to learn the joint distribution of the time and the mark of the next event. For simplicity, conditionally independent TPP models assume time and marks are independent given event history. They factorize the conditional joint distribution of time and mark into the product of individual conditional distributions. This structural limitation in the design of TPP models hurt the predictive performance on entangled time and mark interactions. In this work, we model the conditional inter-dependence of time and mark to overcome the limitations of conditionally independent models. We construct a multivariate TPP conditioning the time distribution on the current event mark in addition to past events. Besides the conventional intensity-based models for conditional joint distribution, we also draw on flexible intensity-free TPP models from the literature. The proposed TPP models outperform conditionally independent and dependent models in standard prediction tasks. Our experimentation on various datasets with multiple evaluation metrics highlights the merit of the proposed approach.

CCS CONCEPTS

• **Information systems** → **Location based services**.

KEYWORDS

multivariate temporal point processes; probabilistic modeling

ACM Reference Format:
Govind Waghmare, Ankur Debnath, Siddhartha Asthana, and Aakarsh Malhotra. 2022. Modeling Inter-Dependence Between Time and Mark in Multivariate Temporal Point Processes. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557399

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557399

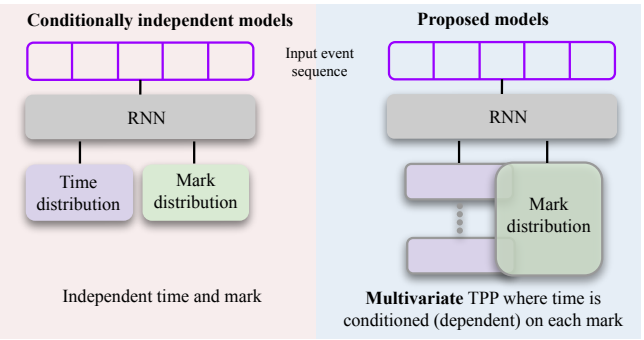

Figure 1: The proposed models are conditionally dependent, multivariate, and capable of employing both intensity-free and intensity-based formulations.

1 INTRODUCTION

TPP is a random process representing irregular event sequences occurring in continuous time. Financial transactions, earthquakes, and electronic health records (EHR) exhibit asynchronous temporal patterns. TPPs are well studied in the literature and have rich theoretical foundations [3, 5, 13]. Classical (non-neural) TPPs focus on capturing relatively simple temporal patterns through Poison process [18], self-excitation process [13], and self-correcting process [15]. With the advent of neural networks, many flexible and efficient neural architectures have been developed to model multi-modal event dynamics, called neural TPPs [30].

Any attribute associated with an event makes it more realistic and represented as a mark. Marks capture a better description of the event, like time and location, interacting entities, and their evolution. Stochastic modeling of such events to study underlying event generation mechanisms is called the marked TPPs. For instance, in seismology, earthquake event dynamics are better understood with the knowledge of magnitude and location [4]. A temporal model solely learned on time may not be of practical relevance where

marks impart realistic and reliable information. Marked TPP is a probabilistic framework [6] which aims to model the joint distribution of time and mark of the next event using previous event history. An estimation of the next event time and the mark has practical application in many domains that exhibit complex time and mark interactions. Such application include online user engagements [11, 16, 37], information diffusion [28], econometrics [1], and healthcare [10]. In personalized healthcare, a patient could have a complex medical history, and several diseases may depend on each other. Predictive EHR modeling could reveal potential future clinical events and facilitate efficient resource allocation.

Time and mark dependency: While modeling the conditional joint distribution of time and marks, many prior works assume marks to be conditionally independent of time [8, 25]. This assumption on the conditional joint distribution of time and mark leads to two types of marked TPPs, (i) conditionally independent, and (ii) conditionally dependent models. The independence assumption allows factorization of the conditional joint distribution into a product of two independent conditional distributions. It is the product of continuous-time distribution and categorical mark distribution[1], both conditioned on the event history. The independence between time and mark limits the structural design of the neural architecture in conditionally independent models. Thus, such models require fewer parameters to specify the conditional joint distribution of time and marks but fail to capture their dependence. On the contrary, conditionally dependent models capture the dependency between time and mark by either conditioning time distribution on mark or mark distribution on time. A recent study by [10] shows that the conditionally independent models perform poorly compared to conditionally dependent models.

Multivariate TPP: Marked TPP is a joint probability distribution over a given time interval. In order to model time and mark dependency, the time distribution should be conditioned on all possible marks. It leads to a multivariate TPP model where a tuple of time distributions is learned over a set of categorical marks [21]. For K distinct marks, k^{th} multivariate distribution ($k \in \{1, \ldots, K\}$) indicates the joint distribution of the time and the k^{th} mark.

Intensity-based vs intensity-free modeling: In both conditionally independent and conditionally dependent models, inter-event time distribution is a key factor of the joint distribution. The standard way of learning time distribution is by estimating conditional intensity function. However, the intensity function requires selecting good parametric formulation [29]. The parametric intensity function often makes assumptions about the latent dynamics of the point process. A simple parametrization has limited expressiveness but makes likelihood computation easy. Though an advanced parametrization adequately captures event dynamics, likelihood computation often involves numerical approximation using Newton-Raphson or Monte Carlo (MC). Besides intensity-based formulation, other ways to model conditional inter-event time distribution involve probability density function (PDF) modeling, cumulative distribution function, survival function, and cumulative intensity function [24, 30]. A model based on an intensity-free focuses on closed-form likelihood, closed-form sampling, and flexibility to approximate any distribution.

[1]Categorical marks are conventional in the prior works.

In this work, we model inter-dependence between time and mark by learning conditionally dependent distribution. While inferring the next event, we model a PDF of inter-event time distribution for each discrete mark. The time distribution conditioned on marks improves the predictive performance of the proposed models compared to others. A high-level overview of our approach is shown in Figure 1. In summary, we make the following contributions:

- We overcome the structural design limitation of conditionally independent models by proposing novel *conditionally dependent, both intensity-free and intensity-based,* and *multivariate* TPP models. To capture inter-dependence between mark and time, we condition the time distribution on the current mark in addition to event history.
- We improve the predictive performance of the intensity-based models through conditionally dependent modeling. Further, we draw on the intensity-free literature to design a flexible multivariate marked TPP model. We model the PDF of conditional inter-event time to enable closed-form likelihood computation and closed-form sampling.
- Using multiple metrics, we provide a comprehensive evaluation of a diverse set of synthetic and real-world datasets. The proposed models consistently outperform both conditionally independent and conditionally dependent models.

2 RELATED WORK

In this section, we provide a brief overview of classical (non-neural) TPPs and neural TPPs. Later, we discuss conditionally independent and conditionally dependent models. In the end, we differentiate the proposed solution against state-of-the-art models in the literature.

2.1 Classical (non-neural) TPPs

TPPs are mainly described via conditional intensity function. Basic TPP models make suitable assumptions about the underlying stochastic process resulting in constrained intensity parametrizations. For instance, Poisson process [18, 26] assumes that inter-event times are independent. In Hawkes process [14, 23] event excitation is positive, additive over time, and decays exponentially with time. Self-correcting process [15] and autoregressive conditional duration process [9] propose different conditional intensity parametrizations to capture inter-event time dynamics. These constraints on conditional intensity limit the expressive power of the models and hurt predictive performance due to model misspecification [8].

