Journal of Computational and Applied Mathematics 447 (2024) 115889

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A neural network based model for multi-dimensional non-linear Hawkes processes

Sobin Joseph, Shashi Jain *

Department of Management Studies, Indian Institute of Science, C V Raman Road, Bangalore, 560012, Karnataka, India

ARTICLE INFO

Keywords:
Non-linear Hawkes processes
Hawkes processes with inhibition
Neural networks for Hawkes process
Online learning for Hawkes processes

ABSTRACT

This paper introduces the Neural Network for Non-linear Hawkes processes (NNNH), a non-parametric method based on neural networks to fit non-linear Hawkes processes. Our method is suitable for analysing large datasets in which events exhibit both mutually-exciting and inhibitive patterns. The NNNH approach models the individual kernels and the base intensity of the non-linear Hawkes process using feed forward neural networks and jointly calibrates the parameters of the networks by maximizing the log-likelihood function. We utilize Stochastic Gradient Descent to search for the optimal parameters and propose an unbiased estimator for the gradient, as well as an efficient computation method. We demonstrate the flexibility and accuracy of our method through numerical experiments on both simulated and real-world data, and compare it with state-of-the-art methods. Our results highlight the effectiveness of the NNNH method in accurately capturing the complexities of non-linear Hawkes processes.

1. Introduction

A.G. Hawkes introduced the Hawkes process, which is a type of multivariate point process used to model a stochastic intensity vector that depends linearly on past events Hawkes [1]. This approach has been widely applied in various fields, such as seismology Ogata [2],Marsan and Lengline [3], financial analysis Filimonov and Sornette [4],Bacry et al. [5], social interaction modelling Crane and Sornette [6],Blundell et al. [7],Zhou et al. [8], criminology Mohler et al. [9], genome analysis Reynaud-Bouret et al. [10],Carstensen et al. [11], and epidemiology Park et al. [12],Chiang et al. [13].

The Hawkes process in its original form is linear, i.e., the intensities depend on past events through a linear combination of kernel functions. The primary concern in modelling the Hawkes process is estimating these kernel functions. A common practice has been to assume a parametric form for the kernel function, such as the exponential and power-law decay kernels, and then use maximum likelihood estimation (Ozaki [14]) to determine the optimal values of the parameters.

While easier to estimate, a parametric kernel function is often not expressive enough for real-world problems. A significant body of work is concerned with the non-parametric estimation of the kernel functions. Lewis and Mohler [15] propose a non-parametric, Expectation–Maximization (EM) based method to estimate the Hawkes kernel and the non-constant base intensity function. Zhou et al. [8] uses the method of multipliers and majorization-minimization approach to estimate the multivariate Hawkes kernels. Zhou et al. [16] present the MMEL algorithm, a non-parametric approach that, when combined with the EM technique, is employed for the estimation of Hawkes kernels. Bacry and Muzy [17] develop a non-parametric estimation based on the Wiener–Hopf equations for estimating the Hawkes kernels. Achab et al. [18] derive an integrated cumulants method for estimating the Hawkes kernel. Yang et al. [19] uses a non-parametric method incorporated with online learning technique to approximate the triggering kernels of the

* Corresponding author.
E-mail addresses: sobinjoseph@iisc.ac.in (S. Joseph), shashijain@iisc.ac.in (S. Jain).

https://doi.org/10.1016/j.cam.2024.115889
Received 12 September 2023; Received in revised form 1 March 2024
Available online 20 March 2024

Hawkes process. Xu et al. [20] proposed a sparse-group-lasso-based algorithm combined with likelihood estimation for estimating the Hawkes kernels. Joseph et al. [21] use a feed-forward network to model the excitation kernels and fit the network parameters using a maximum likelihood approach. Except for Bacry and Muzy [17], these non-parametric kernel estimation methods are generally restricted to the linear Hawkes processes.

Modelling the intensities as a linear combination of kernel functions imposes a non-negativity constraint on the kernel functions, which can be interpreted as excitation kernels. Non-negative kernel functions do not allow incorporating inhibitive effects while modelling the intensity process. Modelling inhibition, where the occurrence of an event reduces the intensity of future arrival, has drawn relatively less attention in the literature. However, inhibitory effects are prevalent in various domains. For example, in neuroscience, inhibitory kernel functions can represent the presence of a latency period before the successive activations of a neuron Reynaud-Bouret et al. [22].

This paper proposes a non-parametric estimation model for the non-linear Hawkes process. The non-linear Hawkes process allows the inclusion of both excitatory and inhibitory effects to model a broader range of phenomena. The stability condition of the non-linear Hawkes process is explored in Brémaud and Massoulié [23]. Reynaud-Bouret et al. [24], a non-parametric approach using histograms is introduced to model the kernel function of a non-linear Hawkes process. This method utilizes the least squares loss function for the purpose of optimization. Bonnet et al. [25] use a negative exponential function to model inhibitive kernels for a univariate non-linear Hawkes process and use maximum likelihood estimation to determine the optimal parameters. Bonnet et al. [26] extend this approach to a multivariate non-linear Hawkes process. Lemonnier and Vayatis [27] develop the Markovian Estimation of Mutually Interacting Process (MEMIP) method, which utilizes weighted exponential functions to determine kernels for the non-linear Hawkes process. To extend the MEMIP for large dimensional datasets, Lemonnier et al. [28] introduce dimensionality reduction features. However, the above approaches require the kernel function to be smooth, which is a drawback. Wang et al. [29] propose an algorithm that learns the non-linear Hawkes kernel non-parametrically using isotonic regression and a single-index model. The Isotonic Hawkes Process, however, assumes only a continuous monotonic decreasing excitation kernel to capture the influence of the past arrivals.

In their work, Du et al. [30] employ a recurrent neural network (RNN) to model event timings and markers simultaneously by leveraging historical data. Conversely, Mei and Eisner [31] propose a novel continuous-time long short-term memory (LSTM) model to capture the self-modulating Hawkes processes, capable of accounting for the inhibiting and exciting effects of prior events on future occurrences. Moreover, this approach can accommodate negative background intensity values, which correspond to delayed response or inertia of some events. While RNN-based models can capture complex long-term dependencies, inferring causal relationships or extracting causal information from these models can be challenging.

We propose a neural networks based non-parametric model for the non-linear Hawkes process where feed-forward networks are used to model the kernels and time-varying base intensity functions. The architecture of the neural network is chosen such that the likelihood function and its gradients with respect to the network parameters can be efficiently evaluated. As the likelihood function is non-convex with respect to the parameter space, we use the Stochastic Gradient Descent (SGD) with Adam (Kingma and Ba [32]) to obtain the network parameters that maximize the log-likelihood. The method is an extension of the Shallow Neural Hawkes model proposed by Joseph et al. [21], which was designed for linear Hawkes processes and only allows for the modelling of kernels with an excitation feature. We evaluate our model against state-of-the-art methods for non-linear Hawkes processes, using both simulated and real datasets.

The paper is structured as follows: Section 2 provides a definition of the non-linear Hawkes process and formulates the associated log-likelihood maximization problem. In Section 3, we introduce the proposed neural network model for the non-linear Hawkes process and discuss the parameter estimation procedure. Section 4 presents the results obtained by our method on synthetic data and its application to infer the order dynamics of bitcoin and ethereum market orders on the Binance exchange. Additionally, we apply our method to a high-dimensional neuron spike dataset. Finally, in Section 5, we summarize our findings and discuss the limitations of our method.

2. Preliminary definitions

Definition of the Hawkes process:. We denote a D-dimensional counting process as $(N_1(T), \dots, N_D(T))$, and its associated discrete events as $\mathcal{S} = (t_n, d_n)_{n\geq 1}$, where $t_n \in [0, T)$ is the timestamp of the nth event and $d_n \in (1, 2, \dots, D)$ is the dimension in which the nth event has occurred. Let $\{t_1^d, \dots, t_k^d, \dots, t_{N_d(T)}^d\}$, be the ordered arrivals for the dimensions $d = 1, \dots, D$. Given $t \geq 0$, the count of events in $[0, t)$ for the dth dimension will be

$$N_d(t) = \sum_{n\geq 1} \mathbb{1}_{t_n^d < t}.$$

The conditional intensity function for the counting process at t for the dth dimension, $\lambda_d^*(t) : \mathbb{R}^+ \to \mathbb{R}^+$, is given by,

$$\lambda_d^*(t) = \lim_{\Delta t\to 0} \frac{\mathbb{E}\left[N_d(t+\Delta t) - N_d(t)|\mathcal{H}_{t^-}\right]}{\Delta t}, \tag{1}$$

where, $\mathcal{H}_{t^-}$, denotes the history of the counting process up to time t.

