Hindawi
Complexity
Volume 2022, Article ID 1616116, 21 pages
https://doi.org/10.1155/2022/1616116

Research Article

A Comparative Study of Some Point Process Models for Dynamic Networks

S. Haleh S. Dizaji, Saeid Pashazadeh, and Javad Musevi Niya

Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

Correspondence should be addressed to Saeid Pashazadeh; pashazadeh@tabrizu.ac.ir

Received 8 March 2022; Revised 15 July 2022; Accepted 3 August 2022; Published 16 September 2022

Academic Editor: Guilherme Ferraz de Arruda

Modeling dynamic networks has attracted much interest in recent years, which helps understand networks' behavior. Many works have been dedicated to modeling discrete-time networks, but less work is done for continuous-time networks. Point processes as powerful tools for modeling discrete events in continuous time have been widely used for modeling events over networks and their dynamics. These models have solid mathematical assumptions, making them interpretable but decreasing their generalizability for different datasets. Hence, neural point processes were introduced that don't have strong assumptions on generative functions. However, these models can be impractical in the case of a large number of event types. This research presents a comparative study of different point process (Hawkes) models for continuous-time networks. Furthermore, a previously introduced neural point process (neural Hawkes) model is applied for modeling network interactions. In this work, network clustering is used for specifying interaction types. These methods are compared using different synthetic and real-world datasets, and their efficiency is evaluated on these datasets. The experiments represent that each model is appropriate for a group of datasets. In addition, the effect of clustering on results is discussed, and experiments for different clusters are presented.

1. Introduction

Networks are ubiquitous everywhere in natural life, such as human and biological networks and chemical reactions, digital life, such as social networks, and technology, such as transportation networks and computer networks. These networks usually have dynamic structures and contain continuous-time events occurring through them. These events include interactions between individuals in a network, such as contacts in human networks or message passing between members of a social network, and protein-protein interactions in a biological system. These dynamics inside a network can reveal exciting information about the network and individuals, such as the roles of the individuals or groups in the network and the unknown structure of the network which can be dynamic. Although dynamic networks have existed for a long time, studying them and considering time as an important factor have become recently attractive [1].

Different temporal granularities can be considered for dynamic networks. Accordingly, different representation models were introduced, which are (i) static networks, (ii) weighted networks, (iii) discrete-time, and (iv) continuous-time networks [2]. Static networks are the simplest representations with the most coarse-grained granularity and contain no dynamics. On the other side, continuous-time networks are the most complex and precise representation methods [3]. In this research, continuous-time networks are studied, requiring more accurate modeling methods.

In the continuous-time networks studied in this research, where only dynamics for edges are assumed (nodes are assumed to remain constant), network events, i.e., edge appearance or removal, occur randomly. These networks vary according to link duration, which are distinguished in some literature [3, 4]. Accordingly, the networks can have instantaneous (or short-length) or long-duration links [3]. The former networks are considered in this research including interaction networks such as e-mail networks and temporal networks, e.g., human proximity networks. These networks have higher dynamicity compared with the other group of networks; hence they are more complex, and more accurate methods are required for analyzing them.

Modeling dynamic networks has attracted many researchers and provided valuable information about the network under study, such as anticipating the future behavior of the network and its dynamic properties. As was mentioned, modeling continuous-time networks demands higher computational complexity. Several continuous-time models have been proposed, such as Stochastic Actor Oriented Models (SAOM) [5–9], Relational Event Models (REM) [10–14], point process models [15–20], and deep learning models [3, 21–23]. REMs and SAOMs are statistical methods where defined statistics are used for modeling network interactions. In REM methods, the intensity of the occurrence of relations (events) depends on the events' history and different covariates. In SAOM models, ties (relations) are actor-oriented, and actors (nodes) select a tie to maximize their utility function, depending on various defined statistics. Point process models that are explained in the follows, define functional dependencies of events on the history of events. On the other hand, deep learning methods use non-functional dependencies to avoid function misspecification. Furthermore, point process models can benefit from the non-functional property of deep learning methods by combining these methods, which is the basis of the method proposed in this research [24, 25].

Point processes are strong models for modeling continuous-time events that have been used for different real-world events. These processes are characterized by their particular intensity functions, which are appropriate for different phenomena. The possibility of defining intensity functions gives a high interpretability power to these models. Some examples of different functional forms are (i) homogeneous and nonhomogeneous Poisson processes, (ii) self-exciting (Hawkes) processes, and (iii) survival processes. Hawkes processes [26] are the most appropriate models for modeling interdependent events. This process that was used for modeling earthquakes [27, 28], has also been applied for modeling dynamic networks in papers [15, 16]. This model which clusters events through time, considers the effect of history on the occurrence of future events. Hence it can capture inter-dependency between consecutive events in networks. Furthermore, marked point processes are appropriate models for including types of events in addition to their times, which is important for the events inside networks.

Although these models have been successful in some applications, due to their strong assumption of the generative function, they might not be suitable for different situations (because of different characteristics of datasets that yield different generative processes). Neural networks have been successful models for tackling these problems, which can extract dataset features and devise models without any previous assumption about their underlying processes. Hence point process models have benefited from the power of neural networks for parameterizing their intensity functions, where it can be fully or partially modeled using these networks.

Recurrent Neural Networks (RNN) have recently successfully modeled point processes because of their success in modeling time-series events. Hence in the current research, an RNN (more specifically a Long Short Term Memory or LSTM) model based on the method introduced in the paper [25] has been used for improving point process models and applied for modeling dynamic interactions in networks. This model inherits some interpretability power of point process models (Hawkes process) and the accuracy of deep learning models.

In order to use marked point processes for network interactions, an appropriate definition of event types is required. Since considering every node pair as a distinct type is not efficient and applicable in most situations (there will be in the order of N^2 types for a network with N nodes). Hence, in this research, similar to the works of papers [15, 16] events are specified by clustering nodes of the network, which changes the scale of the network from node pairs to cluster pairs. By using the clustering method, neural network models are applicable for modeling network interactions. However, selecting an appropriate number of clusters is challenging. Increasing the number of clusters will result in a more accurate model but will increase the model complexity and requires more processing time. Hence, this trade-off should be considered in selecting an appropriate number of clusters.

In a nutshell, the contributions of this work can be summarized as follows: (i) Different point process (Hawkes) models are compared using synthetic and real-world datasets. (ii) A recurrent neural network-based point process model has been used for modeling dynamic interactions in networks. (iii) The clustering algorithm is applied for obtaining event types, and the effect of different numbers of clusters on the accuracy of the interactions model has been examined.

2. Related Work

Several studies have been dedicated to modeling continuous-time events inside networks, including interaction models. In this section, some of these models and also event-based methods are introduced.

2.1. Continuous-Time Dynamic Network Models. According to our knowledge, few works have been dedicated to the continuous-time modeling of networks. Some of these models are relational event models (REMs), stochastic actor oriented models (SAOMs), stochastic block models (SBMs) for continuous-time, point process (Hawkes) models, and deep learning models. The following explains these models.

REMs consider the interaction of the nodes in a network as relational events, where the intensity of each interaction can depend on the events' history and the interaction time [10, 11]. In [11], events with exact time and ordering information are modeled, where the intensity of events is assumed to be piece-wise constant. Paper [12] uses this method to model times of events and a normal distribution to model weights of relations. Leenders et al. in [13] used

REM to analyze team members' behaviors and used various statistics to formulate the intensity function. In [14], REM is combined with stochastic block models [29], where distinct clusters which are specified with the Chinese Restaurant Process (CRP), representing different dynamic processes. This clustering method provides flexibility for the method in the number of clusters. In this model, the intensity function is modeled using relational event models and considers constant intensities between consecutive events.

In addition to discrete-time SBMs, some continuous-time models have also been introduced, such as in [30, 31]. Papers [30, 31] model interaction duration in continuous time using block structures. In [30], which also has some properties of SAOMs, lengths of interactions (and non-interactions) between nodes are modeled using exponential distributions where their parameters depend on the cluster memberships of nodes (as in SBM).

In [5], SAOM was introduced. In this method, network evolution is modeled as actions taken by users using a continuous-time Markov chain process. Actors (users) in this model make actions that maximize a random utility function and a fixed utility that is a function of several statistics. Also in [6], interactions are considered as a function of network structure and attributes of nodes and pairs of nodes. Network structure effects include appearing links between friends of friends or triadic closure and effects of nodes' degrees. In [7], SAOM is modified for modeling undirected networks, where both nodes in a relationship should agree to create a link. In this model (DyNAM), the objective function is composed of generic, signed, weighted, and windowed effects. Stadtfeld et al. in [8] extend DyNAM model for directed links of relational events. In [9], time heterogeneity of SAOMs is studied and a method for testing the heterogeneity of a model is introduced.

The point process, specifically the Hawkes process, has been used for modeling continuous-time interactions inside networks. In [19], directed interactions between nodes are modeled using the Cox intensity model and using static and dynamic covariates in the intensity function (similar to REM method [11]). This paper also models multiplicity in interactions. Fox et al. in [17] proposed three Hawkes models including (i) a Hawkes process with constant background intensities for sending emails, (ii) a Hawkes model with nonstationary background intensities for considering weekends and nights and other effects, and (iii) model (ii) with different e-mail response rates between different users. They used these models to find leaders in the network. In [18], Hawkes process is utilized for modeling network interactions, and also missing information (sender, receiver, or both) is estimated. In [15], latent clusters with specific dynamic processes (Hawkes process) are considered inside the network and interactions between nodes of every cluster pair are considered dependent on each other. In paper [16], this dependency is relaxed, and nodes inside each cluster pair interact independently, sharing the same parameters. This property makes the model tractable and scalable. In [20], the mutual point process (Hawkes process) and latent space model are used to model the intensity of interactions. Although these models have interpretable intensity functions, considering a predefined generative model for interactions might cause them not to be applicable for more datasets.

In order to mitigate the aforementioned problem of point processes, the neural network implementation of these models was introduced, where the specification of the generative process is not required. Graph neural networks (GNNs) as neural networks for encoding network information have recently become popular and dynamic GNNs which model dynamic networks also were introduced that are reviewed in [3, 21]. In [32], inspired by the neural Hawkes model of Mei and Eisner [25], a neural network model of point processes is proposed for modeling interactions in knowledge graphs. In this model, every pair of nodes is considered to have separate dynamic processes and the history of events is reduced to relevant history, enabling it to consider more event types than the original model [25]. However, limiting the history of events might not be appropriate for some situations. In addition to intensity-based models, generative models have also been successful in modeling dynamic networks. In [22], generative adversarial networks (GAN) and RNNs have been used for graph reconstruction, link prediction, and graph prediction. Ma et al. in [33] propose a dynamic GNN model which uses LSTM architecture for encoding nodes according to interactions. This model is used for link prediction and node classification tasks.

2.2. Continuous-Time Network Event Models.

Several point process based models were proposed for modeling continuous-time events over networks such as diffusion processes. In [34], a coevolutionary model based on point processes (Hawkes and survival processes) is proposed, which models network structure dynamics and diffusion processes as interdependent processes. In neural point processes (in particular recurrent neural network (RNN) based models), some features of Hawkes processes are preserved such as the dependency of intensity function on the history of events. In [35], a joint model of event sequences and time-series data is proposed, which uses two Long Short Term Memory (LSTM) networks for each kind of input data. In this model, inspired by the Hawkes process, influences between different event types are modeled using an attention layer. In [25], an RNN (LSTM) based Hawkes process is proposed, introducing a continuous-time LSTM for modeling events. This model consists of a discrete-time LSTM and a continuous exponential decay of memory cells (and hidden states) which resembles the exponential decay of Hawkes intensity. This model was used for modeling diffusion processes over networks. Because of the dependency of the number of parameters of this model on the number of event types, it is not appropriate for events with a vast number of types (e.g., interactions in a rather large network considering every node pair as a different type).

In the current research, the model introduced in [25] has been used for modeling network interactions by clustering nodes of the network to reduce the number of event types. This model is presented in the following sections.