2.2 Neural TPPs

Neural TPPs are more expressive and computationally efficient than classical TPPs due to their ability to learn complex dependencies. A TPP model inferring the time and mark of the next event *sequentially* is called autoregressive (AR) TPP. A seminal work by [8, 35] connects the point processes with a neural network by realizing conditional intensity function using a recurrent neural network (RNN). Generally, the event history is encoded using either recurrent encoders or set aggregation encoders [38, 39].

Conditionally independent models assume time and mark are independent and inferred from the history vector representing past events. This assumption makes this neural architecture computationally inexpensive but hurts the predictive performance as

Table 1: Comparison of the proposed models with other neural temporal point processes.

Model	Conditionally dependent modeling	Intensity-free modeling
Conditional Poisson (CP)	✗	✗
RMTPP ([8])	✗	✗
LNM ([29])	✗	✓
NHP ([21])	✓	✗
SAHP ([38])	✗	✗
THP ([39])	✗	✗
Proposed RMTPP	✓	✗
Proposed LNM	✓	✓
Proposed THP	✓	✗

the influence of mark and time on each other cannot be modeled. Therefore, all conditional independent models perform similarly on mark prediction due to their limited expressiveness [29]. Further, modeling time distribution based on conditional intensity is conventional in multiple prior models [8, 35]. The training objective in these models involves numerical approximations like MC estimates. On the contrary, [29] proposed intensity-free learning of TPPs where PDF of inter-event times is learned directly (bypassing intensity parametrization) via log-normal mixture (LNM) distribution. LNM model focuses on flexibility, closed-form likelihood, and closed-form sampling.

Conditionally dependent models capture dependency either by conditioning time on marks [10, 21, 39] or marks on time [2]. In [10, 21], a separate intensity function is learned for each mark at every time step making it multivariate TPP. [12, 22, 31] discuss the scalability of models when the number of marks is large. These models are intensity-based and hence share the same drawbacks discussed previously compared to intensity-free.

In the proposed models, we realize the conditional dependence of time and marks. We rely on both standard intensity-based and intensity-free formulations to realize PDF of inter-event time. For intensity-based case, we draw on well-known models like RMTPP [8] and THP [39], called as ***proposed RMTPP*** and ***proposed THP*** respectively. The intensity-free model allows analytical (closed-form) computation of likelihood. We draw on the conditionally independent log-normal mixture (LNM) model proposed by [29] to design a conditionally dependent multivariate TPP model known as ***proposed LNM***. The advantage of the proposed methods compared to the state-of-the-art models is shown in Table 1.

3 MODEL FORMULATION

3.1 Background and notations

We represent variable-length event sequence with time and mark attributes as $E = \{e_1 = (t_1, m_1), \ldots, e_N = (t_N, m_N)\}$ over time interval $[0, T]$ where $t_1 < \cdots < t_N$ are event arrival times and $m_i \in \mathcal{M}$ are categorical marks from the set $\mathcal{M} = \{1, 2, \ldots, K\}$. The number of events N in the interval $[0, T]$ is a random variable. The inter-event times are given as $\tau_i = t_i - t_{i-1}$ with $t_0 = 0$ and $t_{N+1} = T$. The event history till the time t is stated as $\mathcal{H}_t = \{(t_i, m_i) : t_i < t\}$. The joint distribution of the next event conditioned on past events is defined as $P(t_i, m_i|\mathcal{H}_{t_i}) = P^*(t_i, m_i)$. Here, $*$ symbol here indicates the joint distribution is conditioned on the event history $\mathcal{H}_{t_i}$ [6]. Ordinarily, multivariate TPP with K categorical marks is characterized by conditional intensity function $\lambda_k^*(t)$ for the event of type k. It is defined as

$$\lambda_k^*(t) = \lim_{dt \to 0} \frac{\Pr(\text{event of type } k \text{ in } [t, t+dt)|\mathcal{H}_t)}{dt} \tag{1}$$

For unmarked case, number of marks, K=1 and Equation 1 becomes $\lambda_k^*(t) = \lambda^*(t)$. Here, $\lambda^*(t)$ is called as *ground intensity* [27]. The conditional inter-event time PDF for the i^{th} event of type k is given as

$$f_{ik}^*(\tau_i) = \lambda_k^*(t_{i-1} + \tau_i) \exp\left(-\sum_{k=1}^{K} \int_{t_{i-1}}^{t_i} \lambda_k^*(t')dt'\right) \tag{2}$$

Here, $\tau_i = t_i - t_{i-1}$ indicates inter-event time is isomorphic with arrival-time and could be used interchangeably.

Conditionally independent models factorize the conditional joint distribution $P_i^*(\tau_i, m_i)$ into product of two independent distributions $P_i^*(\tau_i)$ and $P_i^*(m_i)$. The conditional joint density[2] of the time and the mark is represented as

$$f_i^*(\tau_i, m_i) = f_i^*(\tau_i) \cdot p_i^*(m_i), \tag{3}$$

where, $f_i^*(\tau_i)$ is PDF of the time distribution $P_i^*(\tau_i)$ and $p_i^*(m_i)$ is the probability mass function (PMF) of categorical mark distribution $P_i^*(m_i)$. Now, there are two ways to model the time PDF $f_i^*(\tau_i)$. One way is to use conditional intensity function as follows

$$f_i^*(\tau_i) = \lambda^*(t_{i-1} + \tau_i) \exp\left(-\int_{t_{i-1}}^{t_i} \lambda^*(t')dt'\right), \tag{4}$$

and other is parametric density estimation of PDF $f_i^*(\tau_i)$ using history $\mathcal{H}_{t_i}$. In conditionally independent models, time distribution is not conditioned on the current mark. So, $f_i^*(\tau_i|m_i) = f_i^*(\tau_i)$ and the model does not capture the influence of the current mark on time distribution.

Conditionally dependent models capture the dependency between τ_i and m_i either by conditioning time on mark or by conditioning mark on time. When time is conditioned on marks, a separate distribution $P_i^*(\tau_i|m_i = k)$ is specified for each mark $k \in \mathcal{M}$. Here, the conditional joint density for each mark takes the following form:

$$f_i^*(\tau_i, m_i = k) = f_i^*(\tau_i|m_i = k) \cdot p_i^*(m_i = k), \tag{5}$$

Usually, the time PDF $f_i^*(\tau_i|m_i = k) = f_{ik}^*(\tau_i)$ is represented using parametrized intensity function (Equation 2). When marks are conditioned on the time, a distribution $P_i^*(m_i|\tau_i = \tau)$ is specified for all values of τ. [2] parametrized the distribution $P_i^*(m_i|\tau_i = \tau)$ using Gaussian process. Here, the joint density at k^{th} mark is given as

$$f_i^*(\tau_i, m_i = k) = f_i^*(\tau_i) \cdot p_i^*(m_i = k|\tau_i = t - t_{i-1}), \tag{6}$$

where, $t_{i-1} < t \leq t_i$ and time PDF $f_i^*(\tau_i)$ is generally obtained using Equation 4.

[2]We use the conditional density term in the broad sense. Here, time is continuous random variable and mark is discrete random variable [27].