For a D-dimensional Hawkes process the conditional intensity $\lambda_d(t)$, for the dth dimension is expressed as,

$$\lambda_d(t) = \mu_d(t) + \sum_{j=1}^{D} \sum_{\{\forall k|(t_k^j < t)\}} \phi_{dj}(t - t_k^j), \tag{2}$$

where $\mu_d(t)$, the exogenous base intensity for the dth dimension, does not depend upon the history of the past arrivals. $\phi_{dj}(t - t_k^j)$, $1 \leq d, j \leq D$, are called the excitation kernels that quantify the magnitude of excitation of the conditional intensity for the dth dimension at time t due to the past arrival at t_k^j, $\{\forall k | t_k^j < t\}$ in the jth dimension. These kernel functions are positive and causal, i.e., their support is in $\mathbb{R}^+$. Inferring a Hawkes process requires estimating the base intensity function μ_d and its kernels functions ϕ_{dj}, either by assuming a parametric form for the kernels or in a non-parametric fashion. A more generic form of the Hawkes process conditional intensity $\lambda_d^*(t)$, is given by,

$$\lambda_d^*(t) = \Psi_d(\lambda_d(t)),\ \forall d = 1, \dots, D, \tag{3}$$

where Ψ_d is the dependency function given by, $\Psi_d : \mathbb{R} \to \mathbb{R}^+$.

If Ψ_d is a identity function then, Eq. (3) is equivalent to the linear Hawkes process expressed in Eq. (2). However, if Ψ_d is a non-linear function, then the process is called a Non-linear Hawkes Process. For the stability and stationarity of the non-linear Hawkes process, Ψ_d should be Lipschitz continuous, as given by Theorem 7 of Brémaud and Massoulié [23]. The advantage of the non-linear Hawkes process is that it allows the kernel output to take negative values to model inhibitory effects as well as positive values to model excitatory effects, i.e., $\phi_{dj} : \mathbb{R}^+ \to \mathbb{R}$. Wang et al. [29] uses a sigmoid function $(1+e^{-x})^{-1}$ and a decreasing function $1 - (1+e^{-x})^{-1}$, while Bonnet et al. [25] and Costa et al. [33] use a $\max$ function as Ψ.

The log-likelihood function of non-linear Hawkes process. The categorization of methods used for estimating Hawkes processes, as discussed in Cartea et al. [34], can be broadly classified into three groups:

- *Maximum likelihood estimation (MLE)* It is the most commonly used approach for estimating the kernels of the Hawkes process, as employed by many methods such as Bonnet et al. [25], Du et al. [30], and Lemonnier et al. [28]. However, MLE methods have high computational costs as their time complexity increases quadratically with the number of arrivals. Expectation–maximization methods, such as those used by Lewis and Mohler [15], Zhou et al. [8], Lee et al. [35] and Zhou et al. [36], also belong to this category.
- *Method of moments* Wang et al. [29] and Achab et al. [18] use a moment matching method to estimate the Hawkes process. These methods typically rely on the spectral properties of the Hawkes process. The WH method (Bacry and Muzy [17]), which is used as a benchmark model in numerical experiments, estimates the Hawkes process as a system of Wiener–Hopf equations and also falls under this category.
- *Least squares estimation (LSE)* This method involves minimizing the L2 error of the integrated kernel. Cartea et al. [34] propose an efficient stochastic gradient-based estimation method for linear multivariate Hawkes processes, which applies to large datasets and general kernels. However, the method is not suitable for non-linear Hawkes process.

In this paper the optimal kernels are estimated through the maximum likelihood approach.

For a D-multidimensional non-linear Hawkes process with $\boldsymbol{\mu} = [\mu_d(t)]_{D\times 1}$ and $\boldsymbol{\Phi} = [\phi_{dj}(t)]_{D\times D}$; $1 \leq d, j \leq D$, let $\boldsymbol{\theta} = [\theta_1, \dots \theta_p \dots \theta_P]$ be the set of parameters used to model $\boldsymbol{\mu}$ and $\boldsymbol{\Phi}$. The parameters can be estimated by maximizing the log-likelihood function over the sampled events from the process. The log-likelihood (LL) function corresponding to the dataset $\mathcal{S}$ for the non-linear Hawkes process is given by (see for instance Rubin [37], Daley and Vere-Jones [38]),

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{S}) = \sum_{d=1}^{D} \left[\sum_{(t_n, d_n)\in\mathcal{S}} \left\{ \log(\lambda_d^*(t_n)) - \int_{t_{n-1}^d}^{t_n^d} \lambda_d^*(s) ds \right\} \mathbb{1}_{d_n = d} \right]. \tag{4}$$

Depending on the parametric form of the kernel, $\phi_{dj}(t)$, the LL function may or may not be concave. For instance, even for the exponential kernel, $\phi_{dj}(t) = \alpha_{dj} \exp(-\beta_{dj})(t)$, the LL function is not concave in the parameter space. We use the Stochastic Gradient Descent (SGD) method to search the local optima for the LL function in the parameter space. In order to estimate the parameters $\boldsymbol{\theta}$ using the SGD we need an unbiased estimator of the gradient of the LL, as given by Eq. (4), with respect to θ.

$$\begin{aligned} \left(\nabla_{\theta_p} \mathcal{L}\right)(\boldsymbol{\theta}, \mathcal{S}) &= \nabla_{\theta_p} \left(\sum_{d=1}^{D} \left[\sum_{(t_n, d_n)\in\mathcal{S}} \left\{ \log(\lambda_d^*(t_n)) - \int_{t_{n-1}^d}^{t_n^d} \lambda_d^*(s) ds \right\} \mathbb{1}_{d_n = d} \right] \right) \\ &= \sum_{d=1}^{D} \left[\sum_{(t_n, d_n)\in\mathcal{S}} \nabla_{\theta_p} \left(\left\{ \log(\lambda_d^*(t_n)) - \int_{t_{n-1}^d}^{t_n^d} \lambda_d^*(s) ds \right\} \right) \mathbb{1}_{d_n = d} \right]. \end{aligned} \tag{5}$$

Following Eq. (5), an unbiased estimator of the gradient of $\mathcal{L}$ with respect to θ_p will be,

$$\nabla_{\theta_p} \widehat{\mathcal{L}}(\boldsymbol{\theta}, t_n^d) := \nabla_{\theta_p} \left(\log(\lambda_d^*(t_n^d)) - \int_{t_{n-1}^d}^{t_n^d} \left(\lambda_d^*(s)\right) ds \right), \tag{6}$$

where t_n^d is randomly sampled from $\mathcal{S}$. The proposed model does not assume a specific parametric form for the kernels and the base intensities. Instead, it asssumes a neural network model for both components. The model and its estimation is discussed in Section 3.

3. Proposed model

Hornik et al. [39] show that a multilayer feed-forward networks with as few as one hidden layer are capable of universal approximation to a given level of precision and accuracy. We here propose a neural network-based system called the Neural Network

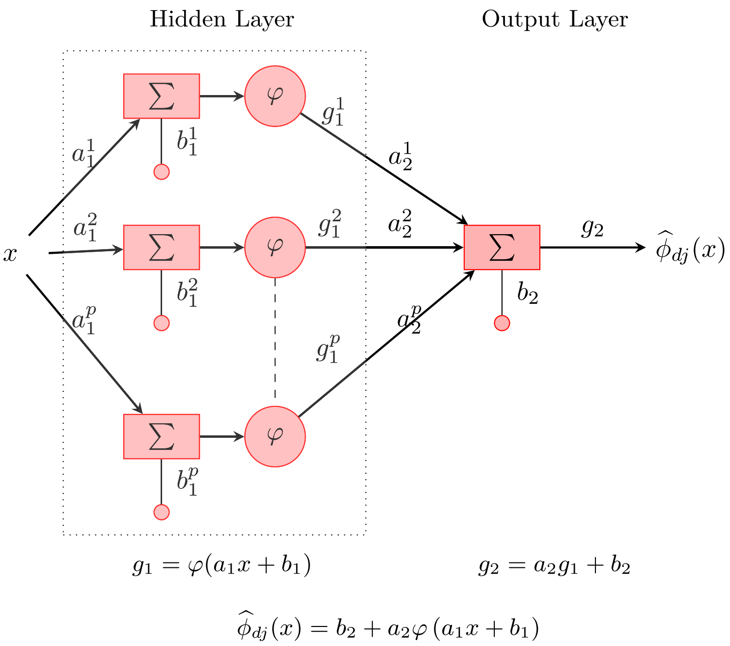


Fig. 1. The architecture of the feed-forward neural network used for modelling kernels and base intensity functions of the non-linear Hawkes process in NNNH. Here φ is the activation function, which is taken as ReLU.

for Non-Linear Hawkes (NNNH) to estimate the kernels and the base intensity of the non-linear Hawkes process. Each kernel, ϕ_{dj}, $1 \leq d, j \leq D$, and base intensity function μ_d, is separately modelled as a feed-forward network. However, the parameters of all the networks are jointly estimated by maximizing the log-likelihood of the training dataset.