3. Dynamic Network Models

In this section, different continuous-time dynamic network models which are evaluated in our research are explained. In addition, our proposed dynamic network model is introduced. First in Sections 3.1 and 3.2, continuous-time dynamic networks assumed in this research and (marked) point process models are explained. Then in Section 3.3, the considered point process models for dynamic networks, which are evaluated in this research, are explained. Finally, in Section 3.4, the proposed neural point process model is explained.

3.1. Continuous-Time Networks. In continuous-time networks, interactions between nodes inside a network, considered as events, can be represented as a series of occurrence times. Let G represent a network containing N number of nodes that are the individuals interacting with each other and are assumed to remain constant. The edges (instantaneous links) between these nodes represent discrete events in continuous time, which are represented as $I_{ij} = \{t_m\}_1^M$, where I_{ij} represents interactions between nodes i and j, t_m is the occurrence time of m-th event between them in the time interval $[0, T]$, and M is the total number of events (interactions) between these nodes. This network is considered an undirected network. A sample of this representation is illustrated in Figure 1.

3.2. Marked Point Processes. Point processes represent random processes which are a sequence of events $\xi = \{t_1, \ldots, t_n\}$ in continuous time in a given interval $[0, T]$. This model is characterized by conditional intensity function $\lambda^*(t)$ which is the conditional probability of occurring an event i at time t given History $\mathscr{H}(t) = \{t_1, \ldots, t_j\}$, which represents events before event i. This intensity function is defined as in [36]

$$\lambda^*(t) = \frac{f^*(t)}{1 - F^*(t)}, \tag{1}$$

where $f^*(t)$ represents the conditional density function and $F^*(t)$ is its cumulative distribution function. Following this definition, the intensity function can also be interpreted as the expected number of events in a small interval $[t, t + dt)$ given history $\mathscr{H}(t)$, i.e., [36]

$$\lambda^*(t)\mathrm{d}t = \mathbb{E}[N(\mathrm{d}t)|\mathscr{H}(t)]. \tag{2}$$

Marked (multivariate) point processes are generalizations of (univariate) point processes where types of events are also considered in addition to the times. These processes are represented in the form of $\xi = \{(k_1, t_1), \ldots, (k_n, t_n)\}$, where k_i is the type of event i. The conditional intensity function is dependent on the type and the time of events. This function is formulated in [36] as

$$\lambda_k^*(t) = \lambda^*(t) f^*(k|t), \tag{3}$$

where $\lambda^*(t)$ is the ground intensity which is the same as the unmarked case, but it can also depend on the marks of previous events. $f^*(k|t)$ is the conditional density function of type k. In the method proposed in this research, interaction events in a network are considered as marked point processes, where the types of events represent the clusters (blocks) to which the interacting nodes belong. This process is explained in more detail in this section.

Different kinds of point processes are defined depending on the intensity function formula. The most general processes are Poisson and Hawkes processes, which are defined as the following.

3.2.1. Poisson Process (Nonhomogeneous). In the nonhomogeneous Poisson process, the conditional intensity function is independent of the history of events and is only dependent on the time, i.e., $\lambda^*(t) = \lambda(t)$ [36].

3.2.2. Multivariate Self-Exciting (Hawkes) Process. In the multivariate Hawkes process, the dependency between events is modeled using the equation:

$$\lambda_k^*(t) = \mu_k + \sum_{i:t_i < t} \alpha_{k_i,k} \exp\left(-\beta_{k_i,k}(t - t_i)\right), \tag{4}$$

where μ_k represents the base intensity of events of type k, $\alpha_{k_i,k}$ is the influence of an event of type k_i on an event of type k, and $\beta_{k_i,k}$ is the corresponding decay rate. With the assumption of positive α values, the occurrence of every event increases the intensity functions, but they decay exponentially towards μ_k [25].

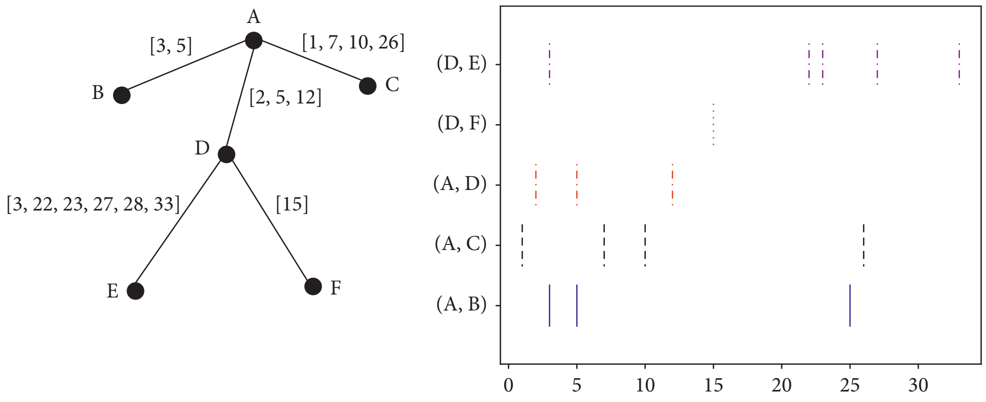
The parameters of these models are estimated using the maximum likelihood method. In this method, log-likelihood of train events (in an interval of $[0, T]$) is calculated using

$$\mathscr{L} = \sum_{i:t_i < T} \log \lambda_{k_i}(t_i) - \int_{t=0}^{T} \lambda(t)\mathrm{d}t. \tag{5}$$

Then model parameters can be inferred by maximizing the log-likelihood function. Maximization can be performed using methods such as Expectation-Maximization (EM) and Bayesian inference.

3.3. Hawkes-Based Dynamic Network Models. Since Hawkes point processes can capture dependency between events during the time, they have been used for modeling interactions in networks in different studies. Some of these models which are evaluated in this research are (i) vanilla Hawkes model, (ii) block point process model (BPPM) [15], and (iii) community Hawkes independent pairs (CHIP) model [16]. These models are explained in this section.

3.3.1. Vanilla Hawkes Model (Hawkes). In this model, all node pairs are considered to have independent and the same process (i.e., every node pair interacts independently but share the same parameters). In other words, the intensity function for the events between every two nodes is formulated as the following equation:



(a) (b)

Figure 1: Continuous-time representation of a dynamic network. (a) Sample dynamic network. (b) Continuous-time representation.

$$\lambda^{*}(t) = \mu + \alpha \sum_{i:t_i < t} \exp\left(-\beta\left(t - t_i\right)\right). \tag{6}$$

The parameters μ, α, and β are constant among all node pairs, and the intensity function for each node pair only depends on the events between them. The parameters of this model are estimated using the EM algorithm [37], which is explained in more detail in the Appendix section.

3.3.2. Block Hawkes Model. The BPPM introduced in [15] is a community-based Hawkes model for modeling events of networks. In this model, the nodes of each block (cluster) pair share the same parameters, but the events between every node pair of each block pair are dependent on each other. The inference procedure is performed using an interior point optimization routine [15, 38]. Class memberships (clusters) are obtained using local search and variational inference methods. These classes are initialized using the spectral clustering algorithm.

3.3.3. CHIP Model. The CHIP model [16] is based on the BPPM [15], but in this model the dependency assumption between events of each node pair inside block pairs is relaxed. This property makes this model more scalable and reduces the complexity of its analysis. The inference procedure is performed using the moments estimator method followed by likelihood maximization using a standard scalar optimization or line search method. In this method, also the clusters are obtained using the spectral clustering algorithm. Besides, the number of clusters is calculated using held-out data by selecting the number of clusters that maximizes the likelihood of this data.

3.4. Proposed Method. Point processes have strong assumptions for intensity function and, as mentioned in [25], have some restrictions, such as positive effects of the history on intensity function. Hence, they cannot be generalized to many real-world datasets. Therefore, in the current research, a neural point process model based on the model introduced in [25] is proposed, which uses clustering for specifying event types and reducing them. This model is explained in this section.

3.4.1. Proposed Block Neural Point Process Model. The neural network-based point process model applied in this research is based on the method introduced in [25]. In this method, a continuous-time LSTM model is proposed, where it consists of two components: (i) discrete-time and (ii) continuous-time components. The discrete-time component is similar to the known LSTM model of [39]. In continuous-time procedure, cell function $c(t)$ decays with decay rate δ. The discrete part of the method is an LSTM model introduced in [25], with additional equations of [25]

$$\begin{aligned} \bar{c}_{i+1} &= \bar{f}_{i+1} \odot \bar{c}\left(t_i\right) + \bar{i}_{i+1} \odot \mathbf{z}_{i+1}, \\ \boldsymbol{\delta}_{i+1} &\leftarrow f\left(\mathbf{W}_d \mathbf{k}_i + \mathbf{U}_d \mathbf{h} t_i + \mathbf{d}_d\right). \end{aligned} \tag{7}$$

$\mathbf{k}$ represents the type of an event which is a one hot vector. $\mathbf{W}$, $\mathbf{U}$, and $\mathbf{d}$ are model parameters. $\mathbf{h}$ represents hidden state. $\bar{f}$, $\bar{i}$, and $\bar{c}$ are defined the same as $\mathbf{f}$ (forget gate), $\mathbf{i}$ (input gate), and $\mathbf{c}$ (cell state) with different parameters. The variable δ in 7 is used in continuous update process, where the value of $\mathbf{c}(t)$ is updated with an exponential decay function which is adapted from Hawkes decay function [25]:

$$\mathbf{c}(t) = \bar{c}_{i+1} + \left(\mathbf{c}_{i+1} - \bar{c}_{i+1}\right) e^{-\delta_{i+1}\left(t - t_i\right)} \quad \text{for}\, t \in \left(t_i, t_{i+1}\right]. \tag{8}$$

According to (8), the value of $\mathbf{c}(t)$ is decreased from $\bar{c}_{i+1}$ towards $\mathbf{c}_{i+1}$ as $t \longrightarrow \infty$. These cell values control hidden states $\mathbf{h}(t)$ and intensities $\lambda_k(t)$ according to [25]

$$\mathbf{h}(t) = \mathbf{o}_i \odot \left(2\sigma(2\mathbf{c}(t)) - 1\right) \text{for}\, t \in \left(t_{i-1}, t_i\right], \tag{9}$$

$$\lambda_k(t) = f_k\left(\mathbf{w}_k^T \mathbf{h}(t)\right), \tag{10}$$

where f_k: $\mathfrak{R} \longrightarrow \mathfrak{R}^{+}$ is the transfer function which is used for converting negative values to positive values [25]. This function is defined as $\lambda_k(t) = f_k(\tilde{\lambda}_k(t)) = s_k \log(1 + \exp(\tilde{\lambda}_k(t)/s_k))$.

In our research, for improving the accuracy of this model, a sequence of a certain length of history (i.e., a sliding window consisting of l previous events) for every event is used for calculating its intensity and is fed into the LSTM model. Therefore, for every next event, this window moves forward and captures the l number of previous events in history for calculating its intensity. Hence, it is assured that a certain number of events in history (if available) are assumed for calculating the intensity. Nevertheless, this increases the time complexity of the method.

3.4.2. Event Types. In the proposed method, network block (cluster) pairs are considered event types. This assumption makes the model proposed in the [25] applicable for modeling network interactions. This is because, if every pair of nodes in a network were considered as event type, the number of types would be in the order of N^2. This yields enormous sizes of parameters in the neural network model, which makes it unfeasible for rather large networks. Hence, in our block neural point process model, nodes of the network are clustered and divided into blocks $\{b_1, \ldots, b_K\}$, where K is the number of clusters in the network, which can be obtained by Eigenvalue decomposition method or by experimental methods and choosing the best K.

Usually, large K values result in more accuracy (especially if the nodes are not well clustered). However, a high number of clusters demand more computations and memory usage. Therefore, this can be considered as a trade-off problem, and the best number of clusters must be selected.

According to this assumption, the type of event i is represented as $k_i = (b_r, b_s)$ where b_r and b_s are the blocks to which interacting nodes belong. Hence, the total number of event types decreases to the order of K^2 (since we have $K^2 \leq N^2$), which makes it feasible for modeling network interactions. In this model similar to the block point process model [15], all nodes inside a block are treated equivalently and have equal probability of being selected as the interacting node inside that block.