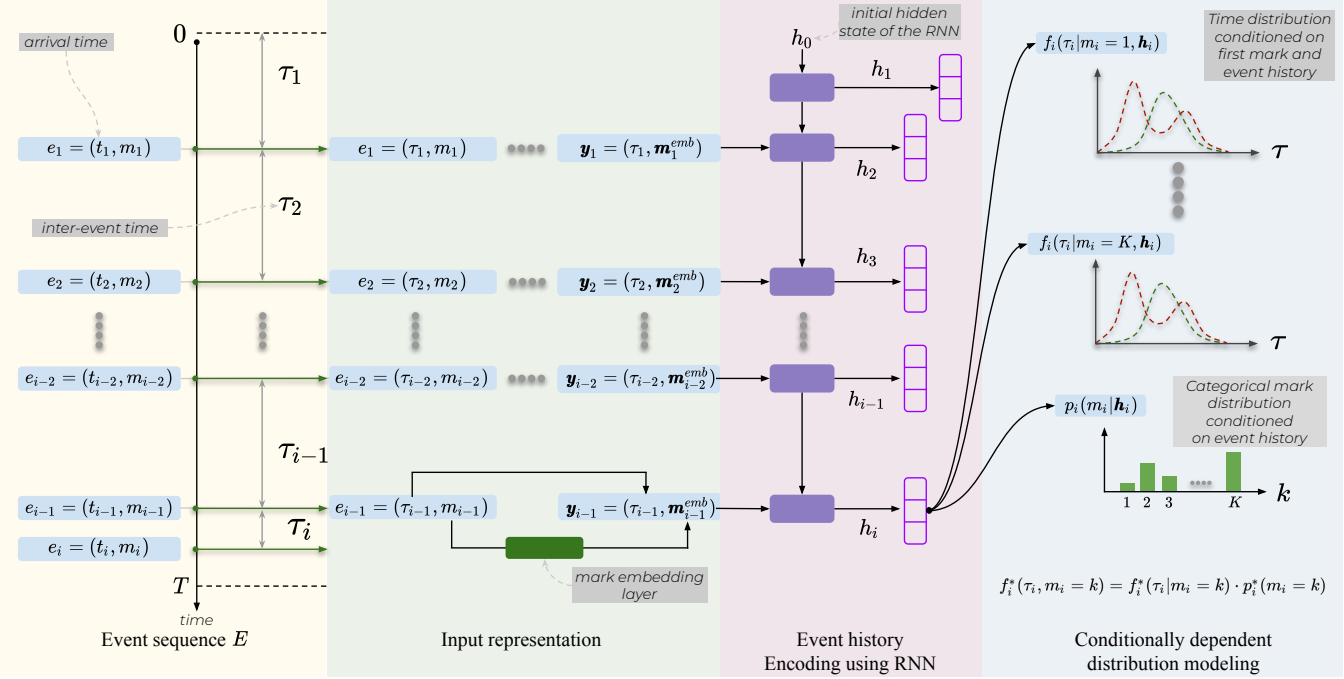

Figure 2: Overview of the proposed multivariate conditionally dependent model. The inter-event time distribution is learned either using an intensity-free or intensity-based approach. Input event sequence contains arrival time and mark for each event. Input representation contains inter-event time and mark embedding. RNN converts event history into a fixed dimension vector. In the end, we compute the conditional joint density of time and marks.

3.2 Proposed approach

We model the conditional joint distribution of the time and the mark by conditioning time on marks (Equation 5). We specify inter-event time PDF conditioned on each mark type. A schematic representation of the proposed approach is shown in Figure 2. Note that, the proposed approach is common for both intensity-based and intensity-free models. For intensity-based models, time PDF $f^*_{ik}(\tau_i)$ in Equation 5 is realized using Equation 2. For both *proposed RMTPP* and *proposed THP* models, we use parametrized intensity functions defined in the papers [8] and [39] respectively. We condition these intensity functions on marks (Equation 2) to alleviate the structurally independent time and mark assumption. Intensity-free based approach (*proposed LNM*) is explained further. A parametric density estimation approach based on the log-normal mixture (LNM) is proposed by [29] for conditionally independent models. We draw on this approach to design multivariate TPP capable of capturing inter-dependence between time and marks. We realize a conditional joint density $f^*_i(\tau_i, m_i = k)$ of the i^{th} event with k^{th} type using event history $\mathcal{H}_{t_i}$ till $(i-1)^{\text{th}}$ event. For a given variable length event sequence E, each event is represented as $e_j = (t_j, m_j)$. Categorical marks are encoded using embedding function as $\boldsymbol{m}^{emb}_j = Embedding(m_j)$. Here, the embedding function is a learnable matrix $\boldsymbol{E} \in \mathbb{R}^{K \times |\boldsymbol{m}^{emb}_j|}$ and $\boldsymbol{m}^{emb}_j = \text{one-hot}(m_j) \cdot \boldsymbol{E}$. We concatenate inter-event time τ_j and mark embedding $\boldsymbol{m}^{emb}_j$ to form input feature $\boldsymbol{y_j} = (\tau_j, \boldsymbol{m}^{emb}_j)$. The RNN converts the input representation $(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{i-1})$ into fixed-dimensional history vector $\boldsymbol{h}_i$. Here, $\mathcal{H}_{t_i} = \boldsymbol{h}_i$. Starting with initial hidden state $\boldsymbol{h}_0$, next hidden state of the RNN is updated as $\boldsymbol{h}_i = Update(\boldsymbol{h}_{i-1}, \boldsymbol{y}_{i-1})$. For conditionally dependent multivariate TPP, we learn PDF of inter-event time using log-normal mixture model as follows:

$$f^*(\tau | m = k) = f^*_k(\tau) = f_k(\tau | \boldsymbol{w}_k, \boldsymbol{\mu}_k, \boldsymbol{s}_k), \tag{7}$$

where, $\boldsymbol{w}$ are the mixture weights, $\boldsymbol{\mu}$ are the mixture means and $\boldsymbol{s}$ are the mixture standard deviations. Further, inline with [29],

$$f_k(\tau | \boldsymbol{w}_k, \boldsymbol{\mu}_k, \boldsymbol{s}_k) = \sum_{c=1}^{C} w_{k,c} \frac{1}{\tau s_{k,c} \sqrt{2\pi}} \exp\left(-\frac{(\log \tau - \mu_{k,c})^2}{2 s^2_{k,c}} \right), \tag{8}$$

where, $c \in \{1, 2, \ldots, C\}$ indicates number of mixture components. We discuss the selection of C and its impact on the result in Table 4. For each mark $k \in \mathcal{M}$, the parameters $\boldsymbol{w}_k, \boldsymbol{\mu}_k$ and $\boldsymbol{s}_k$ are estimated from $\boldsymbol{h}$ as follows[3]:

$$\boldsymbol{w}_k = \text{softmax}(\boldsymbol{W}_{\boldsymbol{w}_k} \boldsymbol{h} + \boldsymbol{b}_{\boldsymbol{w}_k}) \tag{9}$$

$$\boldsymbol{\mu}_k = \exp(\boldsymbol{W}_{\boldsymbol{\mu}_k} \boldsymbol{h} + \boldsymbol{b}_{\boldsymbol{\mu}_k}) \text{ and } \boldsymbol{s}_k = \boldsymbol{W}_{\boldsymbol{s}_k} \boldsymbol{h} + \boldsymbol{b}_{\boldsymbol{s}_k} \tag{10}$$