In the NNNH, each kernel ϕ_{dj}, $1 \leq d, j \leq D$ is modelled as a feed-forward neural network $\hat{\phi}_{dj} : \mathbb{R}^+ \to \mathbb{R}$,

$$\hat{\phi}_{dj}(t) = A_2 \circ \varphi \circ A_1, \tag{7}$$

where A_1: $\mathbb{R} \to \mathbb{R}^p$ and A_2: $\mathbb{R}^p \to \mathbb{R}$. More precisely,

$$A_1(x) = \mathbf{W}_1 x + \mathbf{b_1} \quad \text{for } x \in \mathbb{R},\ \mathbf{W}_1 \in \mathbb{R}^{p\times 1}, \mathbf{b_1} \in \mathbb{R}^p,$$

$$A_2(\mathbf{x}) = \mathbf{W}_2\mathbf{x} + b_2 \quad \text{for } \mathbf{x} \in \mathbb{R}^p,\ \mathbf{W}_2 \in \mathbb{R}^{1\times p}, b_2 \in \mathbb{R},$$

and $\varphi : \mathbb{R}^j \to \mathbb{R}^j, j \in \mathbb{N}$ is the component-wise Rectified Linear Unit (ReLU) activation function given by:

$$\varphi(x_1, \dots, x_j) := \left(\max(x_1, 0), \dots, \max(x_j, 0)\right).$$

We take $\mathbf{W}_1 = [a_1^1, a_1^2, \dots ., a_1^p]^\top$, $\mathbf{W}_2 = [a_2^1, a_2^2, \dots ., a_2^p]$ and $\mathbf{b_1} = [b_1^1, b_1^2, \dots ., b_1^p]^\top$. Therefore, the kernel function can be written as:

$$\hat{\phi}_{dj}(x) = b_2 + \sum_{i=1}^{p} a_2^i \max\left(a_1^i x + b_1^i, 0\right) \tag{8}$$

As the kernel, $\hat{\phi}_{dj}$, maps from positive real numbers to real numbers, it is versatile in its ability to model both inhibitory and excitatory effects.

With a choice of p neurons for the hidden layer, the dimension of the parameter space for the above network will be $3p+1$. For a D-dimensional non-linear Hawkes process there will be D^2 kernels and the total number of parameters to be estimated would be $(3p+1)D^2$. In case the base intensity is not constant, we also model it as a feed-forward neural network, with an architecture similar to the ones used for the kernels. We then need to approximate in total $(3p+1)\left(D^2 + D\right)$ parameters. If the base intensity is assumed to be a constant, the total number of parameters to be estimated would be $(3p+1)D^2 + D$. The Fig. 2 is a schematic representation of the NNNH model, with each ϕ_{dj}, and μ_d being a network (as expressed in Fig. 1). In order to estimate $\lambda_d^*(t)$, we need in addition to t, the history of arrivals until t, i.e., $\mathcal{H}_t^-$. The NNNH therefore estimates $\lambda_d^*(t)$ as:

$$\hat{\lambda}_d^*(t) = \max\left(\hat{\mu}_d(t) + \sum_{j=1}^{D} \sum_{\{\forall k|(t_k^j<t)\}} \hat{\phi}_{dj}(t - t_k^j), 0\right). \tag{9}$$

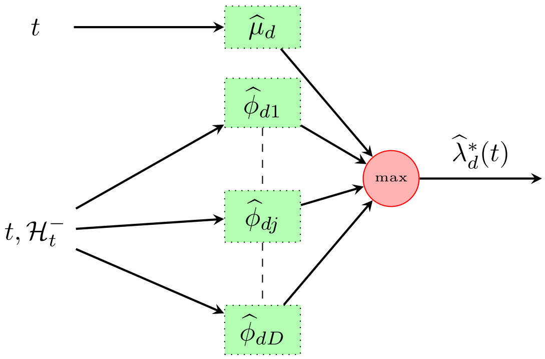


Fig. 2. NNNH model for multivariate non-linear Hawkes process.

The outer max function ensures that the estimated $\lambda_d^*(t)$ is positive and mimics the Ψ function in the non-linear Hawkes process, as given in Eq. (3).

This non-parametric method employs a feed-forward neural network that can precisely estimate any continuous kernel function and base intensity function defined on a compact set, with a desired level of accuracy (Leshno et al. [40]). However, the challenge is to jointly estimate the parameters of the $D^2 + D$ neural networks used to model the base intensity and the kernel functions. To do this, the set of parameter values that maximize the log-likelihood over the observed dataset $\mathcal{S}$ is found. Since the log-likelihood function is non-convex in the parameter space, the batch stochastic gradient descent is used to estimate the parameter values that provide a local maximum of the log-likelihood function. The gradient of the log-likelihood function as given by Eq. (6) with respect to each network parameter is computed for updating the parameter values in the batch stochastic gradient method.

3.1. Estimating the parameters of the NNNH

The challenge for the NNNH model is, given the dataset $\mathcal{S}$ to efficiently obtain the parameters of the networks that locally maximize the log-likelihood function. More precisely, let $\boldsymbol{\theta} = [\theta_1, \ldots \theta_p \ldots \theta_P]^\top$ denote the parameters of the network and let $\boldsymbol{\Theta}$ denote the parameter space; then we want to find, for given dataset $\mathcal{S}$, the values of the parameters that maximize the log-likelihood function over the parameter space, i.e.,

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathcal{L}\left(\boldsymbol{\theta}, \mathcal{S}\right). \tag{10}$$

Gradient descent is an effective optimization technique when the log-likelihood function can be differentiated with respect to its parameters. This is because computing the first-order partial derivatives of all parameters has the same computational complexity as evaluating the log-likelihood function, making it a relatively efficient approach. With neural networks, stochastic gradient descent, has been shown to be more effective and efficient method for optimization. The NNNH parameter estimation algorithm utilizes Adam (Kingma and Ba [32]), a stochastic gradient descent technique that adjusts learning rates adaptively and relies solely on first-order gradients. Eq. (6) provides the necessary unbiased estimator of the log-likelihood function for the NNNH method to be used with the SGD. Algorithm 1 gives the pseudo-code for the NNNH parameter estimation using Adam.

The essential step while estimating the parameters for the NNNH model is computing the unbiased gradient of the likelihood function, i.e.,

$$\nabla_{\theta_p} \widehat{\mathcal{L}}(\boldsymbol{\theta}, t_n^d) := \nabla_{\theta_p} \left(\log(\widehat{\lambda}_d^*(t_n^d)) - \int_{t_{n-1}^d}^{t_n^d} \widehat{\lambda}_d^*(s) ds \right).$$

This in turn involves computing the following gradients,

$$\nabla_{\theta_p} \log(\widehat{\lambda}_d^*(t_n^d)) \text{ and } \nabla_{\theta_p} \int_{t_{n-1}^d}^{t_n^d} \left(\widehat{\lambda}_d^*(s)\right) ds.$$

While it is possible to compute the former gradient efficiently, the challenge lies in accurately estimating the latter gradient. The former gradient calculation involves the following steps:

$$\begin{aligned}
\nabla_{\theta_p} \log(\widehat{\lambda}_d^*(t_n^d)) &= \frac{1}{\widehat{\lambda}^*(t_n^d)} \nabla_{\theta_p} \max \left(\widehat{\mu}_d(t_n^d) + \sum_{j=1}^{D} \sum_{\{\forall k | (t_k^j < t_n^d)\}} \widehat{\phi}_{dj}(t_n^d - t_k^j), 0 \right), \\
&= \frac{1}{\widehat{\lambda}^*(t_n^d)} \left(\nabla_{\theta_p} \left(\widehat{\mu}_d(t_n^d)\right) + \sum_{j=1}^{D} \sum_{\{\forall k | (t_k^j < t_n^d)\}} \nabla_{\theta_p} \left(\widehat{\phi}_{dj}(t_n^d - t_k^j) \right) \right) \mathbb{1}_{\widehat{\lambda}_d(t_n^d) > 0}.
\end{aligned}$$