The framework of the proposed method is illustrated in Figure 2. As it is represented, first, the nodes of the aggregated network are clustered using the spectral clustering algorithm. After obtaining the blocks, all interactions are clustered according to the block pairs they belong to, and sequences of events are obtained. These sequences are used for training the model. Having a trained model, a sequence of events is given as an input to the continuous-time LSTM module, and intensity functions $\lambda_k(t)$ for all event types k are calculated. After obtaining intensities, the future events time and type can be sampled and generated using the thinning algorithm [25], which is explained in this section.

3.4.3. Inference Method. The parameter inference method of the proposed method is the same as the inference method introduced in [25]. In this method, the log-likelihood of the train events (5) is maximized using the gradient descent method. In this equation, the integral expression (and its gradient) is intractable and must be estimated. For tackling this problem, the method presented in [25] is used in our research, where a Monte Carlo algorithm was introduced for estimating the integral and its gradient.

3.4.4. Simulation Method. For generating sample events, i.e., interactions in the network, the thinning algorithm [40] is used as applied in [25]. In this algorithm, the types and times of the next events are generated using the thinning algorithm as explained in [25]. The output of this algorithm is a series of events in the form of $\{(k_i, t_i)\}_1^n$ which are block-level samples. Hence, for generating node level samples, it is required to sample nodes from each block pair as types of events. As mentioned earlier, these nodes are sampled randomly from each block, i.e., with the same probability $1/|b_r|$, where $|b_r|$ represents the number of nodes in block b_r. The overall simulation procedure is given in Algorithm 1.

4. Evaluation Metrics

In our research, different metrics for evaluating the type and time of events are used. For evaluating both types (in node pairs level) and the time of the events, the mean log-likelihood of test events is used. In order to account for the complexity of a model (number of its parameters), the Akaike measure is also utilized for evaluating different models. Furthermore, some information retrieval metrics are applied to evaluate only the types of the simulated events (or link prediction) and their orderings performance. Finally, relative errors are calculated for measuring parameter estimation errors for simulated data. These metrics are explained as the following.

4.1. Mean Log-Likelihood. The log-likelihood of events as presented in (5) is an appropriate metric for evaluating intensity-based models, but this equation evaluates the likelihood of events in block levels (types). Hence, for evaluating the model independent of cluster structures, this equation is converted into

$$\mathscr{L}_{\text{node\_pairs}} = \sum_{i:t_i<T} \log \frac{\lambda_{k_i}(t_i)}{N(k_i)} - \int_{t=0}^{T} \lambda(t)\mathrm{d}t = \sum_{i:t_i<T} \log\lambda_{k_i}(t_i) - \sum_{i:t_i<T} \log N(k_i) - \int_{t=0}^{T} \lambda(t)\mathrm{d}t. \quad (11)$$

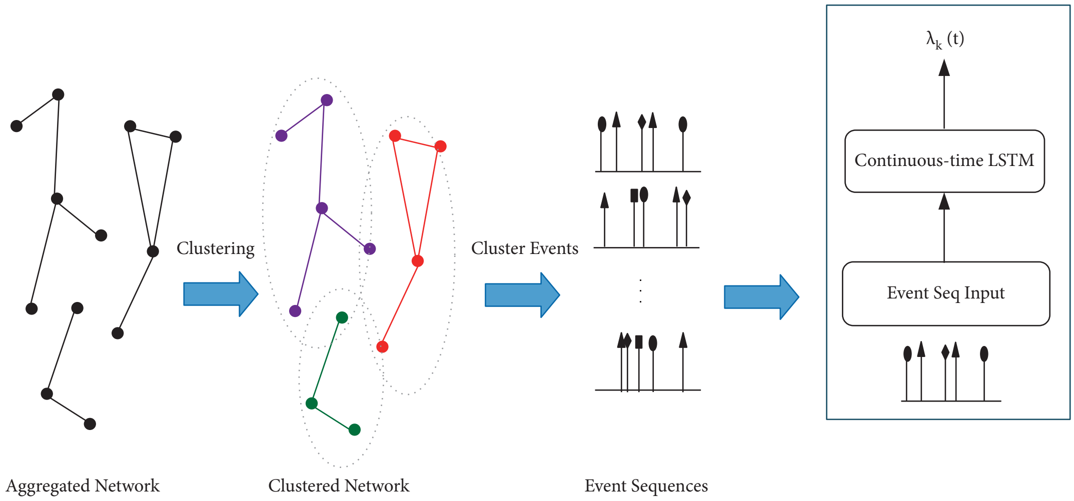


Figure 2: The framework of the proposed method.

```
Input: Interval $[\text{start\_}T, \text{end\_}T]$; Previous events $\{(k_i, t_i)\}_1^M$; Blocks $\{b_i\}_1^K$ and Model parameters
Output: Generated node-level events $\{(s_i, r_i, t_i)\}$
$t_0$←start_$T$; $i$←1prev_events$_i$←Previous events
while $t_{i-1}$ < end_$T$ do
    $((b_{s_i}, b_{r_i}), t_i)$← Generate next event using thinning algorithm [25]
    Randomly select $s_i$ and $r_i$ from blocks $b_{s_i}$ and $b_{r_i}$
    add sample $(s_i, r_i, t_i)$ to prev_events$_i$
    pop first event from prev_events$_i$
    $i$←$i$ + 1.
```

Algorithm 1: Simulation-thinning.

In this equation, $N(k_i)$ represents the number of possible events in event type k_i. In other words, it represents the number of node pairs in block pairs (b_{r_i}, b_{s_i}) representing event type k_i, which is

$$N(k_i) = \begin{cases} |b_{r\_i}| \times |b_{s\_i}|, & b_{r\_i} \neq b_{s\_i}, \\ \dfrac{|b_{r\_i}|\left(|b_{r\_i}| - 1\right)}{2}, & \text{otherwise}. \end{cases} \quad (12)$$

$|b_{r\_i}|$ is the number of nodes in block b_{r_i}. In the third term of (11), $\lambda(t)$ is the summation over all types of events, i.e., $\lambda(t) = \sum_{k:1}^{K} \lambda_k(t)$. Using (11), log-likelihood of events in node pairs level is obtained, which is independent of network clustering.

After obtaining the total log-likelihood of test events, the mean value is calculated by dividing the total log-likelihood by the number of test events. It results in a metric independent of the number of events (or test interval).

4.2. Akaike Measure.

4.2. Akaike Measure. Akaike information criterion (AIC) is a measure for evaluating prediction error and quality of the statistical model and is used to select the appropriate model among different models. This measure considers a model's efficiency by counting the number of parameters in addition to prediction error. This measure is defined as follows:

$$\text{AIC} = 2k - 2\ln\mathscr{L}, \quad (13)$$

where k represents the number of estimated parameters in the model and $\mathscr{L}$ is the likelihood of tested events. Less values for this measure are more desirable, and it includes a penalty for the number of parameters in the model. Hence, this measure can be compared between different models with a different number of parameters.

4.3. Information Retrieval Measures. For evaluating the link prediction, i.e., types of the events, general information retrieval metrics have been used. These metrics consider the number of correctly and mispredicted links in the ordered simulated and real links. In this research, precision, recall, and $F1$-score are used, which are TP/TP + FP, TP/TP + FN, and 2 $*$ recall $*$ precision/recall + precision, respectively. TP, FP, and FN are the numbers of correctly predicted, mispredicted, and falsely unpredicted links. In this research, these measures are calculated for top K predicted links, i.e., precision@K, recall@K, and $F1$ − score@K.

4.4. Mean Relative Error of Parameter Estimation. For evaluating the parameter estimation accuracy of a model for the simulated data generated using that model, relative errors are calculated, and the mean value as mean relative error (MRE) is reported. This error is calculated as the equation:

$$\text{MRE} = \mathbb{E}\left[\frac{x - \tilde{x}}{\tilde{x}}\right]. \tag{14}$$

In this equation, x represents the estimated parameter and $\tilde{x}$ is the true parameter. Mean values of these errors through different simulations are calculated and reported as parameter estimation errors.

5. Experiments and Numerical Results

In this section, experimental details and the obtained results are presented. First, in this section, experimental settings are explained. Then the synthetic and real-world datasets are explained. Finally, the experiments' results are presented and discussed at the end of this section.

5.1. Experimental Settings. This section presents some details about the implementation environment and settings of the model. The proposed method has been implemented on a system with an Intel Core i5 processor, NVIDIA GeForce GTX 1650 GPUs and 4 GB GPU RAM. CUDA technology has been used for parallelization and speeding up the computations. Besides, the Networkx library has been used for processing network data.

According to different network sizes, various values for hyperparameters have been set. The sizes of the hidden layer of neural networks are set to the values in the range of [20, 100]. The sequences of lengths 20 to 100 have been used for modeling events dependency on their history. For training the model with batches, a batch containing several consecutive events (100–200) is selected at each training iteration. The log-likelihood of these events during the time interval containing them is calculated and maximized using the Adam optimization method with a learning rate of 0.001. The training process is iterated for 3 to 100 epochs of data. In order to evaluate the models using information retrieval metrics, the first 50, 100, and 200 events in test events were considered.

5.2. Datasets. The proposed method has been compared with other methods using both synthetic and real-world datasets. These data and their generation details are presented in the following.

5.2.1. Synthetic Data. For generating synthetic data, two event-based network models, i.e., the vanilla Hawkes process and block point process models, were used in this research. The thinning algorithm was used to simulate point processes (Hawkes process). The vanilla Hawkes networks of 50, 100, and 200 nodes are with parameters $\mu = 0.00005, 0.05$, $\alpha = 0.03, 0.25$, and $\beta = 0.03, 0.39$. Every network was simulated ten times, and mean values of evaluation metrics and errors (standard deviations) were reported. The block Hawkes networks of the same sizes with three clusters (with ten simulations for each network) were generated. The average number of events for each generated network type is given in Table 1. The performances of different models on these networks are evaluated in Section 5.3.1.

5.2.2. Real-World Data. Different real-world network interaction data have been used for testing the effectiveness of different models. These datasets consist of different human interaction networks, which are detailed in the following and their statistics are given in Table 2:

(1) MIT Reality Mining. This dataset consists of phone call information between 75 MIT students and faculty [41] from October 2001 to February 2002. The preprocessed data used in [14, 16] were also used in this research for evaluating different methods.

(2) Hospital Dataset. (https://www.sociopatterns.org/datasets/hospital-ward-dynamic-contact-network/) This dataset represents the interactions between patients and faculty of a hospital in Lyon, France, from Monday, December 6, 2010, to Friday, December 10, 2010. The Hospital data includes interaction information of 75 people [42].

(3) Hypertext Dataset. (https://www.sociopatterns.org/datasets/hypertext-2009-dynamic-contact-network/) This dataset contains face-to-face proximity information between 113 participants of the ACM Hypertext 2009 conference during about 2.5 days [43].

(4) Enron Email Dataset. Enron dataset [44] contains e-mail messages between about 150 users of the Enron Corporation from July 2001 to August 2001. The preprocessed version of this dataset from [14, 16] was used in this research, containing 142 users.

(5) Primary School Dataset. (https://www.sociopatterns.org/datasets/primary-school-temporal-network-data/) This dataset contains interactions between students and teachers during about 1.5 days and includes 242 nodes [45, 46].

(6) Highschool Dataset. (https://www.sociopatterns.org/datasets/high-school-contact-and-friendship-networks/) Highschool dataset involves contact information between high school students in France [47]. This dataset contains 327 nodes interacting over five days in December 2013.

Multiple splits of each dataset were selected (with constant duration of half of the total duration) to evaluate the models on different realizations of these datasets, and their statistics are given in (2). The splits with less than 50 events in validation or test events were ignored. The mean values of evaluation metrics and their errors on these splits for each dataset are calculated and discussed in Section 5.3.2. Every split of both synthetic and real-world datasets is divided into three parts, i.e., train, validation, and test events. Validation events are used in our proposed method to select the best model among trained models for different data epochs.