Here, $\{\boldsymbol{W}_{\boldsymbol{w}_k}, \boldsymbol{W}_{\boldsymbol{\mu}_k}, \boldsymbol{W}_{\boldsymbol{s}_k}, \boldsymbol{b}_{\boldsymbol{w}_k}, \boldsymbol{b}_{\boldsymbol{\mu}_k}, \boldsymbol{b}_{\boldsymbol{s}_k}\}$ are the learnable parameters of the neural network. We parametrize the categorical mark distribution for mark prediction. The history vector $\boldsymbol{h}$ is passed through linear layer with weight matrix $\boldsymbol{W}_m = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ and bias vector $\boldsymbol{b}_m = [b_1, \ldots, b_K]$. Here, $\boldsymbol{W}_m \in \mathbb{R}^{|\boldsymbol{h}| \times K}$. The mark distribution is computed using softmax function as follows:

[3] subscript i (event index) is dropped for simplicity

$$p(m = k|\boldsymbol{h}) = p^*(m = k) = \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{h} + b_k)}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^\top \boldsymbol{h} + b_j)} \tag{11}$$

In conditionally independent models, $f_i^*(\tau_i, m_i = k) = f_i^*(\tau_i) \cdot p_i^*(m_i = k)$. As $f_i^*(\tau_i)$ is independent of mark, estimation of mark is done as follows:

$$\arg\max_{k \in \mathcal{M}} f_i^*(\tau_i, m_i) \coloneqq \arg\max_{k \in \mathcal{M}} p_i^*(m_i = k) \tag{12}$$

On the other hand, in conditionally dependent models, mark is estimated as follows:

$$\arg\max_{k \in \mathcal{M}} f_i^*(\tau_i, m_i) \coloneqq \arg\max_{k \in \mathcal{M}} f_i^*(\tau | m = k) \cdot p_i^*(m_i = k) \tag{13}$$

3.3 Likelihood estimation

As neural TPP is a generative framework, maximum likelihood estimation (MLE) is a widely used training objective. Other objectives could be inverse reinforcement learning [20, 32], Wasserstein distance [7, 34] and adversarial losses [33, 36]. In the proposed approaches we use the MLE objective. For the event sequence $E = \{e_1 = (t_1, m_1), \ldots, e_N = (t_N, m_N)\}$ in the interval $[0, T]$, the likelihood function represents the joint density of all the events. So, likelihood is factorized into the product of conditional joint densities (of time and mark) for each event. The negative log-likelihood (NLL) is formulated as:

$$-\log p(E) = -\sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1}_{(m_i=k)} \log f_i(\tau_i, m_i = k|\mathcal{H}_{t_i}) - \log(1 - P(T|\mathcal{H}_T)), \tag{14}$$

where, $f_i(\tau_i, m_i = k|\mathcal{H}_{t_i})$ is the conditional joint density of event with mark type k and $(1 - P(T|\mathcal{H}_T)$ indicates no event of any type has occurred in the interval between t_N and T (survival probability of the last interval). As *proposed RMTPP* and *proposed THP* use conditional intensity-based NLL formulation, they requires approximation of the integral in the Equation 1 using MC [8, 21]. In the *proposed LNM*, mixture model enables computation of NLL analytically which is more accurate and computationally efficient than MC approximations [30]. For NLL computation, we factorize $f_i(\tau_i, m_i = k|\mathcal{H}_{t_i})$ according to the Equation 5.

4 EXPERIMENTAL EVALUATION

4.1 Datasets

We perform experiments on commonly used synthetic and real-world benchmark datasets in the marked TPP literature. All datasets contain multiple unique sequences and show variations in the sequence length. We include three synthetic datasets and three real-world datasets for experimentation. Dataset details and summary statistics are given in the Table 2.

4.1.1 Synthetic datasets. Using Hawkes dependent and independent processes, we generate three datasets. These datasets are commonly used in state-of-the-art models like [10, 25, 29]. Hawkes process is self-exciting point process with following conditional intensity function represented as [4]:

$$\lambda_k^*(t) = \mu_k + \sum_{j=1}^{K} \sum_{i:t_{j,i}<t} \alpha_{k,j} \beta_{k,j} \exp(-\beta_{k,j}(t - t_{j,i})) \tag{15}$$

Here, μ_k is base intensity, $\alpha_{k,j}$ is excitation (intensity) between event types and $\beta_{k,j}$ is a decay of the exponential kernel. Inline with [10, 21], using different values of $\boldsymbol{\mu}, \boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we generate Hawkes independent (denoted as Hawkes Ind.) and Hawkes dependent dataset (denotes as Hawkes Dep. (I)). Hawkes Ind. and Hawkes Dep. (I) are comparatively simple datasets. Therefore, we also generate another Hawkes dependent dataset (denoted as Hawkes Dep. (II) with five different marks and longer average sequence length to make prediction challenging (see Table 2). For the Hawkes Ind. dataset, we use the following parameters:

$$\boldsymbol{u} = \begin{bmatrix} 0.1 & 0.05 \end{bmatrix} \quad \boldsymbol{\alpha} = \begin{bmatrix} 0.2 & 0.0 \\ 0.0 & 0.4 \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 2.0 \end{bmatrix} \tag{16}$$

For Hawkes Dep. (I) dataset, we use following parameters:

$$\boldsymbol{u} = \begin{bmatrix} 0.1 & 0.05 \end{bmatrix} \quad \boldsymbol{\alpha} = \begin{bmatrix} 0.2 & 0.1 \\ 0.2 & 0.3 \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix} \tag{17}$$

For Hawkes Dep. (II) dataset, we randomly sample parameters inline with [21] as follows:

$$\boldsymbol{u} = \begin{bmatrix} 0.713 & 0.057 & 0.844 & 0.254 & 0.344 \end{bmatrix} \tag{18}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} 0.689 & 0.549 & 0.066 & 0.819 & 0.007 \\ 0.630 & 0.000 & 0.457 & 0.622 & 0.141 \\ 0.134 & 0.579 & 0.821 & 0.527 & 0.795 \\ 0.199 & 0.556 & 0.147 & 0.030 & 0.649 \\ 0.353 & 0.557 & 0.892 & 0.638 & 0.836 \end{bmatrix} \tag{19}$$

$$\boldsymbol{\beta} = \begin{bmatrix} 9.325 & 9.764 & 2.581 & 4.007 & 9.319 \\ 5.759 & 8.742 & 4.741 & 7.320 & 9.768 \\ 2.841 & 4.349 & 6.920 & 5.640 & 3.839 \\ 6.710 & 7.460 & 3.685 & 4.052 & 6.813 \\ 2.486 & 2.214 & 8.718 & 4.594 & 2.642 \end{bmatrix} \tag{20}$$

4.1.2 Real-world datasets. For real-world datasets, we use publicly available common benchmark datasets like Stack Overflow[5] [8], MOOC[6] [19] and MIMIC-II[7] [10]. Stack Overflow is a question-answering website. Users on this site earn badges as a reward for contribution. For each user, the event sequence represents different badges received over two years. MOOC dataset captures interactions of learners with the online course system. Different actions like taking a course and solving an assignment are different kinds of marks. MIMIC-II dataset contains anonymized electronic health records of the patients visiting the intensive care unit for seven years. Each event represents the time of the hospital visit. The mark indicates the type of disease (75 unique diseases). Further dataset statistics including the number of events, event start time, and event end time are shown in Table 2.