Algorithm 1: Estimation of parameters of NNNH using Adam.

```
Input: dataset $\mathcal{S}$
Input: learning rate $\alpha = 0.01$, decay rates $\beta_1 = 0.9, \beta_2 = 0.999$, small constant $\epsilon = 10^{-8}$
Output: Optimal parameter value $\hat{\theta}$
initialize parameter vector $\boldsymbol{\theta_0}$;
$m_0 \leftarrow 0$ (initialize first moment vector);
$v_0 \leftarrow 0$ (initialize second moment vector);
initialize iteration step $i = 0$;
while stopping criterion not met do
    $i \leftarrow i + 1$
    randomly sample $(t_i, d_i) \in \mathcal{S}$
    $g_i \leftarrow \nabla_{\theta}\hat{\mathcal{L}}(\boldsymbol{\theta}_{i-1}, t_i^d)$ compute unbiased gradient of the likelihood function
    $m_i \leftarrow \beta_1 m_{i-1} + (1 - \beta_1)g_i$ update biased first moment estimate
    $v_i \leftarrow \beta_2 v_{i-1} + (1 - \beta_2)g_i^2$ update biased second moment estimate
    $\hat{m}_i \leftarrow \frac{m_i}{1-\beta_1^i}$ compute bias-corrected first moment estimate
    $\hat{v}_i \leftarrow \frac{v_i}{1-\beta_2^i}$ compute bias-corrected second moment estimate
    $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_{i-1} - \alpha \frac{\hat{m}_i}{\sqrt{\hat{v}_i + \epsilon}}$ update parameters
end
```

Here,

$$\hat{\lambda}_d^*(t) = \max\left(\hat{\lambda}_d(t), 0\right).$$

In Appendix A, you can find the expressions for the gradients of the neural networks $\hat{\mu}_d(t)$ and $\hat{\phi}_{dj}(t)$. In Section 3.2 we provide an approach for efficient estimation of $\nabla_{\theta_p} \int_{t_{n-1}^d}^{t_n^d} \left(\hat{\lambda}_d^*(s)\right) ds$.

3.2. Gradient of the integrated Hawkes intensity function

The computation of the first order derivatives of,

$$\hat{\Lambda}_d(t_n^d) := \int_{t_{n-1}^d}^{t_n^d} \hat{\lambda}_d^*(s) ds,$$

with respect to all the parameters will have the same computational complexity as evaluating the function itself. In the SGD method, it is crucial to perform efficient computations of $\hat{\Lambda}_d(t_n^d)$ since these gradients must be computed multiple times. Gaussian Quadrature is a numerical technique that can be used to estimate the definite integral, but it is limited to well-behaved integrands that can be approximated by a certain degree of polynomial. Unfortunately, the non-linear Hawkes intensity function lacks smoothness, making the use of Gaussian Quadrature prone to substantial error. Moreover, Gaussian Quadrature entails computing the quadrature points and weights, which can be time-consuming and resource-intensive. The number of quadrature points required for accurate approximation also grows quickly with the polynomial degree, resulting in increased computational cost.

Our selection of network architecture can enable efficient evaluation of the integral $\hat{\Lambda}_d(t_n^d)$ in the NNNH. The above integral can be expanded as:

$$\begin{aligned}
\int_{t_{n-1}^d}^{t_n^d} \hat{\lambda}_d^*(s) ds &= \int_{t_{n-1}^d}^{t_n^d} \max\left(\hat{\lambda}_d(s), 0\right) ds \\
&= \int_{t_{n-1}^d}^{t_n^d} \max\left(\left[\hat{\mu}_d(s) + \sum_{j=1}^{D} \sum_{\{t_k^j < s\}} \hat{\phi}_{dj}(s - t_k^j)\right], 0\right) ds \\
&= \int_{t_{n-1}^d}^{t_n^d} \max\left(\left[\hat{\mu} + \sum_{j=1}^{D} \sum_{\{t_k^j < s\}} \sum_{i=1}^{p} a_2^i \max\left(a_1^i (s - t_k^j) + b_1^i, 0\right)\right], 0\right) ds,
\end{aligned}$$

with $\hat{\phi}_{dj}(s)$ substituted using Eq. (8) and $\hat{\mu}_d(s)$ set as a constant for simplicity in the third line. In addition, we set the bias b_2 to zero for ease of exposition. Therefore, the above integral takes the following form:

$$\int_{t_{n-1}^d}^{t_n^d} \hat{\lambda}_d^*(s) ds = \int_{t_{n-1}^d}^{t_n^d} \max\left(\left[\hat{\mu} + \sum_{\{k,i,j\}} a_2^i \max\left(a_1^i (s - t_k^j) + b_1^i, 0\right)\right], 0\right) ds. \tag{11}$$

Let $\mathbf{X} \equiv \{x_1, \dots, x_{P^*}\}$ be the sequence of *zero-crossings* obtained for all the P^* neurons, where the zero-crossing for the ith neuron is obtained by solving:

$$a_1^i(x_i - t_k^j) + b_1^i = 0 \qquad (12)$$
$$x_i = t_k^j - \frac{b_1^i}{a_1^i}.$$

P^* represents the total number of neurons used in the neural networks for modelling the kernels, $\hat{\phi}_{dj}$ with $1 \le d, j \le D$, of a D-dimensional non-linear Hawkes process. For example, if each kernel in a D-dimensional non-linear Hawkes process is modelled using a network with P neurons, then $P^* = PD^2$.

We re-index the sequence $\mathbf{X}$ in monotone increasing order as $\mathbf{S} \equiv \{s_1 \le s_2 \le \cdots \le s_{P^*}\}$. Let $\mathbf{S}^* \equiv \{s_l \le \cdots \le s_u\}$, where $t_{n-1}^d \le s_l \le \cdots \le s_u \le t_n^d$, and $1 \le l \le u \le P^*$, be the largest subsequence of the sorted sequence $\mathbf{S}$. Therefore, $\mathbf{S}^*$ is the subset of $\mathbf{S}$, with all the zero-crossings that lie between the range $[t_{n-1}^d, t_n^d]$.

We then write,

$$\int_{t_{n-1}^d}^{t_n^d} \hat{\lambda}_d^*(s)ds = \int_{t_{n-1}^d}^{s_l} \max\left(\hat{\lambda}_d(s), 0\right) ds + \cdots \int_{s_{q-1}}^{s_q} \max\left(\hat{\lambda}_d(s), 0\right) ds \ldots + \int_{s_u}^{t_n^d} \max\left(\hat{\lambda}_d(s), 0\right) ds,$$

and exploit the fact that $\hat{\lambda}_d(s)$ will be linear in s in the sub-intervals, $[t_{n-1}^d, s_l], \dots, [s_u, t_n^d]$. If $\hat{\lambda}_d(s)$ is linear in s in the interval $[s_{q-1}, s_q]$, then there will be at most one zero crossing for the function $\max(\hat{\lambda}_d(s), 0)$ that lies within this interval. If s_q^{out} is the zero-crossing for $\max(\hat{\lambda}_d(s), 0)$ and $s_{q-1} \le s_q^{\text{out}} \le s_q$, we can write:

$$\int_{s_{q-1}}^{s_q} \max\left(\hat{\lambda}_d(s), 0\right) ds = \left(\int_{s_{q-1}}^{s_q^{\text{out}}} \hat{\lambda}_d(s)ds\right) \mathbb{1}_{\hat{\lambda}_d(s_{q-1})>0} + \left(\int_{s_q^{\text{out}}}^{s_q} \hat{\lambda}_d(s)ds\right) \mathbb{1}_{\hat{\lambda}_d(s_q)>0}.$$

In case $s_q^{\text{out}} \notin [s_{q-1}, s_q]$ the above can be integrated as:

$$\int_{s_{q-1}}^{s_q} \max\left(\hat{\lambda}_d(s), 0\right) ds = \left(\int_{s_{q-1}}^{s_q} \hat{\lambda}_d(s)ds\right) \mathbb{1}_{\hat{\lambda}_d(s_{q-1})>0}.$$

Therefore, we evaluate $\hat{\Lambda}_d(t_n^d)$, by splitting the integral into intervals within which $\max(\hat{\lambda}_d(s), 0)$ is linear in s. The integral,

$$\int_{t_{n-1}^d}^{t_n^d} \hat{\lambda}_d^*(s)ds,$$

as well as its gradients with respect to the network parameters can then be exactly evaluated. The expression of the above integral in a sub-interval $[s_{q-1}, s_q]$, where $\hat{\lambda}_d^*(s)$ is linear is provided in the Appendix B.

Remark 1. While both the SNH model in [21] and the proposed model (NNNH) use a 2-layered feed-forward neural network to model the kernels, the network architecture differs in two aspects. First, the activation functions for the outer layer of the network in the SNH and the NNNH are exponential and identity functions, respectively. This allows SNH to admit only excitation kernels, and the NNNH could have kernels that could take negative values. The second difference is that the kernels in the SNH contribute linearly to the conditional intensity. In contrast, in the NNNH, the aggregation of the contribution of individual kernels to the conditional intensity is non-linear so that the intensity is positive in Eq. (9). An immediate implication of the latter difference is that for the SNH, the term $\int \lambda_d^*(s)ds$ in the log-likelihood function can be decoupled into the integrals of the individual kernels. However, in the case of the NNNH, due to the outer max function, the integral cannot be written as the sum of the integral of the individual kernels. This makes the computation of the gradient of the likelihood function for the NNNH model computationally challenging. An efficient approach for computing the integral and its gradient is discussed in Section 3.2.