Table 1: Statistics for different realizations of synthetic networks.

Network type	Number of nodes	Average number of events
Vanilla Hawkes	50	6114.7
	100	7432.2
	200	6966
BPPM	50	6173.1
	100	6142.6
	200	6144.7

Table 2: Dataset statistics for dynamic networks.

Dataset	# of nodes	# of train events	# of valid events	# of test events	# of splits	Average number of events in each split
Reality	70	1303	198	661	9	1297.3
Hospital	75	30869	864	692	12	17061.3
Hypertext	113	16468	2665	1686	9	10396.2
Enron Email	142	2673	328	1000	10	1982.4
Primary School	242	123573	1114	1087	8	45396.5
Highschool	327	187109	925	475	4	93843

In all methods (except vanilla Hawkes method), network datasets are clustered for assigning event types using spectral clustering. In order to obtain an aggregated network for each interaction dataset, the number of interactions between every two nodes is considered as edge weights between them in the aggregated network. Then, this network is clustered using the spectral clustering algorithm considering different numbers of clusters. The results of modelings using different clusters are presented in the next section. In the block point process model, cluster memberships can be optimized using the local search algorithm.

5.3. Results and Discussion. In this section, numerical results of testing methods mentioned above on both synthetic and real-world data are presented and discussed.

5.3.1. Results for Synthetic Networks. The models mentioned above are evaluated on the synthetic networks explained in Section 5.2.1 using different evaluation metrics explained in Section 4. In this section, parameter estimation accuracy, the likelihood of test events, and data generation performance of different models are evaluated, which are explained in the following.

For both synthetic data types generated by vanilla Hawkes and BPPM, the MRE metric evaluates parameter estimation accuracy by the corresponding model. These results are given in Tables 3 and 4. For the Hawkes data, estimation accuracies for μ and α parameters are acceptable, but MRE values for the estimated β parameter are higher (especially for the networks with 100 nodes), and its high standard deviation represents instability of the predicted parameter. It might result from bad initialization of the method, which results in disparate estimations, and β is the parameter of the exponent more affected by the initialization. Furthermore, by increasing the number of simulations, the deviation might decrease. For the BPPM data, as represented in Table 4, relative errors of parameter estimations and their deviations are acceptable, representing a more accurate estimation of this method.

Table 3: Parameter estimation errors by vanilla Hawkes model for synthetic Hawkes data.

# of nodes	μ	α	β
50	0.0109 (0.0076)	0.9841 (0.0153)	2.0952 (2.7723)
100	0.0129 (0.0070)	0.9943 (0.0097)	29.9112 (70.2657)
200	0.0103 (0.0062)	0.9995 (0.0008)	1.3768 (1.8095)

Table 4: Parameter estimation errors by BPPM for synthetic BPPM data.

# of nodes	μ	α	β
50	0.9929 (0.4299)	0.9891 (0.0025)	0.9820 (0.0062)
100	0.7251 (0.4919)	0.9895 (0.0088)	0.9977 (0.0741)
200	0.3897 (0.0766)	0.9819 (0.0170)	1.0128 (0.1346)

The results of measuring test data's likelihood (log-likelihood) and data generation performance are also illustrated in Figures 3–6. The illustrated information retrieval metrics evaluate types of generated events which is dependent on the clustering. Hence, they should be considered separately for each number of clusters. But log-likelihood and Akaike measures can be compared through different clusterings. As can be seen, the mean log-likelihood results represent the similar performance of different methods for the small number of clusters (except for the BPPM with one cluster in BPPM data with 100 nodes). In order to include model complexity in log-likelihood, the Akaike measure was also evaluated. According to these figures, the Akaike of our proposed method (BNHM) is high, representing its high complexity, which should be decreased in the case of insufficient resources. It can be performed by decreasing the number of hidden layers of the model, but it might affect the model's performance, which should be configured according to the application. As Figures 3–6 represent, the log-likelihood of the BNHM decreases with increasing the number of clusters of the model. Since the complexity of the model

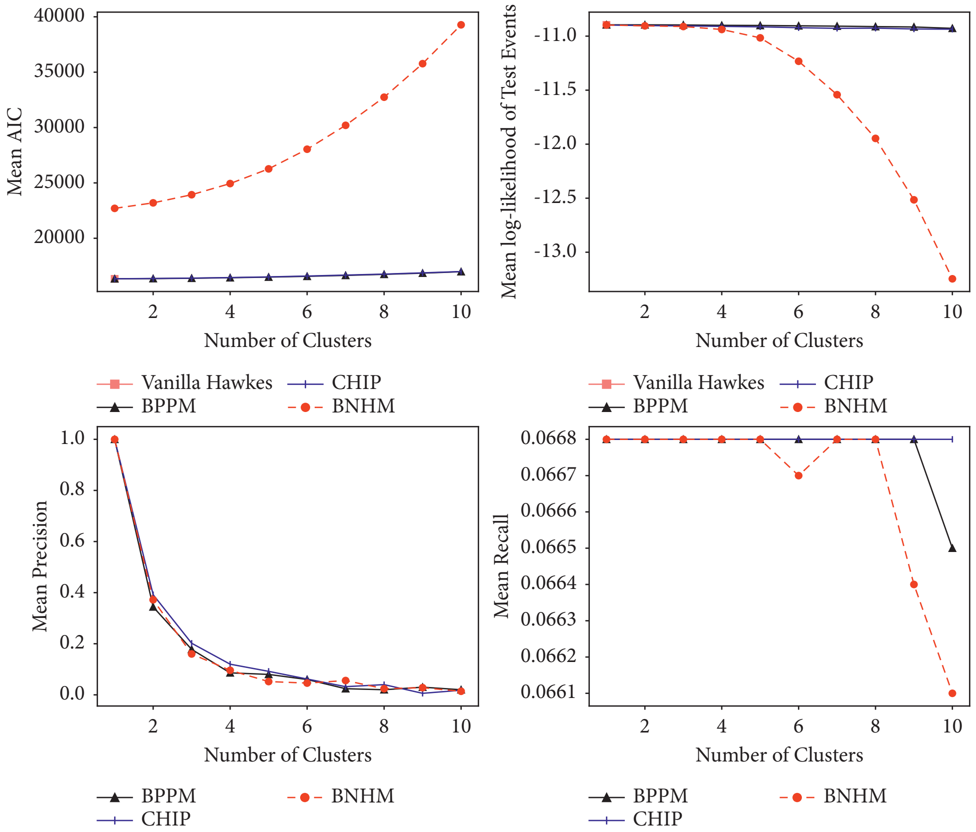


FIGURE 3: Mean results for synthetic Hawkes data with 100 nodes.

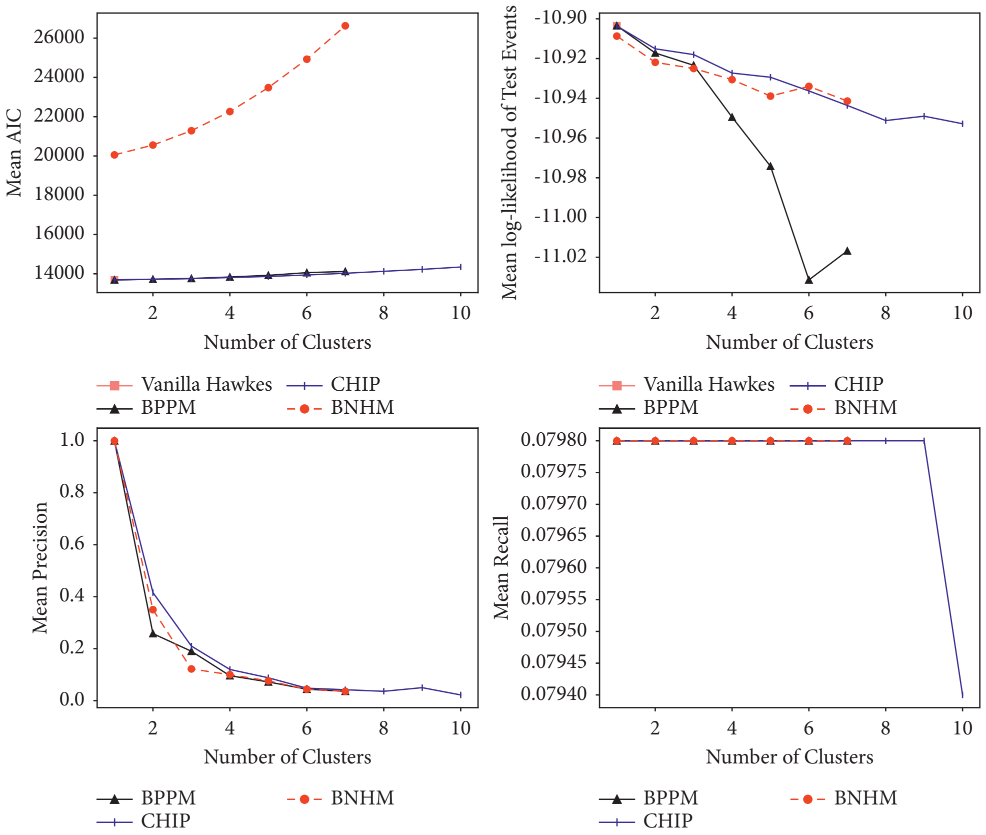


FIGURE 4: Mean results for synthetic Hawkes data with 200 nodes.

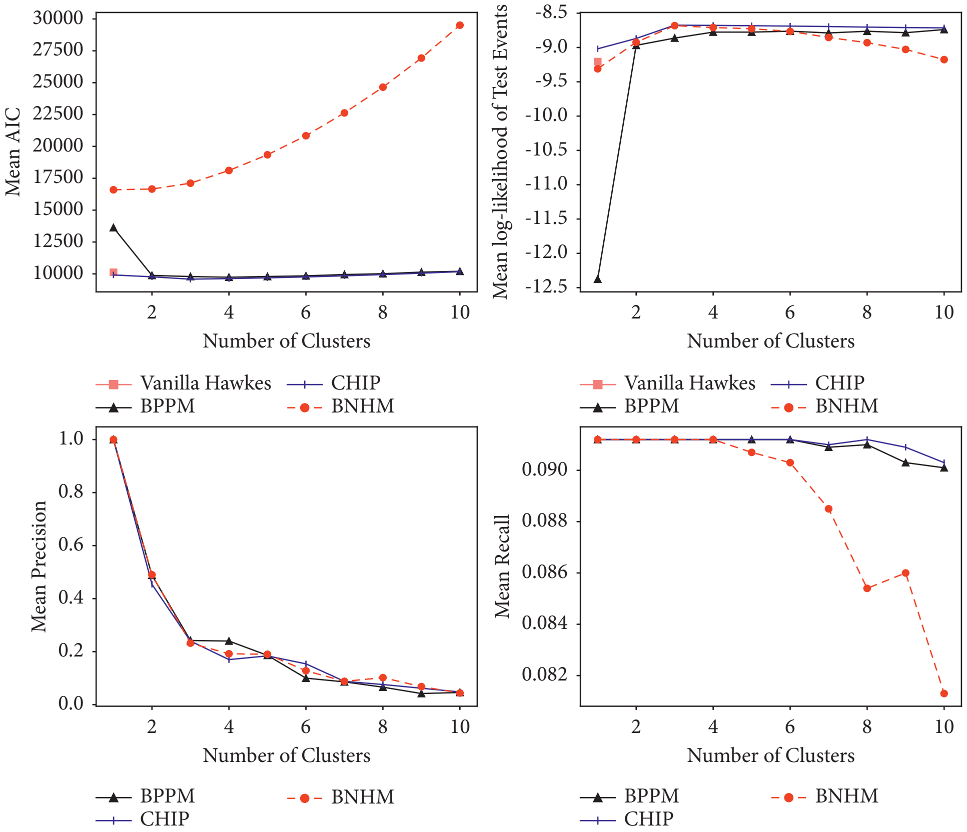


Figure 5: Mean results for synthetic BPPM data with 100 nodes.