[4] We use tick library to generate Hawkes datasets
[5] https://archive.org/details/stackexchange
[6] https://github.com/srijankr/jodie/
[7] https://github.com/babylonhealth/neuralTPPs

Table 2: Dataset statistics and hyperparameters

	Statistics					Hyperparameters							
	#Marks	#Seq.	#Events	Start time	End time	Train size	Val size	Test size	Batch size	Mixture comp C	History vector size	Mark emb size	Time scale
Hawkes Ind.	2	24576	457788	0	100	14745	4915	4916	512	64	64	32	1
Hawkes Dep. (I)	2	24576	607512	0	100	14745	4915	4916	512	64	64	32	1
Hawkes Dep. (II)	5	30000	12741668	0	100	18000	6000	6000	512	64	64	32	1
MIMIC II	75	715	2419	0	6	429	143	143	64	64	64	32	1
MOOC	97	7047	396633	0	$25.73e5$	4228	1409	1410	64	64	64	64	1
Stack Overflow	22	6633	480413	$1.32e9$	$1.38e9$	3979	1327	1327	64	64	64	32	$1.e-05$

Table 3: Predictive performance of marked TPP models. NLL/time is normalized NLL score over event sequence interval. For marks, we report micro F1 score and weighted F1 score (denoted as Wt. F1 score). Bold numbers indicate the best performance. Results on the remaining datasets are provided in the Tables 5 and 6. Prop. stands for Proposed.

Model	Hawkes Dependent I (Synthetic dataset)						MOOC (Real dataset)					
	Time NLL	Mark NLL	NLL	NLL/ Time	Micro F1	Wt. F1	Time NLL	Mark NLL	NLL	NLL/ Time	Micro F1	Wt. F1
CP	57.192	17.787	74.979	0.858	53.078	45.109	348.092	100.832	448.924	0.034	30.518	28.281
RMTPP	57.301	17.117	74.418	0.851	53.273	42.261	350.847	101.094	451.941	0.034	29.951	26.958
LNM	56.081	16.948	73.029	0.845	57.062	52.517	341.494	98.897	440.391	0.031	44.528	42.757
NHP	57.416	17.265	74.681	0.851	51.901	41.908	350.015	101.307	451.322	0.033	30.023	28.184
SAHP	56.827	17.147	73.974	0.849	52.874	43.669	348.152	99.845	447.997	0.033	30.855	28.878
THP	56.744	**16.748**	73.492	0.849	53.633	45.852	347.943	98.936	446.879	0.032	31.401	30.206
Prop. RMTPP	52.795	17.018	69.813	0.806	57.063	52.621	337.743	101.983	439.726	0.031	40.425	38.153
Prop. LNM	**52.566**	16.942	**69.508**	**0.802**	**60.372**	**58.813**	**328.301**	**98.873**	**427.174**	**0.028**	**56.002**	**54.952**
Prop. THP	52.932	17.047	69.979	0.808	56.973	52.152	334.753	100.975	435.728	0.03	42.876	42.403

Table 4: Robustness of the proposed LNM model with respect to the number of mixture components C.

C	Hawkes Ind.		Hawkes Dep. (I)		Hawkes Dep. (II)		MIMIC-II		MOOC		Stack Overflow	
	Time NLL	Mark NLL	Time NLL	Mark NLL	Time NLL	Mark NLL	Time NLL	Mark NLL	Time NLL	Mark NLL	Time NLL	Mark NLL
1	48.654	11.567	54.455	16.939	-220.702	628.711	-11.844	5.926	349.01	96.269	211.657	108.403
2	47.495	11.568	53.595	16.938	-237.75	628.706	-19.478	5.757	334.442	95.438	204.965	108.204
4	47.201	11.572	53.127	16.939	-240.703	628.709	-20.591	5.977	330.769	96.093	203.775	108.445
8	47.202	11.567	53.132	16.938	-240.931	628.712	-20.475	5.938	329.379	97.765	204.543	108.553
16	47.165	11.57	53.125	16.939	-240.931	628.706	-20.377	5.838	329.485	97.881	204.784	108.816
32	47.189	11.573	53.106	16.939	-240.937	628.72	-20.635	5.741	328.766	98.325	204.442	109.767
64	47.176	11.574	53.122	16.938	-240.925	628.746	-20.464	5.903	327.396	99.283	204.567	109.86

4.2 Baseline Algorithms

Intensity-based models approximate the integral in Equation 4 using MC estimation. The event history could be encoded either using a recurrent neural network (RNN, LSTM, or GRU) or a self-attention mechanism. We compare against following state-of-the-art models (decoders) on the standard prediction task:

- Conditional Poisson **CP** is a time-independent multi-layer perceptron (MLP) based decoder.
- **RMTPP**: This is an exponential intensity-based decoder agreeing to a Gompertz distribution by [8]. Here, events are encoded using a recurrent neural network.
- **LNM**: This decoder is intensity-free log-normal mixture model by [29]. It employs RNN as an event encoder.
- **NHP**: An intensity-based multivariate decoder proposed by [21]. It uses a continuous-time LSTM encoder for event history encoding.
- **SAHP**: This model uses a self-attention mechanism for event history encoding operation as discussed in [38].
- **THP**: Transformer based model developed by [39]. It leverages the self-attention mechanism for long-term event dependency. This model is intensity-based and requires MC approximation in likelihood computation.

While SAHP and THP models use attention mechanisms for history encoding, CP, RMTPP, LNM, and NHP use recurrent encoders. Recurrent encoders take $O(N)$ time to encode an event sequence with N events, contrarily, self-attention-based encoders require $O(N^2)$ time. On one hand, CP, RMTPP, LNM, SAHP, and THP are conditionally independent models. On the other hand, NHP is a conditionally dependent model. In the proposed approach, we have two intensity-based models namely, *proposed RMTPP* and *proposed THP*, and one intensity-free model, *proposed LNM*. GRU encodes the event history in *proposed RMTPP* and *proposed LNM*. Our decoders are multivariate, intensity-free mixture (*proposed LNM*) or intensity-based attention models (*proposed THP*) where time distribution is conditioned on all possible marks.

In the following sections, we provide additional technical details of the baselines used.

Conditional Poisson (**CP**) is a simple time-independent decoder based on multi-layer perceptron (MLP). Let $\boldsymbol{h}_t$ denote the event history vector for all the events occurring before time t. CP decodes the history vector $\boldsymbol{h}_t$ into conditional intensity function $\lambda_k^*(t)$ and cumulative intensity function $\Lambda_k^*(t)$. Here, subscript k represents mark type. These functions are as follows:

$$\lambda_k^*(t) = \text{MLP}(\boldsymbol{h}_t) \text{ and } \Lambda_k^*(t) = \text{MLP}(\boldsymbol{h}_t)(t - t_i), \tag{21}$$

where, t_i is the arrival time of the event occurring just before time t.