4. Experiments and results

In this section, we evaluate the effectiveness of the NNNH method on synthetic and real-world datasets.[1] We assess the performance of the NNNH method in estimating the conditional intensity function of a non-homogeneous Poisson Processes (NHPP). We then showcase the adaptability of the NNNH method by utilizing simulated data of one-dimensional non-linear Hawkes processes. Specifically, we explore the NNNH's capacity to estimate Hawkes processes with smooth, non-smooth, and negative kernels, and examine its robustness in handling different variants of non-linear Hawkes models. Additionally, we investigate the NNNH's ability to estimate kernels and base intensity function for data simulated from a multidimensional non-linear Hawkes processes, where we evaluate a simultaneous combination of smooth, non-smooth, and inhibitive kernels.

We investigate the practical applications of non-linear Hawkes models through two case studies. In the first case, we apply the NNNH method to analyse tick-by-tick cryptocurrency trading data on the Binance exchange, seeking to identify any causal

[1] The source code and the dataset used in the experiments described here are available at https://github.com/sobin-joseph/NNNH.

relationships between the buy and sell market orders of the BTC/USD and the ETH/USD pairs. For the second case, we utilize the NNNH to analyse neuronal spike trains recorded from the motor cortex of a monkey, revealing the interdependence between individual neurons and how they function in tandem.

Prior to delving into the outcomes of our numerical experiments, we provide a brief overview of the data preprocessing steps and the hyper-parameters selected for fitting the NNNH model to a dataset. Specifically we discuss the scaling approach adopted for the dataset, the choice of the number of neurons opted for the network, the initial learning rates employed for the Adam optimizer, and the stopping criteria for the optimizer.

4.1. Data preprocessing and choice of hyper-parameters:

For all our experiments, we divide the dataset into training, validation, and test set. The dataset is partitioned in a 60:20:20 ratio for analysis. Due to the dependence on history in the Hawkes process, we do not alter the chronology of the events. Before splitting the dataset, we first scale the timestamps. Scaling the dataset is critical step in the preprocessing stage of building neural networks as it can enhance their performance, stability, and convergence. Additionally, scaling helps to ensure that all input features have a similar range of variance, which can lead to improved initialization of the network parameters and overall performance during training. Given dataset $\mathcal{S} = (t_n, d_n)$, where $t_n \in [0, T)$, let $\{t_1^d, \dots, t_k^d, \dots, t_{N_d(T)}^d\}$, be the ordered arrivals for the dimensions $d = 1, \dots, D$. We define $T_{\max}$ as

$$T_{\max} = \max\left(t^1_{N_1(T)}, \dots, t^d_{N_d(T)}, \dots, t^D_{N_D(T)}\right),$$

and $N(T_{\max})$ as:

$$N(T_{\max}) = \sum_{d=1}^{D} N_d(T_{\max}).$$

We scale the original timestamps as:

$$\hat{t}_n^d = t_n^d \frac{N(T_{\max})}{T_{\max}}.$$

Initialization of network parameters and choice of batch size. In the NNNH method as discussed in Section 3 we model each kernel and base intensity function as a feed-forward neural network. To model the base intensity functions, $\hat{\mu}_d(t)$, we use a feed-forward network with fifty neurons in the hidden layer. The following initialization is used for the parameters of the network,

$$\begin{aligned} a_1^i &\sim \mathcal{U}(-10^{-3}, 10^{-3}), \\ a_2^i &\sim \mathcal{U}(0, 0.2), \\ b_1^i &\sim \mathcal{U}(-1, 1), \end{aligned}$$

where $\mathcal{U}(a, b)$, denotes uniform distribution between a and b.

For the kernels, $\hat{\phi}_{dj}(t)$, $1 \leq d, j \leq D$, of the non-linear Hawkes process we use feed-forward networks with thirty two neurons in the hidden layer. The selection of thirty two neurons is guided by the hyperparameter tuning process we conduct, which is elaborated upon in Section 4.5. The network parameters of the kernel are initialized using the following scheme,

$$\begin{aligned} a_1^i &\sim \mathcal{U}(0, -0.3), \\ a_2^i &\sim \mathcal{U}(0, 0.2), \\ b_1^i &\sim \mathcal{U}(0, 0.3). \end{aligned}$$

If we use a constant baseline intensity we initialize $\hat{\mu} = 1$.

We use a batch size of hundred for calculating the stochastic gradient. We use different learning rates for updating the parameters in the hidden layer and the output layer, as we observe faster convergence with this choice. The following learning rates were used for all the experiments: For fitting the networks used to model the base intensity function we used a learning rate of 10^{-3} for updating the network parameters in the output layer and 10^{-6} for updating the parameters in the hidden layer. For the networks used to model the kernel function the we used corresponding learning rates of 10^{-2} and 10^{-3} for the output and hidden layers, respectively.

Stopping criteria:. In this study, for all the experiments, we utilize a stopping criterion based on the negative log-likelihood value computed on the validation dataset. The negative log-likelihood values are computed at the end of each iteration of the batch stochastic gradient descent algorithm. The NNNH parameter estimation algorithm stops when the updated parameters fail to improve the best recorded validation error for a specified number of iterations, following the approach proposed by Goodfellow et al. [41]. This early stopping criterion helps prevent over fitting, as demonstrated in Fig. 3, where we observe a reduction in training negative log-likelihood value while the validation negative log-likelihood value starts increasing after a certain number of iterations, indicating that the model is starting to over-fit the training data.

Through numerical experiments, Section 4.5 presents our investigation of the sensitivity of NNNH estimation to various hyper-parameter choices.

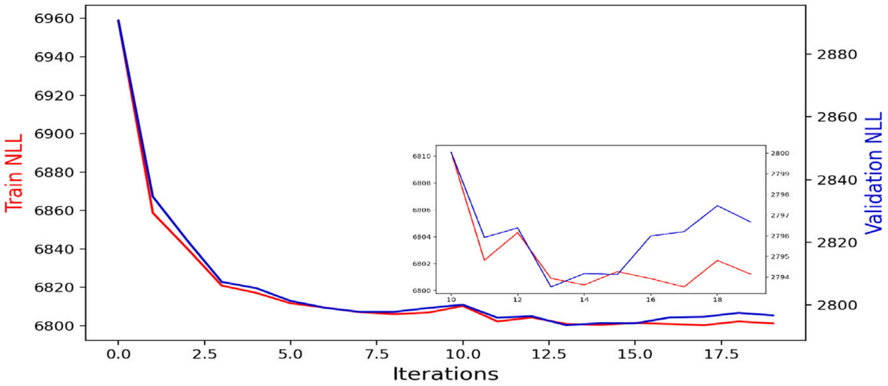


Fig. 3. Plot of Negative log-likelihood for train and validation datasets in one dimensional case, indicating the stopping criteria.

Table 1
The NHPP functions and their corresponding True Negative Log-Likelihood(TNLL) and NNNH Negative Log-Likelihood(NNLL) obtained from the NNNH method.

Function	Underlying equation	TNLL	NNLL
Exponential	$\mu(t) = 0.5e^{-0.001t}$	1177	1173
Linear	$\mu(t) = 0.5$	2528	2518
Parabola	$\mu(t) = 2 \times 10^{-7}(1.5t - 2000)$	1703	1699
Sin Curve	$\mu(t) = 0.4(\sin(2\pi t \times 0.0004 - 1000) + 1.1)$	3482	3538

4.2. Estimation of non homogeneous Poisson process intensity

A Non-Homogeneous Poisson Process (NHPP) $N(t), t \geq 0$ is a generalized counting process in which the rate at which events occurs varies with time. The intensity function, denoted by $\mu(t) : \mathbb{R}^+ \to \mathbb{R}^+$, captures the varying rate of the events over time, which could be influenced by factors such as external events, seasonal patterns, or other underlying phenomena. To comprehend this counting process, estimating the intensity function is necessary. The intensity function of NHPP can be estimated parametrically or non-parametrically. Parametric estimation assumes that the parametric form of the underlying intensity function is known. For instance, Rigdon and Basu [42] estimates the parameters by assuming the intensity function as power law function, while Lee et al. [43] uses a general exponential-polynomial-trigonometric intensity function. If one does not assume the parametric form of the intensity function, they would resort to non-parametric techniques to estimate the conditional intensity function. In Leemis [44], a technique is proposed that utilizes a piecewise linear function to estimate the cumulative intensity of the NHPP, while Xiao and Dohi [45] presents another non-parametric approach that employs a wavelet-based estimation method to estimate the intensity function.