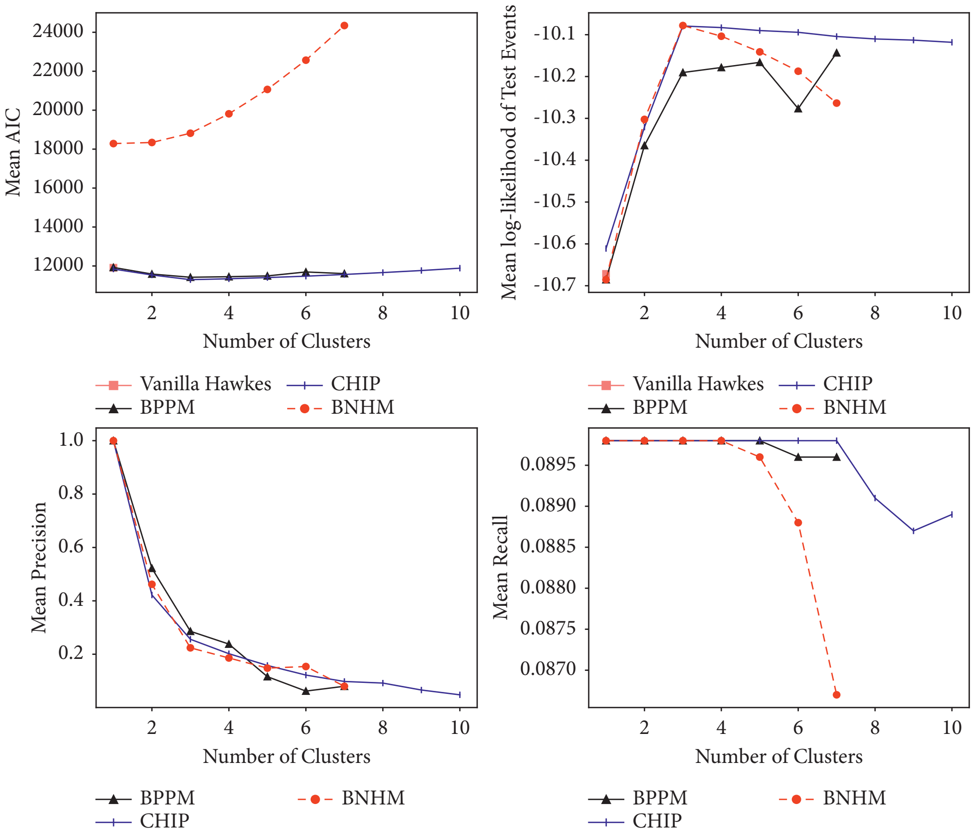


Figure 6: Mean results for synthetic BPPM data with 200 nodes.

Table 5: Best mean results for synthetic Hawkes dynamic networks with different simulations

# of nodes	Method	Best K	AIC	Mean log-likelihood	Mean P	Mean R
50	Hawkes	1	**4414.8 (83.7)**	**−4.0 (0.03)**		
	BPPM	1	**4415.1 (83.6)**	**−4.0 (0.03)**	1	0.09 (0.002)
	CHIP	1	**4415.1 (83.4)**	**−4.0 (0.03)**	1	0.09 (0.002)
	BNHM	1	12447.5 (155.5)	−5.51 (0.02)	1	0.09 (0.002)
100	Hawkes	1	**16340.2 (600.9)**	**−10.90 (0.04)**		
	BPPM	1	**16340.2 (601.0)**	**−10.90 (0.04)**	1	0.07 (0.003)
	CHIP	1	**16343.0 (601.0)**	**−10.90 (0.04)**	1	0.07 (0.003)
	BNHM	1	22702.5 (598.3)	**−10.90 (0.04)**	1	0.07 (0.003)
200	Hawkes	1	**13688.7 (384.5)**	**−10.90 (0.03)**		
	BPPM	1	**13688.8 (384.5)**	**−10.90 (0.03)**	1	
	CHIP	1	**13688.9 (384.4)**	**−10.90 (0.03)**	1	0.08 (0.002)
	BNHM	1	20055.4 (386.6)	**−10.91 (0.03)**	1	0.08 (0.002)

Table 6: Best mean results for synthetic BPPM dynamic networks with different simulations.

# of nodes	Method	Best K	AIC	Mean log-likelihood	Mean P	Mean R
50	Hawkes	1	8388.5 (240.3)	−7.68 (0.07)		
	BPPM	3	8311.9 (191.6)	−7.6 (0.08)	0.27 (0.06)	0.09 (0.003)
	CHIP	3	**7990.4 (205.8)**	**−7.27 (0.05)**	0.25 (0.06)	0.09 (0.003)
	BNHM	1	15008.6 (224.5)	−7.92 (0.03)	1	0.09 (0.003)
100	Hawkes	1	10122.2 (493.9)	−9.21 (0.06)		
	BPPM	4	9737.2 (435.3)	−8.78 (0.10)	0.24 (0.07)	0.09 (0.005)
	CHIP	3	**9584.7 (435.6)**	**−8.67 (0.09)**	0.24 (0.07)	0.09 (0.005)
	BNHM	1	16597.2 (512.1)	−9.31 (0.05)	1	0.09 (0.005)
200	Hawkes	1	11909.7 (462.6)	−10.67 (0.04)		
	BPPM	3	11420.3 (426.2)	−10.19 (0.06)	0.29 (0.04)	0.09 (0.004)
	CHIP	3	**11296.6 (428.8)**	**−10.08 (0.05)**	0.26 (0.08)	0.09 (0.004)
	BNHM	1	18284.9 (459.0)	−10.69 (0.04)	1	0.09 (0.004)

increases with the increasing number of clusters, this might represent the need for more training iterations for these models, which is more time-consuming. Accordingly, for higher numbers of clusters in larger networks (with 200 nodes) due to higher time complexity, BNHM and BPPM were not trained. But the CHIP model has better speed in training and is more appropriate for time-constrained applications. Furthermore, for each model, the best model according to the Akaike measure was selected and reported in Tables 5 and 6. As Figures 3 and 6 and Table 5 illustrate, for Hawkes data, vanilla Hawkes, BPPM, and CHIP perform similarly (considering one cluster). According to Table 6, for BPPM data, BPPM and CHIP models can usually find the proper number of clusters (which is three) in addition to better performance in terms of log-likelihoods (which is close to vanilla Hawkes result). But BNHM fails to find the correct number of clusters, which can arise from a small number of training iterations. Overall, according to these results, the CHIP model outperforms all other methods (its accuracy and time complexity) for both types of synthetic data.

5.3.2. Results for Real-World Networks. This section presents the results of testing Hawkes-based models explained in Section 3.3 and our proposed model (BNHM) on different real-world datasets. Similar to the experiments performed for synthetic networks, several experiments were done for different splits of real-world data, and mean values of evaluation metrics and their errors (standard deviation) are reported. In addition, for cluster-based models (BPPM, CHIP, and BNHM), different numbers of clusters were tested, and the best cluster numbers for each model (in each dataset) were obtained. Finally, a comparison of different cluster numbers for some datasets (using the whole dataset) is also presented.

The average results of evaluating different methods on different splits of six real-world datasets and their selected models according to the Akaike measure are given in Figures 7–12 and Table 7. According to Figure 7, BNHM does not have good performance for the Reality dataset, and it has high model complexity according to the Akaike measure. It has a log-likelihood close to BPPM for the small number of clusters. For this dataset, the CHIP and vanilla Hawkes models represent better performances in terms of Akaike and log-likelihood measures. Considering recall and precision values, BPPM and CHIP have similar performance. As indicated in Table 7, all models except BPPM find one cluster as the best number of clusters. It represents a lower clustering feature of this dataset. The results for Hospital and Hypertext datasets in Figures 8 and 9 and Table 7 illustrate the good performance of BNHM on these datasets. But still, vanilla Hawkes is the best model according to AIC and likelihood values, and BNHM has higher model complexity. In the Hypertext dataset, after vanilla Hawkes, BNHM has overall the best log-likelihood. In these datasets,

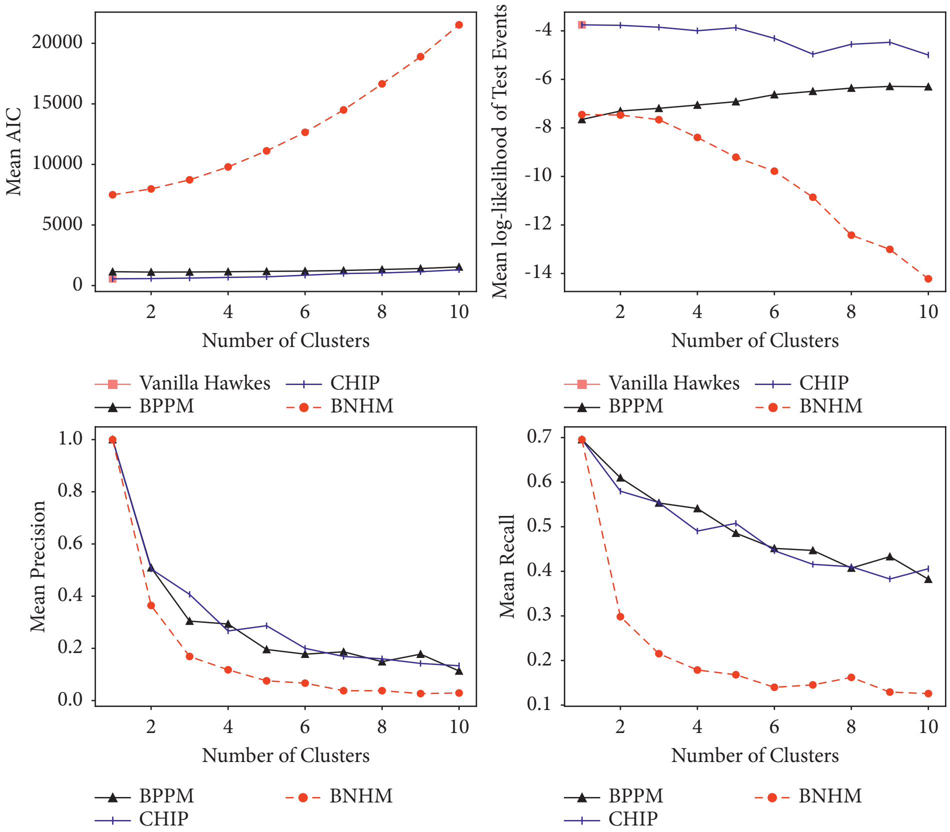


Figure 7: Mean results for reality data.

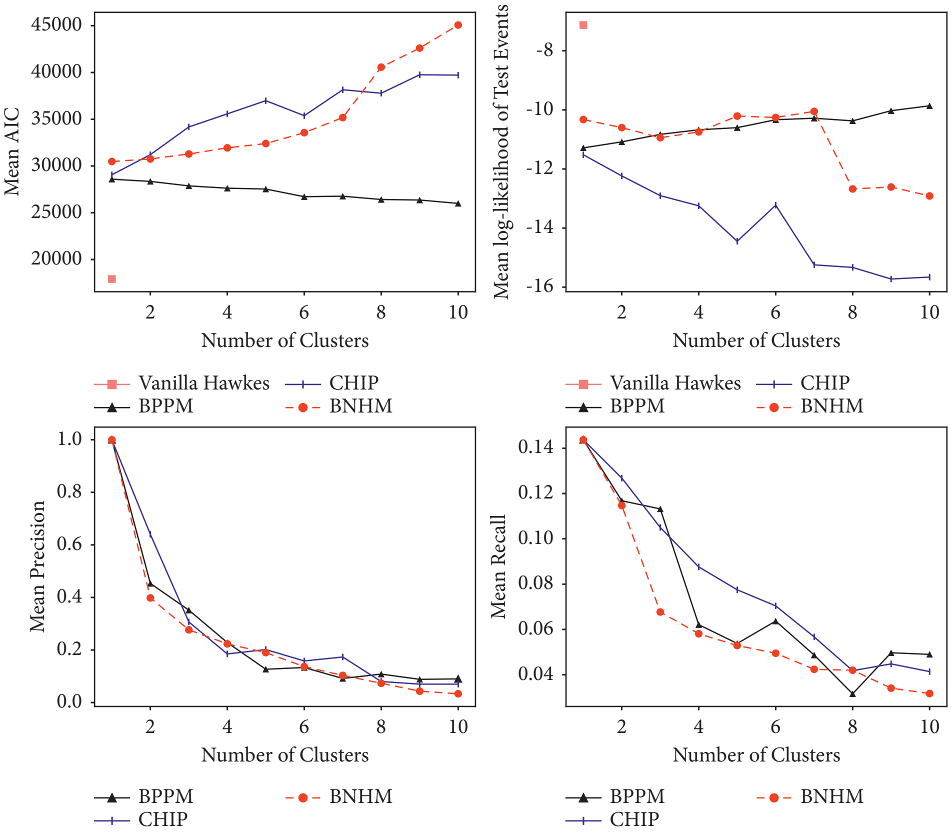


Figure 8: Mean results for Hospital data.