RMTPP is an exponential intensity-based unimodal decoder agreeing to a Gompertz distribution and is proposed by [8]. RMTPP is a conditionally independent decoder. Here, the conditional intensity and cumulative intensity are formulated as follows:

$$\lambda_k^*(t) = \exp\left(\boldsymbol{W}_1\boldsymbol{h}_t + w_2(t - t_i) + \boldsymbol{b}_1\right)_k \tag{22}$$

$$\Lambda_k^*(t) = \frac{1}{w_2}\left[\exp\left(\boldsymbol{W}_1\boldsymbol{h}_t + \boldsymbol{b}_1\right) - \exp\left(\boldsymbol{W}_1\boldsymbol{h}_t + w_2(t - t_i) + \boldsymbol{b}_1\right)\right]_k, \tag{23}$$

where, $\boldsymbol{W}_1$, w_2, and $\boldsymbol{b}_1$ are learnable parameters of the neural network and $\boldsymbol{W}_1 \in \mathbb{R}^{|\boldsymbol{h}_t| \times K}$, $w_2 \in \mathbb{R}$ and $\boldsymbol{b}_1 \in \mathbb{R}^K$. Note that, K represents the total number of marks and k represents the mark type.

LNM is an intensity-free log-normal mixture decoder proposed by [29]. LNM is a conditionally independent decoder that models PDF of inter-event time as follows:

$$f(\tau|\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{s}) = \sum_{c=1}^{C} w_c \frac{1}{\tau s_c \sqrt{2\pi}} \exp\left(-\frac{(\log \tau - \mu_c)^2}{2s_c^2}\right), \tag{24}$$

where, $\boldsymbol{w}$ are the mixture weights, $\boldsymbol{\mu}$ are the mixture means and $\boldsymbol{s}$ are the mixture standard deviations. Here, number of mixture component are represented by $c \in \{1, 2, \ldots, C\}$. The parameters $\boldsymbol{w}, \boldsymbol{\mu}$ and $\boldsymbol{s}$ are estimated from $\boldsymbol{h}$ as follows:

$$\boldsymbol{w} = \text{softmax}(\boldsymbol{W}_{\boldsymbol{w}}\boldsymbol{h} + \boldsymbol{b}_{\boldsymbol{w}}) \tag{25}$$

$$\boldsymbol{\mu} = \exp(\boldsymbol{W}_{\boldsymbol{\mu}}\boldsymbol{h} + \boldsymbol{b}_{\boldsymbol{\mu}}) \text{ and } \boldsymbol{s} = \boldsymbol{W}_{\boldsymbol{s}}\boldsymbol{h} + \boldsymbol{b}_{\boldsymbol{s}}, \tag{26}$$

where, $\{\boldsymbol{W}_{\boldsymbol{w}}, \boldsymbol{W}_{\boldsymbol{\mu}}, \boldsymbol{W}_{\boldsymbol{s}}, \boldsymbol{b}_{\boldsymbol{w}}, \boldsymbol{b}_{\boldsymbol{\mu}}, \boldsymbol{b}_{\boldsymbol{s}}\}$ are the learnable parameters of the neural network. Note that the LNM model does not condition time distribution on marks and shares the same drawbacks as that of conditionally independent models.

For **NHP, SAHP, and THP** models, we use the parametrized intensity functions specified in the papers [21, 38, 39] respectively. We condition these formulation on marks to obtain conditionally dependent TPP as indicated in Equation 2.

4.3 Evaluation Protocols

To quantify the predictive performance of TPP models, we use the NLL score metric as shown in Equation 14. Different event sequences could be defined over different time intervals, therefore, we report NLL normalized by time (NLL/time) score. Additionally, as datasets used are multi-class with class imbalance, we report micro F1 score (accuracy) and weighted F1 score for marks. Ideally, a model should perform equally well on all metrics.

4.4 Training and Results

Our experimentation code and datasets are available on GitHub[8]. For all datasets, we use 60% of the sequences for training, 20% for validation and rest 20% for test. We train the proposed model by minimizing the NLL score (Equation 14). For a fair comparison, we try out different hyperparameter configurations on the validation split. Using the best set of hyperparameters, we evaluate performance on the test split. The train set size, validation set size, and test set size along with the best set of hyperparameters for each dataset are given in Table 2. Each dataset is defined on a different time scale. For example, start time and end time in the Stack Overflow dataset are in the order of $1e9$. Thus, for numerical stability, many methods scale the time values with the appropriate time scale. As different event sequences have different lengths, we employ batch-level padding on arrival times and event marks to match the batch dimensions. We use zeros as padding values. We minimize the NLL in the training using Adam optimizer [17]. The learning rate used for all experiments is $1e-3$ with Adam optimizer regularization decay of $1e-5$. We use early stopping in the training with the patience of 50. We see the performance of the model on the validation set and choose the best model. Finally, we report metrics on the test set.

[8] https://github.com/waghmaregovind/joint_tpp

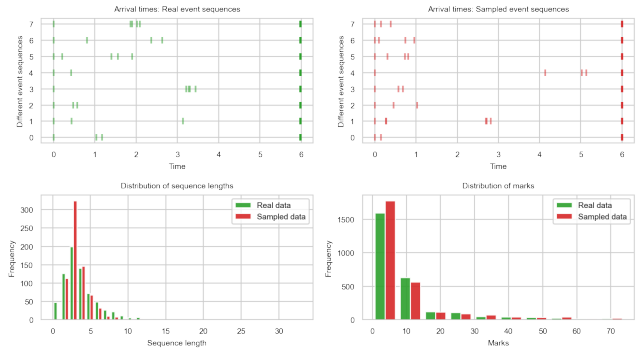

Figure 3: Sampling statistics for MIMIC-II dataset.

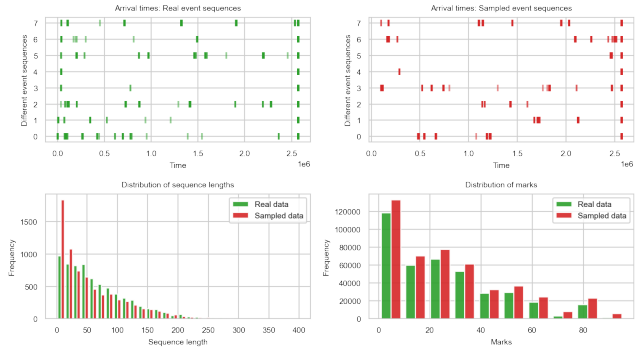

Figure 4: Sampling statistics for MOOC dataset.

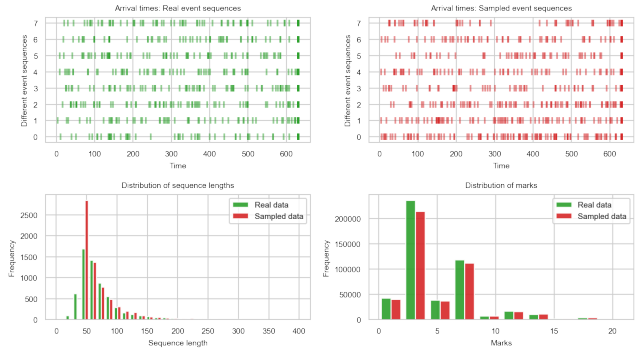

Figure 5: Sampling statistics for Stack Overflow dataset.