In this section, we show that the NNNH technique can be used to model the intensity function, $\mu(t)$, of a non-homogeneous Poisson process. A single-layered feed-forward neural network can be used to model the intensity function as,

$$\hat{\mu}(t) = \max\left(b_2 + \sum_{i=1}^{p} a_2^i \max(a_1^i t + b_1^i, 0), 0\right), \tag{13}$$

where, as explained in Section 3, $[a_1^i, b_1^i, a_2^i, b_2]$, $1 \leq i \leq p$, are the parameters of the network that are estimated by maximizing the log-likelihood value,

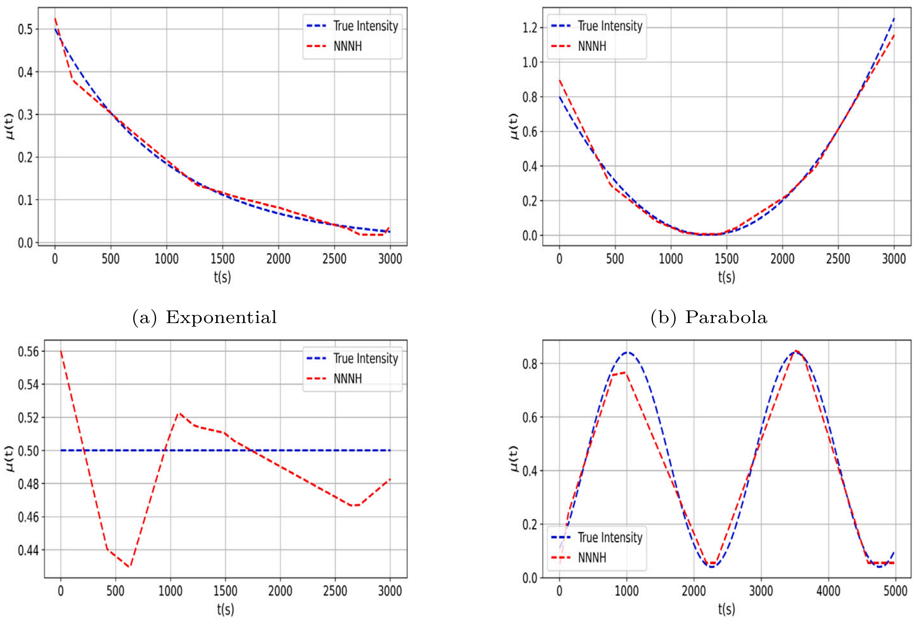
$$\mathcal{L}(\theta, \mathcal{S}) = \sum_{t_n \in \mathcal{S}} \log(\hat{\mu}(t_n)) - \int_0^T \hat{\mu}(s) ds, \tag{14}$$

for the training dataset $\mathcal{S}$. We use the SGD with Adam, for adaptive learning rates, to find the optimal network parameters, as discussed in Section 3.1.

We consider an exponential, linear, polynomial, and a trigonometric function, as given in Table 1, to model the conditional intensity. The arrival times are simulated using the thinning algorithm of Lewis and Shedler [46]. Table 1 compares the true negative log-likelihood of the simulated NHPP with the negative log-likelihood values obtained using the fitted NNNH model.[2]

We see that for all the cases considered, the estimated negative log-likelihood values are reasonably close to the true negative log-likelihood values. Fig. 4 compares the NNNH estimates of the intensity function with the true intensity function. We see for all the cases considered the NNNH is able to recover the intensity function reasonably well.

[2] The source code and the dataset used in the experiments described here are available at https://github.com/sobin-joseph/NNNH.



(a) Exponential

(b) Parabola

(c) constant

(d) Sine

Fig. 4. NNNH estimated kernel and base intensity for a one-Dimensional Hawkes process with sin base intensity and exponential kernel.

4.3. Univariate hawkes estimation

This section focuses on examining the effectiveness of the NNNH method in estimating one-dimensional non-linear Hawkes processes using the following criteria:

- Estimation of non-smooth kernels.
- Estimation of negative kernels.
- Estimation of non-linear Hawkes processes for different variations of Ψ.
- Estimation of Hawkes processes with varying base intensity function.

We simulate the arrival times for the different variants of the Hawkes processes using the *Ogata's thinning algorithm* proposed in Ogata [47]. We compare the performance of our algorithm to the following state of the art methods:

- WH[3]: An algorithm proposed in Bacry and Muzy [17], a non-parametric estimation method which solves a Wiener Hopf system derived from the auto-covariance of the multivariate Hawkes processes. This method demonstrates the capability to approximate both excitation and inhibitive kernels. Nevertheless, we observe that the outcomes derived from WH display a certain level of noise, which can pose challenges in terms of interpretation.
- Bonnet: A maximum likelihood based estimation method for Hawkes processes with self excitation or inhibition as proposed in Bonnet et al. [25] for one dimensional Hawkes and Bonnet et al. [26] for multi dimensional Hawkes. This parametric method assumes Hawkes kernels to follow a negative exponential function, which restricts its applicability to estimating only exponential or negative exponential kernels. Consequently, it may not be suitable for estimating other types of Hawkes kernels with different functional forms. While this method offers advantages in terms of simplicity and computational efficiency, its limitations in accommodating more diverse kernel shapes should be taken into account.
- EM: A non-parametric method proposed by Lewis and Mohler [15], which utilizes the EM(Expectation–Maximization) algorithm to estimate the Hawkes kernels and base intensity rates. This method is known for its high stability in estimating excitation kernels. However, it faces limitations in estimating negative kernels due to the probabilistic interpretation of the kernels. Moreover, the convergence ability of the EM method decreases when dealing with Hawkes kernels that exhibit slow decreasing behaviour, such as power law functions(Bacry and Muzy [17]).

[3] For both WH and EM we utilize a python library named "tick"(available at https://x-datainitiative.github.io/tick/) to perform the analysis.

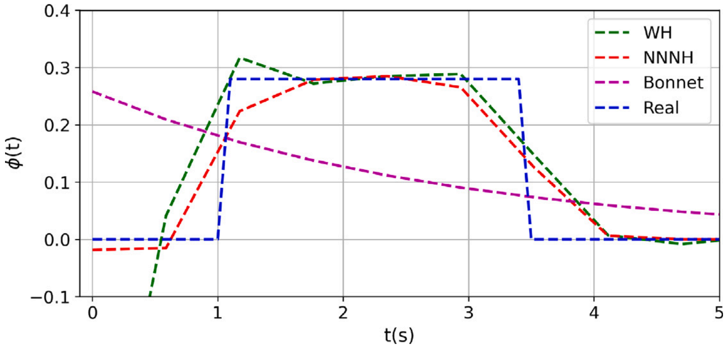


Fig. 5. Fitted Rectangular kernel: A comparison of the NNNH, WH, and Bonnet estimates with the actual (real) kernel.

Estimation of non-smooth kernels:. We first consider a linear Hawkes process with a non-smooth kernel, specifically a rectangular kernel of the form,

$$\text{Rectangular,}\ \ \phi(t) = \begin{cases} \alpha\beta, \text{if } \delta \le t \le \delta + \frac{1}{\beta}, \\ 0, \text{otherwise.} \end{cases}$$

The smoothness assumption of the kernel is a prerequisite for some non-parametric methods used in estimating the kernels of Hawkes processes, such as the Markovian Estimation of Mutually Interacting Process (MEMIP) proposed in Lemonnier and Vayatis [27]. As a result, it is necessary to compare the performance of the NNNH method using a non-smooth kernel in order to assess its effectiveness.

We simulate the process using the following parameter values for the rectangular kernel,

$$\mu = 0.05,\ \alpha = 0.7,\ \beta = 0.4,\ \delta = 1.$$

Simulating the process until $T = 60\,000$ yields $N(T) = 10\,001$ arrivals. As a first step, we visually compare the estimated kernels obtained using the NNNH and the benchmark methods with the true kernel. Fig. 5 illustrates the kernels fitted by the three estimation methods. As expected we see that the non-parametric models, i.e., the NNNH and the WH are able to better recover the ground truth. This is also reflected in the negative log-likelihood values obtained for WH, Bonnet, and NNNH methods, i.e., 4228, 12 838, and 2800 respectively.