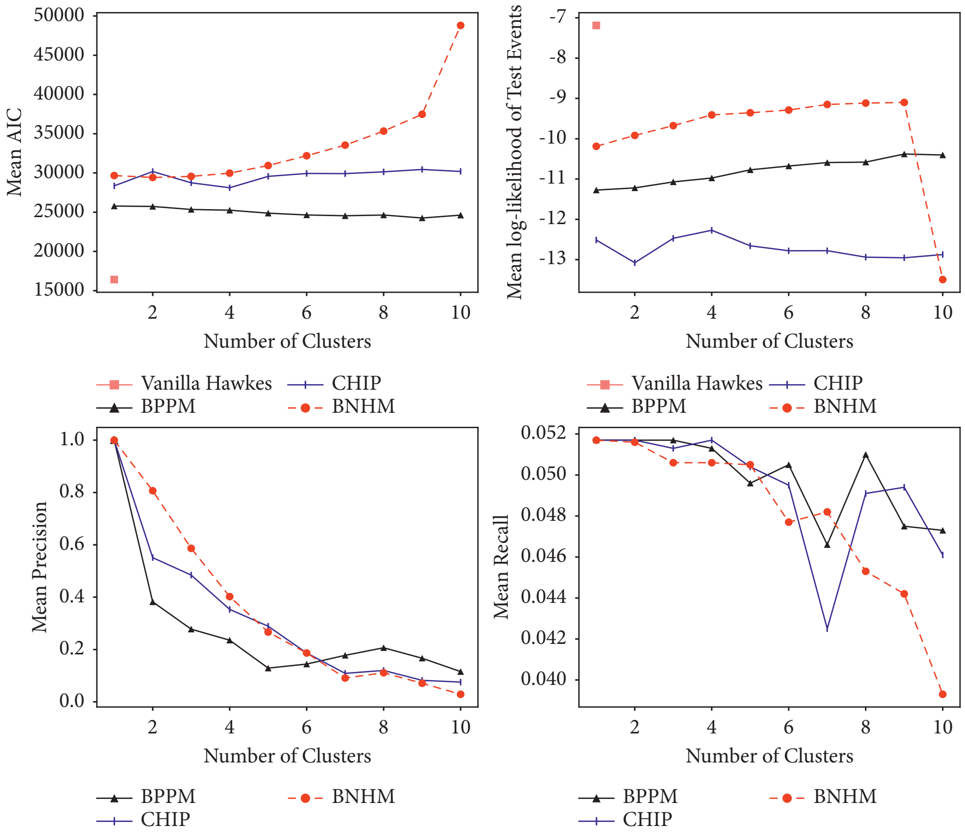


Figure 9: Mean results for Hypertext data.

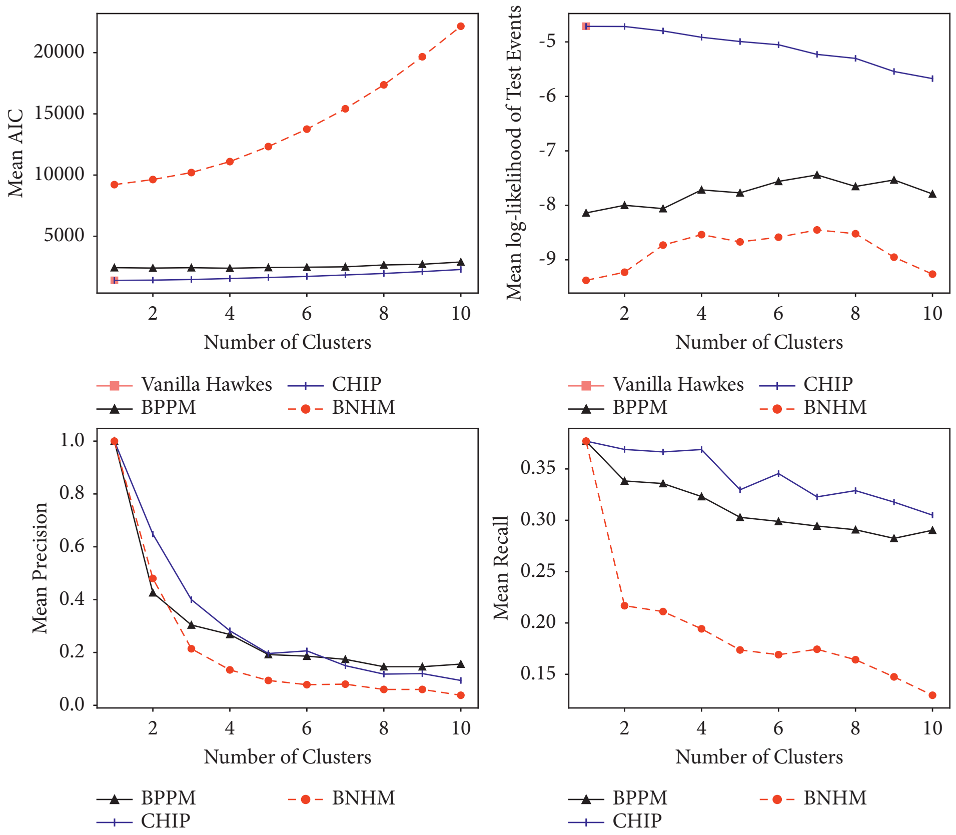


Figure 10: Mean results for Enron data.

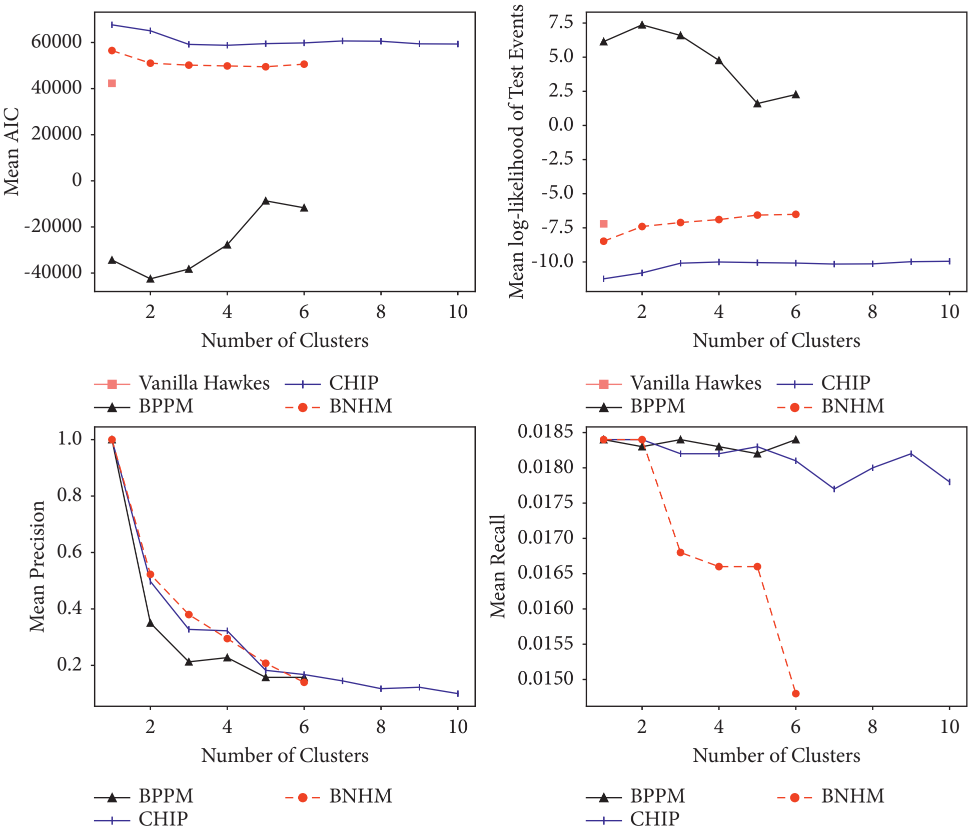


Figure 11: Mean results for Primary School data.

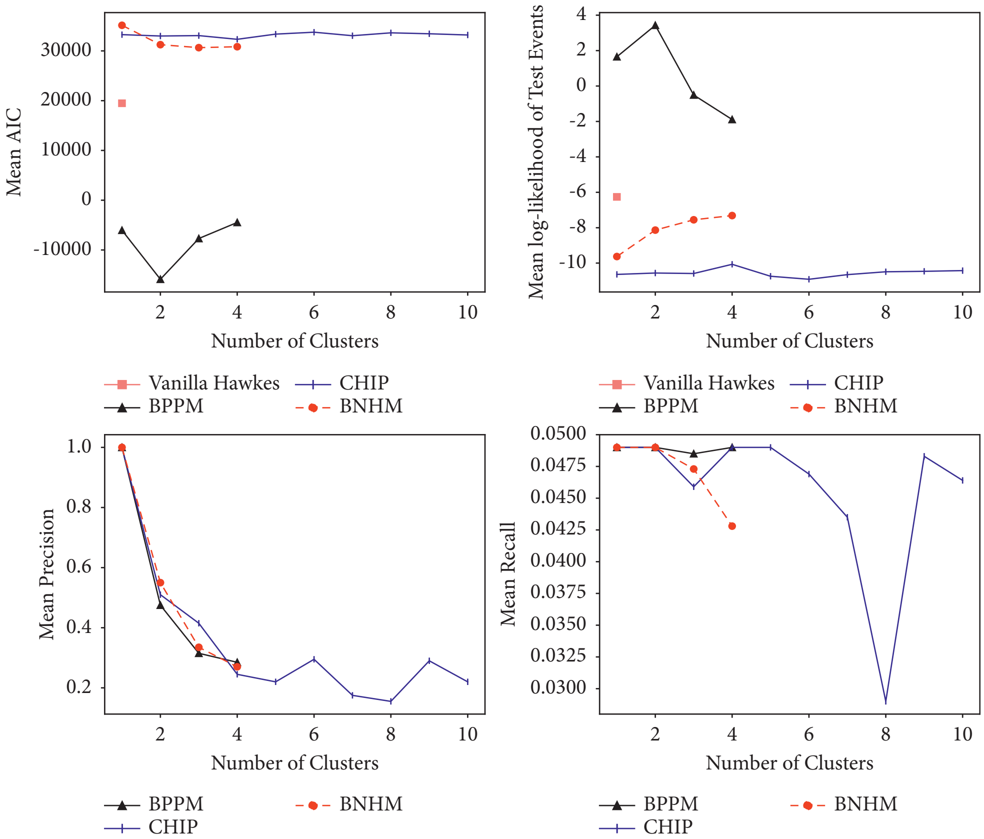


Figure 12: Mean results for Highschool data.

Table 7: Best mean results for real-world dynamic networks with different splits.