The training procedure for the proposed model involves three key steps as shown in Figure 2. These are input representation, event history encoding, and distribution modeling. In the first step, arrival time is converted into inter-event time. The categorical marks are converted into fixed embedding through the mark embedding layer. As different datasets have a different number of marks, we adjust the size of mark embedding accordingly. In the second step, the input representation obtained for all $i-1$ events $(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{i-1})$ is passed through RNN to obtain fixed dimensional history vector $\boldsymbol{h}_i$ for the i^{th} event. The dimension of the mark embedding and history vector is shown in Table 2 as history vector size. Using this history vector $\boldsymbol{h}_i$ we model the distribution of inter-event time for all mark types in the final step.

The predictive performance of the proposed models is shown in Table 3, 5, and 6. We also provide a breakdown of NLL score into time NLL and mark NLL in the same table to quantify the inter-dependency of time and marks. As emphasized before, *A marked TPP model is considered better if it performs well on all the metrics.* The proposed conditionally dependent models show better predictive performance compared to conditionally independent models. All conditionally independent models show similar predictive performance on marks. It is mainly due to the structural design limitation of conditionally independent models. The *proposed LNM* (conditionally dependent) decoder and the LNM decoder are mixture models. Mixture models have universal approximation property to approximate any multimodal distribution [29]. Due to independence, the LNM mixture model performs poorly compared to conditionally dependent models. In conditionally dependent models, *proposed LNM* model shows superior performance on nearly all the metrics. *Proposed RMTPP* and *proposed THP* models use multivariate intensity-based formulation shown in the Equation 2. The likelihood in the training objective does not have a closed-form and requires MC estimates. MC approximation for Equation 2 is slower and less accurate. Hence, the approximation involved in the likelihood computation is a bottleneck for the predictive performance of the TPPs. As *proposed LNM* model is a conditionally dependent mixture model, we evaluate the likelihood in closed-form. It makes the *proposed LNM* model more flexible and accurate than other conditionally dependent models, as observed in Table 3, 5 and 6.

Average event sequence length, number of marks, and mark class distribution play a crucial role in the predictive performance of the marked TPP models (see Table 2 for statistics). For MIMIC-II, the average sequence length is four. Thus, all models show high variation in the metrics on different data splits. RMTPP performs competitively on simple datasets like Hawkes Ind. and Hawkes Dep. (I) but fails to perform on a dataset with a larger number of marks and longer event sequences. In Table 3, we closely observe impact made by our multivariate TPP model on NLL score. We observe significant improvement in time NLL score as time distribution is conditioned on each mark. Improvement in the time NLL improves marker classification metrics. For conditionally independent models, mark class is inferred as per the Equation 12 and for conditionally dependent models marks class is inferred using Equation 13. MOOC dataset contains interactions of learners with the online courses. Here, the event sequence represents the time-evolving course journey of the learner. Marks represent different activities performed towards course completion. It contains entangled time and marks, and conditionally independent models fail to capture this relationship. The number of marks in the MOOC dataset is 97. Thus, the intensity-based model numerically approximates 97 such function using MC estimates. On MOOC, the *proposed LNM*, conditionally dependent mixture model shows boost of 11.5% on micro F1 score and 12.2% on weighted F1 score in mark prediction compared to next best model (refer Table 3). It is mainly due to the intensity-free modeling of inter-event time PDF and multivariate formulation. The proposed models consistently outperform other baselines in time and marker prediction tasks on all datasets.

Table 5: Predictive performance of marked TPP models on synthetic datasets Hawkes independent and Hawkes dependent II. Bold numbers indicate the best performance. Prop. stands for Proposed.

Model	Hawkes Independent						Hawkes Dependent II					
	Time NLL	Mark NLL	NLL	NLL/ Time	Micro F1	Wt. F1	Time NLL	Mark NLL	NLL	NLL/ Time	Micro F1	Wt. F1
CP	51.304	12.058	63.362	0.726	62.271	48.474	-227.08	647.219	420.139	4.22	31.51	21.269
RMTPP	50.625	11.629	62.254	0.723	63.261	48.783	-239.102	646.725	407.623	4.108	32.429	22.691
LNM	50.073	11.571	61.644	0.716	67.081	56.884	-234.089	628.723	394.634	3.978	32.87	24.702
NHP	50.598	12.66	63.258	0.728	61.927	48.137	-239.38	645.472	406.092	4.143	32.023	22.539
SAHP	50.461	12.473	62.934	0.724	62.398	48.528	-236.875	642.701	405.826	4.185	32.028	22.866
THP	50.584	12.31	62.894	0.723	63.504	48.328	-238.276	639.813	401.537	4.24	32.146	22.831
Prop. RMTPP	47.129	11.72	58.849	0.681	66.99	57.169	**-249.928**	650.631	400.703	4.038	32.742	24.315
Prop. LNM	**46.676**	**11.571**	**58.247**	**0.675**	**70.802**	**65.248**	-240.915	**628.719**	**387.804**	**3.909**	**33.175**	**26.256**
Prop. THP	47.061	11.691	58.752	0.68	67.128	56.885	-245.588	639.305	393.717	3.955	32.721	24.367

Table 6: Predictive performance of marked TPP models on real datasets Stack Overflow and MIMIC-II. Bold numbers indicate the best performance. Prop. stands for Proposed.

Model	Stack Overflow						MIMIC-II					
	Time NLL	Mark NLL	NLL	NLL/ Time	Micro F1	Wt. F1	Time NLL	Mark NLL	NLL	NLL/ Time	Micro F1	Wt. F1
CP	218.612	118.947	337.559	0.576	43.656	30.179	-18.377	5.948	-12.429	-25.497	56.461	53.842
RMTPP	224.172	117.659	341.831	0.581	43.842	28.241	-18.826	5.952	-12.874	-27.595	35.126	24.787
LNM	208.131	111.137	319.268	0.534	46.451	32.925	-18.703	5.927	-12.776	-26.725	66.208	63.668
NHP	217.685	114.887	332.572	0.561	44.439	30.203	-17.53	6.595	-10.935	-26.332	58.274	52.125
SAHP	216.995	112.686	329.681	0.557	45.139	30.892	-18.384	7.089	-11.295	-27.819	59.492	53.502
THP	215.093	112.544	327.637	0.551	45.459	31.861	-19.73	6.511	-13.219	-28.121	60.724	54.146
Prop. RMTPP	220.284	119.779	340.063	0.575	45.271	29.68	-21.883	5.874	-16.009	-34.318	33.998	23.836
Prop. LNM	**204.344**	**110.147**	**314.491**	**0.531**	**47.885**	**34.364**	-21.761	**5.849**	-15.912	-33.548	**66.61**	**63.725**
Prop. THP	211.755	113.103	324.858	0.548	46.897	33.294	**-22.365**	6.013	**-16.352**	**-34.893**	59.528	53.221

In the *proposed LNM* model, for all datasets, we have used the number of mixture components as 64. This value is suggested by [29], which is equivalent to a number of parameters in the single-layer model proposed by [25]. We also provide the sensitivity of NLL metrics with respect to the number of mixture components, C, in Table 4. Empirically, the proposed mixture model is robust to the different values of C. For the *proposed LNM* model, the NLL function does not contain any integration term as inter-event time PDF is modeled using mixture models. Therefore, we leverage mixture models to estimate likelihood in closed-form. In closed-form sampling, we first sample the categorical mark distribution. Using this sampled mark of type $m_i = k$, we select the time PDF $f_k(\tau|\boldsymbol{w}_k, \boldsymbol{\mu}_k, \boldsymbol{s}_k)$. Further, we sample from this PDF to get the next inter-event time τ_i of mark type k. To evaluate sampled event sequences qualitatively, we plot arrival times, distribution of event sequence lengths, and distribution of marks for each dataset. Sampling analysis for real-world datasets is shown in Figures 3, 4, and 5. The total NLL score consists of the time NLL component of continuous inter-event time and the mark NLL component of categorical marks. Both these components play a key role in model training and influence future predictions. In Table 3, we provide a breakdown of the NLL score for all the models. The proposed conditionally dependent models show better time NLL and the mark NLL value due to multivariate modeling.