Estimation of non-monotonic kernels:. As an example of a non-monotonic kernel,[4] we consider Erlang's kernel of the form,

$$\phi(t) = \alpha t^{-\beta t}.$$

We simulate the process using the following parameter values:

$$\mu = 0.2,\ \alpha = 0.3,\ \beta = 0.9.$$

We simulate the Hawkes process with the above kernel for $T = 10\,000$, resulting in a dataset of count $N(T) = 4859$ events. We compare the estimated NNNH kernel with the benchmark kernels. Fig. 6 NNNH illustrates the kernels fitted by the three estimation methods. As expected, the non-parametric models, i.e., the NNNH and the WH, can better recover the shape of the ground truth. The estimates from WH are noisier, which is also reflected in the corresponding values of the negative log-likelihood. The negative log-likelihood values obtained for WH, Bonnet, and NNNH methods are 7529, 6759, and 6736, respectively.

Estimation of negative kernels:. We next consider a non-linear Hawkes process,

$$\lambda^*(t) = \max\left(\mu + \sum_{\tau<t_n} \phi(t-\tau), 0\right),$$

where, Ψ in Eq. (3) is a max function, i.e. $\Psi(\lambda(t)) = \max(\lambda(t), 0)$. The inhibitive kernel, ϕ, is specified as an exponential function given by,

$$\phi(t) = -0.5e^{-2t}.$$

Given that non-parametric methods, such as the Expectation–Maximization (EM) approach introduced in Lewis and Mohler [15], are capable of estimating only positive kernels, it is of interest to evaluate the capacity of the NNNH method in estimating the negative kernels.

[4] We thank the referee for the suggestion

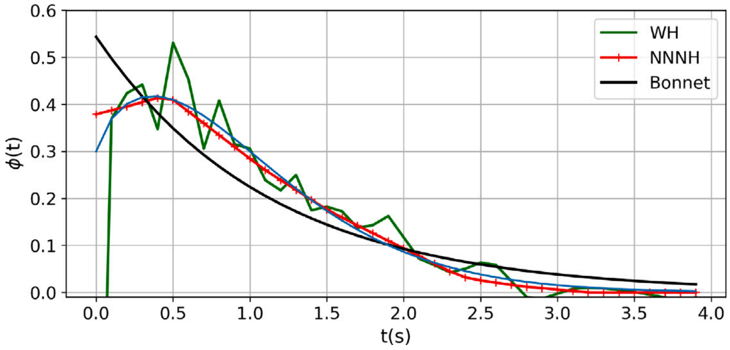


Fig. 6. Fitted Erlang kernel: A comparison of the NNNH, WH, and Bonnet estimates with the actual kernel.

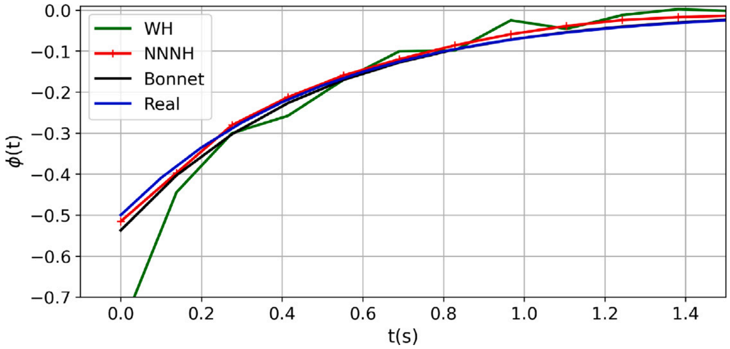


Fig. 7. Fitted negative exponential kernel: A comparison of the NNNH, WH, and Bonnet estimates with the actual kernel.

We simulate the above process for $T = 14\,000$, with a base intensity, $\mu = 0.9$. This resulted in $N(T) = 10\,001$ events. We first compare the kernel obtained using the NNNH, WH, and Bonnet method with the ground truth in Fig. 7. All three methods are able to recover the kernel, but visually the NNNH and Bonnet methods appear to provide better estimates. Furthermore, this conclusion is supported by the estimated negative log-likelihood values, which are 9833, 9559, and 9564 for the WH, Bonnet, and the NNNH, respectively.

Estimation of non-linear hawkes processes for different variations of Ψ :. Diverse variations of the non-linear Hawkes process can be derived by making distinct selections of the function $\Psi(.)$ in Eq. (3). In their work, Wang et al. [29] explore various variations of the non-linear dependency function, $\Psi(.)$ beyond the usual $\max$ function, including sigmoid and decreasing functions. To provide a more understanding of our proposed method, we conduct simulations of a non-linear Hawkes process using an alternative dependency function, specifically the sigmoid function defined as follows:

$$\Psi(x) = \frac{1}{1 + e^{-(x-2)}}.$$

For this particular simulation, we consider the following kernel:

$$\phi(t) = e^{-2t}.$$

This simulated data is utilize to estimate the Hawkes process using the NNNH model, there by demonstrating the effectiveness and versatility of our approach. Its important to note that while the dependency function, $\Psi(.)$ used for generating the Hawkes process is the sigmoid, but the underlying network architecture and the choice of activation function for estimation using NNNH remains the same as described in Section 3. The non-linear Hawkes process was simulated for a duration of $T = 10\,000$, resulting in $N(T) = 3028$ events. According to Eq. (9), the NNNH model converts a non-linear Hawkes process with any Ψ into a non-linear Hawkes process with Ψ as a $\max$ function. As a result, the recovered kernels from the NNNH method might not correspond to the kernels of the original process. However, we can compare the actual simulated intensity process with the intensities recovered from the NNNH method and the WH method. Fig. 8 demonstrates that the recovered intensities match the simulated intensities. The accuracy of NNNH estimates of quantiles is evident from the QQ(Quantile–Quantile) plot obtained from the fitted WH and the NNNH method on the test dataset, as shown in Fig. 9.

Predictive capacity of the NNNH method. Accurately inferring the kernels is critical in developing an improved predictive model for the arrival process. For example, a algorithm relies on predicting the time at which next arrival happens with 90% confidence. With

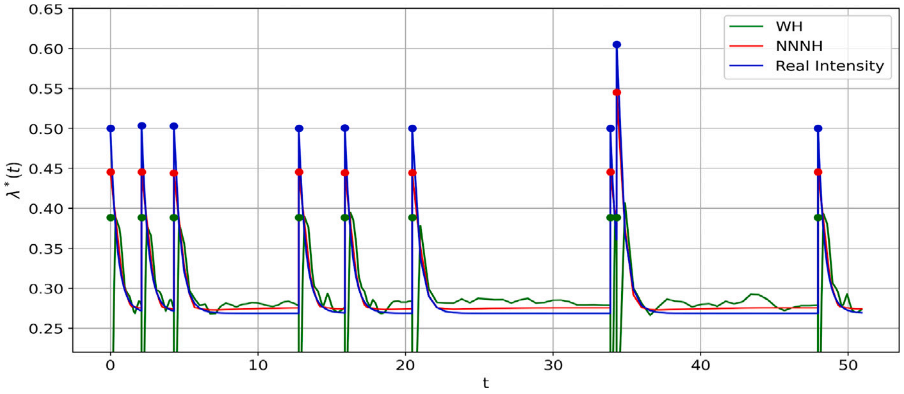


Fig. 8. Comparison of real and estimated intensity function by WH and NNNH for non-linear Hawkes process with sigmoid as ψ function.

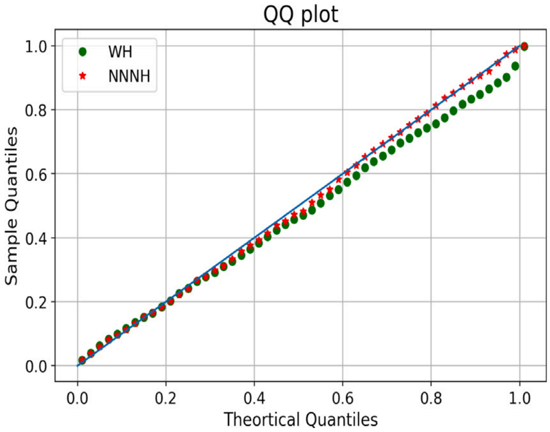


Fig. 9. QQ plot comparing estimated and true intensities for sigmoid non-linear Hawkes process using WH and NNNH.

an accurate prediction model, approximately 90% of the predictions are expected to be correct, and in roughly 10% of the cases, the events will arrive after the predicted time. If the model consistently produces accurate predictions (i.e., does not fail for 10% of the cases), then the predictions are overly conservative, indicating that the predicted time is set too far in the future. Conversely, if more than 10% of the predictions fail, the predicted time is closer than anticipated. Therefore, a prediction model that produces predictions with Q% certainty should have Q% correct outcomes. The accuracy of such a prediction model can be evaluated using a QQ plot.

In this context, we compare the NNNH method with the WH method. From the QQ plot presented in Fig. 9, NNNH model closely aligns with the 45-degree reference line than WH model, representing the ideal model's QQ plot. Furthermore, it is of significance to note that the negative log-likelihood values for NNNH and WH are 2954 and 3813 respectively. These observations lends support to our conclusion that NNNH offers a better fit compared to WH.