Dataset	Method	Best K	AIC	Mean log-likelihood	Mean P	Mean R
Reality	Hawkes	1	**557.4 (105.5)**	**−3.75 (0.66)**		
	BPPM	2	1112.2 (216.4)	−7.30 (0.40)	0.51 (0.16)	0.61 (0.12)
	CHIP	1	**557.7 (104.0)**	**−3.75 (0.64)**	1	
	BNHM	1	7495.2 (342.2)	−7.45 (1.48)	1	
Hospital	Hawkes	1	**17915.4 (14586.4)**	**−7.13 (0.95)**	1	0.14
	BPPM	10	26002.5 (20429.3)	−9.86 (1.25)	0.09 (0.07)	0.05 (0.05)
	CHIP	1	29062.5 (28700.1)	−11.52 (3.67)	1	0.14 (0.20)
	BNHM	1	30481.0 (18757.3)	−10.33 (2.39)	1	0.14 (0.20)
Hypertext	Hawkes	1	**16418.9 (6463.4)**	**−7.19 (0.54)**		
	BPPM	9	24260.6 (9404.9)	−10.38 (0.49)	0.17 (0.17)	0.05 (0.02)
	CHIP	4	28115.1 (12354.2)	−12.27 (2.80)	0.35 (0.22)	0.05 (0.02)
	BNHM	2	29425.8 (8701.2)	−9.92 (0.67)	0.81 (0.18)	0.05 (0.02)
Enron email	Hawkes	1	**1399.2 (522.8)**	**−4.71 (0.66)**		
	BPPM	4	2389.8 (830.8)	−7.72 (0.71)	0.27 (0.18)	0.32 (0.09)
	CHIP	1	**1398.4 (524.6)**	**−4.72 (0.70)**	1	0.38 (0.15)
	BNHM	1	9218.3 (1195.8)	−9.38 (0.16)	1	0.38 (0.15)
Primary School	Hawkes	1	42305.0 (12058.3)	−7.21 (0.56)		
	BPPM	2	**−42497.2 (17832.3)**	**7.37 (2.19)**	0.35 (0.19)	0.02 (0.006)
	CHIP	4	58783.9 (24182.7)	−10.00 (2.93)	0.32 (0.14)	0.02 (0.006)
	BNHM	5	49477.5 (15642.1)	−6.57 (0.66)	0.21 (0.06)	0.02 (0.004)
Highschool	Hawkes	1	19478.6 (11788.4)	−6.26 (0.43)		
	BPPM	2	**−15885.7 (19601.6)**	**3.43 (3.60)**	0.48 (0.17)	0.05 (0.04)
	CHIP	4	32330.1 (24086.3)	−10.06 (4.54)	0.25 (0.12)	0.05 (0.04)
	BNHM	3	30653.5 (13693.0)	−7.55 (0.62)	0.34 (0.08)	0.05 (0.04)

BNHM in more clusters has a degradation in its performance because of fewer training iterations. For the Enron dataset, also CHIP and vanilla Hawkes models perform better (in terms of Akaike and log-likelihood). The BPPM outperforms in Primary School and Highschool datasets which can represent dependency between cluster members' interactions. Its performance in these datasets is significantly better in terms of Akaike and log-likelihood measures. However, it should be mentioned that CHIP has a faster training process, and BPPM and BNHM models have higher training time complexity.

The results in Table 7 have high errors (standard deviations) for some datasets, which can arise from the small number of splits selected for these datasets and the variety of event patterns in the selected splits (corresponding to different timings). Due to the data dependency feature of deep learning methods and requiring enough data for training the BNHM, the splits were selected to include enough events for training.

In addition to the experiments on multiple splits of real data, all models were also trained and tested using all the events in these datasets. These results are illustrated in Table 8, where the best cluster number for each model (cluster-based models) is selected according to log-likelihood values. According to these results, the CHIP model has the best results in Reality Mining and Enron Email datasets, which is very similar to the results of the vanilla Hawkes model. In these datasets, the CHIP model has a small number of clusters (2 and 3) which makes it close to the vanilla Hawkes model (CHIP model with 1 cluster). For these datasets, BPPM and BNHM have close results with the same number of clusters (10 and 4). This table illustrates that BNHM has the best result for Hospital and Hypertext datasets with 4 clusters. In addition, the vanilla Hawkes model performs well for these datasets (as the second best method). For Primary School and Highschool datasets, BPPM has significantly the best results with 1 cluster, i.e., no clustering. It can confirm the interdependence between interactions of node pairs in these datasets. In addition, according to this model, all nodes have the same dynamic process (because of no clustering). BNHM has the second best results for the Primary School dataset with more clusters, but BPPM performs considerably better. In the Highschool dataset, the vanilla Hawkes model also performs well as the second best model. According to these results, the compared models perform differently on different datasets. Besides, because of the large number of parameters of our proposed model, it has an extendibility problem that can not consider a large number of clusters for larger datasets and longer sequences for capturing more history information.

In addition, information retrieval metrics were compared on two datasets considering different clusterings. The results of these experiments are given in Tables 9 and 10. In Table 9, the results for the Reality Mining dataset are illustrated. For this dataset, BPPM and BNHM have better log-likelihood with increasing the number of clusters. However, for the CHIP model, fewer clusters result in better performance. It is also noticeable that our proposed method has better precision (and $F1$-score) for fewer clusters (2 and 6). For the Enron dataset, CHIP has the best results regarding the log-likelihood of test events. Although BNHM has not had good efficiency for this dataset, precision@K values are comparable with other methods.

Table 8: Results for real-world dynamic networks.

Dataset	Method	Mean log-likelihood	K
Reality	Hawkes	−3.87	1
	BPPM	−5.95	10
	CHIP	−3.83	2
	BNHM	−6.01	10
Hospital	Hawkes	−6.78	1
	BPPM	−9.17	10
	CHIP	−7.87	2
	BNHM	−5.74	4
Hypertext	Hawkes	−7.36	1
	BPPM	−10.32	10
	CHIP	−11.66	5
	BNHM	−7.3	4
Enron Email	Hawkes	−4.8	1
	BPPM	−7.2	4
	CHIP	−4.8	3
	BNHM	−7.5	4
Primary School	Hawkes	−7.87	1
	BPPM	6.7	1
	CHIP	−11.3	13
	BNHM	−2.46	13
Highschool	Hawkes	−6.4	1
	BPPM	0.39	1
	CHIP	−9.08	7
	BNHM	−6.66	7

Table 9: Results of Reality dataset.

Method	K	Mean log-likelihood	Precision @200	Recall @200	$F1$-score @200
Hawkes	1	−3.87	—	—	—
BPPM	2	−7.55	0.45	0.3	0.36
	6	−6.86	0.26	0.3	0.28
	10	−6.61	0.04	0.29	0.08
CHIP	2	−3.83	0.46	0.3	0.37
	6	−5.05	0.18	0.28	0.22
	10	−5.9	0.14	0.3	0.2
BNHM	2	−8.08	0.93	0.30	0.46
	6	−7.88	0.9	0.29	0.44
	10	−6.01	0.14	0.28	0.19

Table 10: Results for Enron dataset.

Method	Cluster num.	Mean log-likelihood	Precision @50	Recall @50	$F1$-score @50
Hawkes	1	−4.8	—	—	—
BPPM	3	−7.43	0.32	0.05	0.09
	4	−7.2	0.22	0.05	0.08
	6	−7.3	0.06	0.05	0.05
CHIP	3	−4.8	0.34	0.05	0.09
	4	−5	0.26	0.05	0.08
	6	−4.9	0.3	0.05	0.09
BNHM	3	−7.97	0.34	0.05	0.09
	4	−7.5	0.44	0.05	0.09
	6	−7.6	0.26	0.05	0.08

Due to the difference of models in different datasets, some relations between features of datasets and the appropriate model can be concluded. According to the results of real-world datasets, we might conclude that BPPM outperforms other models in larger networks. It can be the result of the model's assumption, which considers interactions on a large scale (cluster level). Hence, it is more appropriate for larger networks (such as Primary School and

Highschool datasets). The BNHM has the same assumption but requires more training iterations to be more efficient because of its model complexity. Furthermore, another conclusion that can be drawn is that BPPM performs better in datasets including face-to-face interactions and a high clustering feature with a higher influence of interactions on each other (which are the characteristics of Primary School and Highschool datasets). On the other side, the CHIP (and vanilla Hawkes) model performs better for datasets where interactions are not face-to-face (such as online interactions in Enron and Reality datasets). For BNHM, it is required to decrease the model complexity to increase its efficiency in real applications.

6. Conclusion

In this research, different point process (Hawkes) based dynamic network models were compared using different synthetic and real-world datasets. Furthermore, a neural network Hawkes process based on the method introduced in [25] was proposed, which uses network clustering for specifying types of interactions and models interactions of dynamic networks using a continuous-time LSTM introduced in [25]. These models were tested on six different datasets, and the best number of clusters was selected according to the results. These results for synthetic data express the efficiency of the CHIP model. In addition, the experiments on real-world datasets represent that the performance of the tested methods depends on the dataset, and these methods have different efficiency for the tested datasets. Furthermore, because of the dependence of the proposed model on the number of event types, it is not extensible for more event types such as interactions in networks. Hence, to be more appropriate for the network interactions, it is required to improve the model to not depend on the number of event types. In addition, the efficiency of this model increases by more training iterations (which was not performed in most simulations because of its time complexity). Hence, for this model's better performance, sufficient training must be conducted. Furthermore, this model can not capture much history dependency for larger networks, which decreases its accuracy in calculating intensities.

Considering the aforementioned deficiencies of the proposed model, the avenues for future work are (i) extending the model to consider more event types by devising a model independent of event types, (ii) extending the model to consider more history information, (iii) using methods such as attention mechanism to reflect Hawkes influence matrix, as used in [35], and (iv) using Wasserstein generative adversarial networks (WGANs) [48, 49] for modeling network interactions which have been used for modeling point processes but it requires to extend it to model the types of events in addition to their times.

Appendix

Vanilla Hawkes Inference Method

The parameters of the vanilla Hawkes model can be inferred using the EM method introduced in [37]. In this method, log-likelihood of all events in an interval $[0, T]$ is given by the following equation:

$$\begin{aligned}\mathscr{L} &= \sum_{a,b\in N}\left(\sum_{t_i\in I(a,b)}\log\left(\mu + \sum_{\substack{t_j\,\in\, I(a,b)t\\ j<t_i}}\alpha e^{-\beta\left(t_i-t_j\right)}\right) - \int_0^T\left(\mu + \sum_{t_j<s}\alpha e^{-\beta\left(s-t_j\right)}\right)\mathrm{d}s\right)\\ &= \Bigg(\sum_{a,b\in N}\log\Bigg(\mu + \sum_{\substack{t_j\,\in\, I(a,b)\\ t_j<t_i}}\alpha e^{-\beta\left(t_i-t_j\right)}\Bigg)\Bigg)\Bigg) - \mu T - \frac{\alpha}{\beta}\sum_{t_i<T}\left(1-e^{-\beta\left(T-t_i\right)}\right),\end{aligned} \tag{A.1}$$

where $I(a,b)$ represents all interactions between nodes a and b. This summation is performed over all node pairs of the network. By having $T\beta \gg 1$, expression $\sum_{t_i<T}(1-e^{-\beta(T-t_i)}) \approx n_{ab}$, where n_{ab} is the number of all events between nodes a and b. Therefore, we get

$$\mathscr{L} = \sum_{a,b\in N}\left(\sum_{t_i\in I(a,b)}\log\left(\mu + \sum_{\substack{t_j\in I(a,b)\\ \\ t_j<t_i}}\alpha e^{-\beta\left(t_i-t_j\right)}\right) - \mu T - \frac{\alpha}{\beta}n_{ab}\right). \tag{A.2}$$

Since maximizing the log-likelihood in (A.2) using gradients is difficult, an EM method introduced in [37] is used. We assume hidden variables p_{ij} indicating the probability of event i being triggered by event j. By this definition, the formula for log-likelihood changes into the form of

$$\begin{aligned}\mathscr{L} &= \sum_{a,b\in N}\left(\sum_{t_i\in I(a,b)} p_{ii}\log\mu + \sum_{\substack{t_i,t_j\,\in\, I(a,b)\\ t_j<t_i}} p_{ij}\log\left(\alpha e^{-\beta\left(t_i-t_j\right)}\right) - \mu T - \frac{\alpha}{\beta}n_{ab}\right)\\ &= \sum_{a,b\in N}\left(\sum_{t_i\in I(a,b)} p_{ii}\log\mu + \sum_{\substack{t_i,t_j\,\in\, I(a,b)\\ t_j<t_i}} p_{ij}\left(\log\alpha - \beta\left(t_i-t_j\right)\right) - \mu T - \frac{\alpha}{\beta}n_{ab}\right).\end{aligned} \tag{A.3}$$