5 LIMITATIONS AND CONCLUSION

Conditionally dependent models use multivariate formulation to condition inter-event time distribution on the set of categorical marks. If the number of marks K is extremely large, mark prediction becomes an extreme class classification problem. To address this, [12, 22] have proposed noise-contrastive-estimation-based models.

In this work, we discuss the adverse effect of the independence assumption between time and mark on the predictive performance of the marked TPPs. We address this structural shortcoming by proposing a conditionally dependent multivariate TPP model under both intensity-based and intensity-free settings. The *proposed LNM* architecture overcomes the drawbacks of an intensity-based conditionally dependent model and poses desired properties like closed-form likelihood, and closed-form sampling. Multiple evaluation metrics on diverse datasets highlight the impact of our work against state-of-the-art conditionally dependent and independent marked TPP models.

REFERENCES

[1] Emmanuel Bacry, Adrian Iuga, Matthieu Lasnier, and Charles-Albert Lehalle. 2015. Market impacts and the life cycle of investors orders. *Market Microstructure and Liquidity* (2015).

[2] Marin Biloš, Bertrand Charpentier, and Stephan Günnemann. 2019. Uncertainty on Asynchronous Time Event Prediction. In *NeurIPS*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.).

[3] E. Brockmeyer, H.L. Halstrøm, A. Jensen, and A.K. Erlang. 1948. *The Life and Works of A.K. Erlang*. Academy of Technical sciences, Vol. 2.

[4] Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. 2021. Neural Spatio-Temporal Point Processes. In *ICLR*.

[5] Harald Cramér. 1969. Historical review of Filip Lundberg's works on risk theory. *Scandinavian Actuarial Journal* (1969).

[6] Daryl J Daley and David Vere-Jones. 2007. *An introduction to the theory of point processes: volume II: general theory and structure*.

[7] Prathamesh Deshpande, Kamlesh Marathe, Abir De, and Sunita Sarawagi. 2021. Long Horizon Forecasting With Temporal Point Processes. *WSDM* (2021).

[8] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *ACM SIGKDD KDD*.

[9] R. Engle and Jeffrey R. Russell. 1998. Autoregressive Conditional Duration: A New Model for Irregularly Spaced Transaction Data. *Econometrica* (1998).

[10] Joseph Enguehard, Dan Busbridge, Adam Bozson, Claire Woodcock, and Nils Hammerla. 2020. Neural Temporal Point Processes For Modelling Electronic Health Records. In *ML4H*.

[11] Mehrdad Farajtabar, Nan Du, Manuel Gomez Rodriguez, Isabel Valera, Hongyuan Zha, and Le Song. 2014. Shaping social activity by incentivizing users. *NeurIPS* (2014).

[12] Ruocheng Guo, Jundong Li, and Huan Liu. 2018. INITIATOR: Noise-contrastive Estimation for Marked Temporal Point Process. In *IJCAI*.

[13] Alan G. Hawkes. 1971. Point Spectra of Some Mutually Exciting Point Processes. *Journal of the Royal Statistical Society. Series B* 33, 3 (1971). http://www.jstor.org/stable/2984686

[14] Alan G. Hawkes and David Oakes. 1974. A Cluster Process Representation of a Self-Exciting Process. *Journal of Applied Probability* (1974).

[15] Valerie Isham and Mark Westcott. 1979. A self-correcting point process. *Stochastic Processes and their Applications* (1979).

[16] Sharma Karishma, Zhang Yizhou, Ferrara Emilio, and Liu Yan. 2021. Identifying Coordinated Accounts on Social Media through Hidden Influence and Group Behaviours. In *ACM SIGKDD KDD*.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[18] J.F.C. Kingman. 1992. *Poisson Processes*.

[19] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *ACM SIGKDD KDD*.

[20] Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. 2018. Learning temporal point processes via reinforcement learning. *NeurIPS* (2018).

[21] Hongyuan Mei and Jason Eisner. 2017. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In *NeurIPS*.

[22] Hongyuan Mei, Tom Wan, and Jason Eisner. 2020. Noise-Contrastive Estimation for Multivariate Point Processes. In *NeurIPS*.

[23] Yosihiko Ogata. 1998. Space-Time Point-Process Models for Earthquake Occurrences. *Annals of the Institute of Statistical Mathematics* (1998).

[24] Maya Okawa, Tomoharu Iwata, Takeshi Kurashima, Yusuke Tanaka, Hiroyuki Toda, and Naonori Ueda. 2019. Deep Mixture Point Processes. *KDD* (2019).

[25] Takahiro Omi, naonori ueda, and Kazuyuki Aihara. 2019. Fully Neural Network based Model for General Temporal Point Processes. In *NeurIPS*.

[26] C. Palm. 1943. *Intensitätsschwankungen im Fernsprechverkehr*.

[27] Jakob Gulddahl Rasmussen. 2011. Temporal point processes: the conditional intensity function. *Lecture Notes, Jan* (2011).

[28] Manuel Rodriguez, David Balduzzi, and Bernhard Schölkopf. 2011. Uncovering the Temporal Dynamics of Diffusion Networks. In *ICML*.

[29] Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. 2020. Intensity-Free Learning of Temporal Point Processes. *ICLR* (2020).

[30] Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. 2021. Neural Temporal Point Processes: A Review. In *IJCAI*.

[31] Ali Caner Türkmen, Yuyang Wang, and Alex Smola. 2019. FastPoint: Scalable Deep Point Processes. In *ECML PKDD*.

[32] Utkarsh Upadhyay, Abir De, and Manuel Gomez-Rodrizuez. 2018. Deep Reinforcement Learning of Marked Temporal Point Processes. In *NeurIPS*.

[33] Qitian Wu, Chaoqi Yang, Hengrui Zhang, Xiaofeng Gao, Paul Weng, and Guihai Chen. 2018. Adversarial Training Model Unifying Feature Driven and Point Process Perspectives for Event Popularity Prediction. In *CIKM*.

[34] Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. 2017. Wasserstein Learning of Deep Generative Point Process Models. In *NeurIPS*.

[35] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. 2017. Modeling the intensity function of point process via recurrent neural networks. In *AAAI*.

[36] Junchi Yan, Xin Liu, Liangliang Shi, Changsheng Li, and Hongyuan Zha. 2018. Improving Maximum Likelihood Estimation of Temporal Point Process via Discriminative and Adversarial Learning. In *IJCAI*.

[37] Zhang Yizhou, Sharma Karishma, and Liu Yan. 2021. VigDet: Knowledge Informed Neural Temporal Point Process for Coordination Detection on Social Media. In *NeurIPS*.

[38] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. 2020. Self-attentive hawkes process. In *ICML*.

[39] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer hawkes process. In *ICML*.