According to the definition of p_{ij}, E-Step of EM algorithm can be written as

$$\begin{aligned} p_{ii} &= \frac{\mu^k}{\mu^k + \sum_{t_r<t_i}\alpha^k e^{-\beta^k\left(t_i-t_r\right)}},\\ p_{ij} &= \frac{\alpha^k e^{-\beta^k\left(t_i-t_j\right)}}{\mu^k + \sum_{t_r<t_i}\alpha^k e^{-\beta^k\left(t_i-t_r\right)}}.\end{aligned} \tag{A.4}$$

By determining derivatives of log-likelihood in (A.3) with respect to the parameters and setting them to zero, we obtain parameters for M-Step:

$$\begin{aligned} \mu &= \frac{2\times\sum_{a,b\in N}\sum_{t_i\in I(a,b)}p_{ii}}{T\times N\times(N-1)},\\ \alpha &= \frac{\left(\sum_{a,b\in N}\sum_{\substack{t_i,\,t_j\,\in\, I(a,b)\\ t_j<t_i}} p_{ij}\right)^2}{n_e\times\sum_{a,b\in N}\sum_{\substack{t_i,\,t_j\in I(a,b)\\ t_j<t_i}} p_{ij}\left(t_i-t_j\right)},\\ \beta &= \frac{\sum_{a,b\in N}\sum_{\substack{t_i,\,t_j\in I(a,b)\\ t_j<t_i}} p_{ij}}{\sum_{a,b\in N}\sum_{\substack{t_i,\,t_j\in I(a,b)\\ t_j<t_i}} p_{ij}\left(t_i-t_j\right)}.\end{aligned} \tag{A.5}$$

Data Availability

(i) The "Hospital" [42], "Hypertext" [43], "Primary School" [45, 46], and "Highschool" [47] datasets are publicly available on the website of SocioPatterns (https://www.sociopatterns.org/datasets). (ii) The "MIT Reality Mining" and "Enron Email" datasets supporting this manuscript are from previously reported studies and datasets, which have been cited. The original data of MIT Reality Mining [41] and Enron [44] datasets were preprocessed by the previous studies of [14, 16]. These datasets are publicly available from https://github.com/IdeasLabUT/CHIP-Network-Model/tree/master/storage/datasets. (iii) The synthetic data used to support the findings of this study can be obtained from the first author upon request (h.sdizaji@tabrizu.ac.ir).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] O. Michail and G. Paul, "Elements of the theory of dynamic networks," *Communications of the ACM*, vol. 61, p. 72, 2018.

[2] G. Rossetti, *Social network dynamics*, PhD thesis, University of Pisa, Pisa, Italy, 2015.

[3] J. Skarding, B. Gabrys, and K. Musial, "Foundations and modeling of dynamic networks using dynamic graph neural networks: a survey," *IEEE Access*, vol. 9, Article ID 79168, 2021.

[4] P. Holme and J. Saramäki, "Temporal networks," *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.

[5] T. A. B. Snijders, "The statistical evaluation of social network dynamics," *Sociological Methodology*, vol. 31, no. 1, pp. 361–395, 2001.

[6] T. A. Snijders, G. G. V. D. Bunt, C. E. Steglich, and G. Steglich, "Introduction to stochastic actor-based models for network dynamics," *Social Networks*, vol. 32, no. 1, pp. 44–60, 2010.

[7] C. Stadtfeld, J. Hollway, and P. Block, "Dynamic network actor models: investigating coordination ties through time," *Sociological Methodology*, vol. 47, no. 1, 40 pages, 2017.

[8] C. Stadtfeld and P. Block, "Interactions, actors, and time: dynamic network actor models for relational events," *Sociological Science*, vol. 4, no. 14, pp. 318–352, 2017.

[9] J. A. JoshuaLospinoso, M. Schweinberger, TomSnijders, and R. M. Ripley, "Assessing and accounting for time heterogeneity in stochastic actor oriented models," *Advances in Data Analysis and Classification*, vol. 5, no. 2, pp. 147–176, 2017.

[10] C. T. Butts and C. S. Marcum, "A relational event approach to modeling behavioral dynamics," pp. 51–92, 2017, https://arxiv.org/abs/1707.09902.

[11] C. T. Butts, “A relational event framework for social action,” *Sociological Methodology*, vol. 38, no. 1, pp. 155–200, 2008.

[12] U. Brandes, J. Lerner, A. Tom, and B. Snijders, “Networks evolving step by step: statistical analysis of dyadic event data,” in *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*, p. 200, Athens, Greece, July 2009.

[13] R. Leenders, N. S. Contractor, and L. A. De Church, “Once upon a time: understanding team processes as relational event networks,” *Organizational Psychology Review*, vol. 6, no. 1, pp. 92–115, 2016.

[14] C. DuBois, C. Butts, and P. Smyth, “Stochastic blockmodeling of relational event dynamics,” in *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, Volume 31 of Proceedings of Machine Learning Research*, pp. 238–246, Scottsdale, AZ, USA, May 2013.

[15] R. Junuthula, M. Haghdan, K. S. Xu, and V. Devabhaktuni, “The block point process model for continuous-time event-based dynamic networks,” in *Proceedings of the 2019 World Wide Web Conference*, pp. 829–839, New York, NY, USA, May 2019.

[16] M. Arastuie, S. Paul, and K. Xu, “Chip: a hawkes process model for continuous-time networks with scalable and consistent estimation,” in *Advances in Neural Information Processing Systems*vol. 33, Curran Associates, Inc, Article ID 16996, 2020.

[17] E. W. Fox, M. B. Short, F. P. Schoenberg, K. D. Coronges, and A. L. Bertozzi, “Modeling e-mail networks and inferring leadership using self-exciting point processes,” *Journal of the American Statistical Association*, vol. 111, no. 514, pp. 564–584, 2016.

[18] J. R. Zipkin, F. P. Schoenberg, K. Coronges, and A. L. Bertozzi, “Point-process models of social network interactions: parameter estimation and missing data recovery,” *European Journal of Applied Mathematics*, vol. 27, no. 3, pp. 502–529, 2016.

[19] P. O. Perry and P. J. Wolfe, “Point process modelling for directed interaction networks,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 75, no. 5, pp. 821–849, 2013.

[20] F. S. Passino and A. Nicholas, “Mutually exciting point process graphs for modelling dynamic networks,” 2022, https://arxiv.org/abs/2102.06527.

[21] S. M. Kazemi, R. Goel, K. Jain et al., “Relational representation learning for dynamic (knowledge) graphs: a survey,” vol. 21, pp. 1–73, 2020.

[22] A. Maheshwari, A. Goyal, and M. K. Hanawal, *Dyngan: Generative Adversarial Networks for Dynamic Network Embedding*, Springer-Verlag, Berlin, Germany, 2019.

[23] Ma Yao, Z. Guo, Z. Ren, J. Tang, and D. Yin, *Streaming Graph Neural Networks*Association for Computing Machinery, New York, NY, USA, 2020.

[24] J. Cao, X. Lin, X. Cong et al., “Deep structural point process for learning temporal interaction networks,” in *Proceedings of the Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference*, pp. 305–320, Berlin, Heidelberg, September 2021.

[25] H. Mei and J. Eisner, “The neural hawkes process: a neurally self-modulating multivariate point process,” in *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 6754–6764, Long Beach, CA, USA, December 2017.

[26] A. G. Hawkes, “Spectra of some self-exciting and mutually exciting point processes,” *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[27] Y. Ogata, “Statistical models for earthquake occurrences and residual analysis for point processes,” *Journal of the American Statistical Association*, vol. 83, pp. 9–27, 1988.

[28] Y. Ogata, “Space-time point-process models for earthquake occurrences,” *Annals of the Institute of Statistical Mathematics*, vol. 50, no. 2, pp. 379–402, 1998.

[29] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: first steps,” *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983.

[30] R. Rastelli and M. Fop, “A stochastic block model for interaction lengths,” *Advances in Data Analysis and Classification*, vol. 14, no. 2, pp. 485–512, 2020.

[31] K. S. Xu, “Stochastic block transition models for dynamic networks,” 2015, https://arxiv.org/abs/1411.5404.

[32] Z. Han, Y. Wang, Y. Ma, S. Günnemann, and V. Tresp, “Graph hawkes neural network for future prediction on temporal knowledge graphs,” in *Proceedings of the Automated Knowledge Base Construction*, Irvine, CA, USA, Febuary 2020.

[33] M. Yao, Z. Guo, Z. Ren, J. Tang, and D. Yin, “Streaming graph neural networks,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR’20*, pp. 719–728, New York, NY, USA, July 2020.

[34] M. Farajtabar, Y. Wang, M. Rodriguez, S. Li, H. Zha, and L. Song, “Coevolve: a joint point process model for information diffusion and network co-evolution,” *Journal of Machine Learning Research*, vol. 18, p. 7, 2015.

[35] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, “Joint modeling of event sequence and time series with attentional twin recurrent neural networks,” 2017, https://arxiv.org/abs/1703.08524.

[36] J. G. Rasmussen, “Lecture notes: temporal point processes and the conditional intensity function,” 2018, https://arxiv.org/abs/1806.00221.

[37] E. Lewis and M. George, “A nonparametric EM algorithm for multiscale Hawkes processes,” *Journal of Nonparametric Statistics*, vol. 1, 2011.

[38] R. H. Byrd, J. C. Gilbert, and J. Nocedal, “A trust region method based on interior point techniques for nonlinear programming,” *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.

[39] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] P. A. W. Lewis and G. S. Shedler, “Simulation of nonhomogeneous Poisson processes by thinning,” *Naval Research Logistics Quarterly*, vol. 26, no. 3, pp. 403–413, 1979.

[41] N. Eagle, A. S. Pentland, and D. Lazer, “Inferring friendship network structure by using mobile phone data,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, Article ID 15278, 2009.

[42] P. Vanhems, A. Barrat, C. Cattuto et al., “Estimating potential infection transmission routes in hospital wards using wearable proximity sensors,” *PLoS One*, vol. 8, no. 9, Article ID e73970, 2013.

[43] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J. F. Pinton, and W. Van den Broeck, “What’s in a crowd? analysis of face-to-face behavioral networks,” *Journal of Theoretical Biology*, vol. 271, no. 1, pp. 166–180, 2011.

[44] B. Klimt and Y. Yang, “The enron corpus: a new dataset for email classification research,” in *Machine Learning: ECML 2004*, Jean-François Boulicaut, Floriana Esposito,

Fosca Giannotti, and Dino Pedreschi, Eds., pp. 217–226, Springer, Berlin, Germany, 2004.

[45] V. Gemmetto, A. Barrat, and C. Cattuto, “Mitigation of infectious disease at school: targeted class closure vs school closure,” *BMC Infectious Diseases*, vol. 14, no. 1, 2014.

[46] J. Stehlé, N. Voirin, A. Barrat et al., “High-resolution measurements of face-to-face contact patterns in a primary school,” *PLoS One*, vol. 6, no. 8, Article ID e23176, 2011.

[47] J. Fournet and A. Barrat, “Contact patterns among high school students,” *PLoS One*, vol. 9, no. 9, Article ID e107878, 2014.

[48] A. Martin, S. Chintala, and L. . Bottou, “Wasserstein generative adversarial networks,” *Proceedings of the 34th International Conference on Machine Learning-Volume*, vol. 70, pp. 214–223, 2017.

[49] S. Xiao, M. Farajtabar, X. Ye, J. Yan, L. Song, and H. Zha, “Wasserstein learning of deep generative point process models,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3250–3259, Red Hook, NY, USA, December 2017.