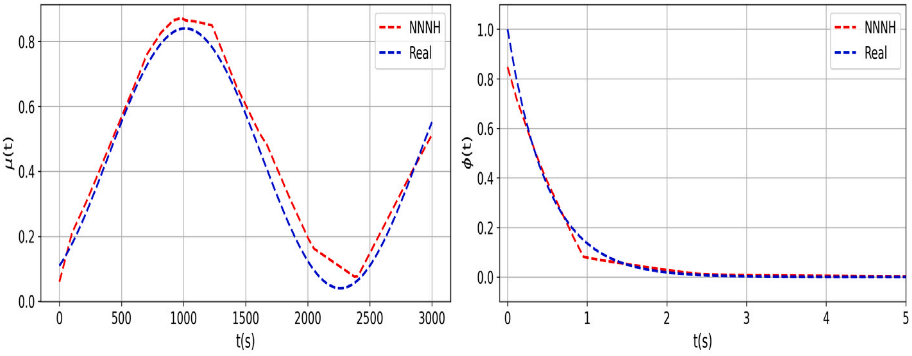
Estimation of Hawkes processes with varying base intensity function. To complete the univariate case test for the NNNH, we examine a Hawkes process with a base intensity that varies over time. We adopt a trigonometric sin function for the time-varying base intensity $\mu(t)$, which is specified in Table 1. The associated kernel for the Hawkes process is expressed as:

$$\phi(t) = e^{-2t}.$$

We simulate the process till $T = 3000$, which results in $N(T) = 2482$ events. The comparison between the recovered base intensity function and the excitation kernel using the NNNH method with the true values is depicted in Fig. 10. The results demonstrate that the NNNH approach can accurately estimate both the base intensity function and the kernel concurrently.

4.4. Multivariate Hawkes estimation

In this section, we evaluate the NNNH approach for estimating multivariate non-linear Hawkes processes using a simulated dataset. The multivariate non-linear Hawkes process is simulated using the *Ogata's thinning algorithm*. The parameters of the neural



(a) Base Intensity

(b) Kernel

Fig. 10. Estimated kernel and base intensity by applying NNNH to a one-dimensional Hawkes process with sin base intensity and an exponential kernel.

networks utilized to model the kernels in the NNNH method are optimized by maximizing the log-likelihood through the stochastic gradient descent method, as described in Section 3.1. We utilize the WH method (Bacry and Muzy [17]) and the Bonnet Multivariate method (Bonnet et al. [26]) as our comparative models. While the WH method is non-parametric, the Bonnet Multivariate approach assumes a parametric structure for the kernels in the Hawkes process. Both reference models are capable of estimating kernels that exhibit inhibitory effects.

We consider two examples for the simulated case. We first consider a multivariate Hawkes process, with both positive and negative exponential kernels. Specifically the parameter chosen for the model are:

$$\alpha = \begin{bmatrix} -0.9 & 3 \\ 1.2 & 1.5 \end{bmatrix},$$

$$\beta = \begin{bmatrix} 5 & 5 \\ 8 & 8 \end{bmatrix},$$

and a constant base intensity

$$\mu = \begin{bmatrix} 0.5 \\ 1.0 \end{bmatrix}.$$

The kernels of the process are defined as: $\phi_{dj}(t) = \alpha_{dj} e^{-\beta_{dj} t}$, $1 \leq d, j \leq 2$.

We simulate the process till $T = 1000$, which results in $N(T) = 1002$ events. Fig. 11 plots the true kernels and the kernels recovered using the NNNH, the WH, and the Bonnet Multivariate methods. Through a visual examination, it can be observed that the Bonnet Multivariate approach closely approximates the actual kernels. This outcome is in line with expectations as the simulated data employed a parametric form of the Hawkes process that is specifically suitable for the Bonnet Multivariate model. Regarding the non-parametric models, the WH estimates exhibit greater variability compared to the NNNH estimates. The respective negative log-likelihoods for the NNNH, the WH, and the Bonnet Multivariate methods are 1480, 1967, and 1460.

Subsequently, we examine a bivariate non-linear Hawkes process that encompasses a combination of diverse types of kernels. Precisely, we select the kernels as follows:

$$\text{exponential kernel}\, \phi_{11}(t) = 0.3e^{-3t},$$

$$\text{rectangular kernel}\, \phi_{12}(t) = \begin{cases} 0.7 \times 0.4, \text{if } 1 \leq t \leq 1 + \frac{1}{0.4}, \\ 0, \text{otherwise.} \end{cases},$$

$$\text{negative exponential}\, \phi_{21}(t) = -0.2e^{-t},$$

$$\text{exponential}\, \phi_{22}(t) = 0.4e^{-2t},$$

and use the following base intensity values:

$$\mu = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}.$$

We perform simulations up to time $T = 10\,000$, resulting in the occurrence of a total 8489 events in the two dimensions. Fig. 12 presents the findings obtained from the NNNH approach in comparison with other techniques. Notably, the advantages of employing a non-parametric estimation method for Hawkes processes are evident in this context. Both the NNNH and the WH methods successfully capture all the kernels, whereas the Bonnet Multivariate approach does not accurately depict the rectangular kernel. It is worth mentioning that the estimated kernels obtained using the WH method appear to be more erratic than those of the NNNH. The corresponding negative log-likelihood values for the WH, the Bonnet Multivariate, and the NNNH are 3051, 3062, and 2899 respectively.

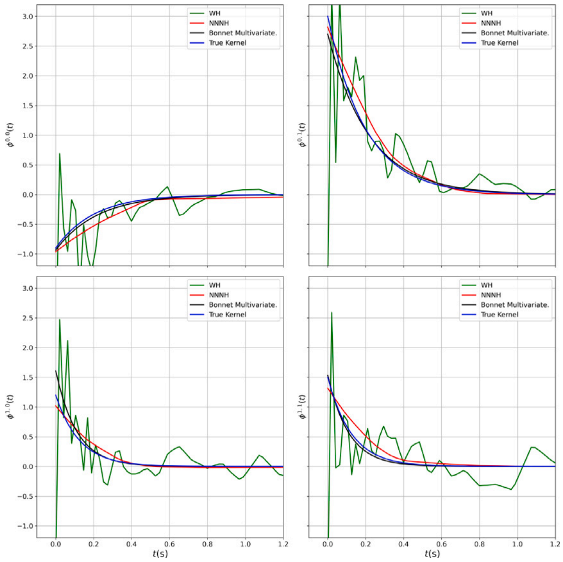


Fig. 11. The kernels fitted using the NNNH, WH, and Bonnet Multivariate methods, for a bivariate Hawkes process with both positive and negative kernels.

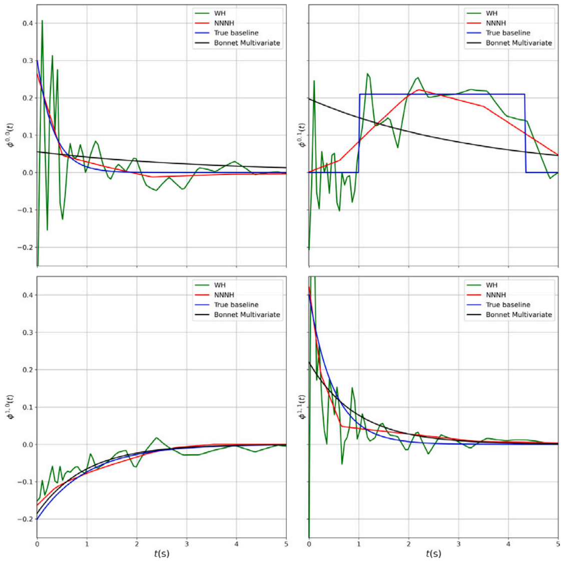


Fig. 12. The fitted kernels, using the NNNH, WH, and Bonnet Multivariate method, for a bivariate non-linear Hawkes process with a mix of exponential and rectangular kernels.

4.5. Sensitivity analysis

In the context of the NNNH approach, we have made deliberate decisions concerning the quantity of neurons deployed in the network's hidden layer to emulate the kernels and the learning rate employed to regulate the parameter updates in the Adam algorithm. To gauge the sensitivity of the NNNH method, we have conducted a numerical investigation with regard to these decisions. Specifically, we have examined the influence of variations in the number of neurons and learning rates on the negative log-likelihood value, while holding all other factors constant. Note that the likelihood values reported are for the validation dataset which was not used in the training of the network

Table 2
Percentage change in negative log-likelihood (NLL) values and corresponding computational time for different numbers of neurons in the neural network.

No of neurons	NLL	% change	% time change
2	2057.78	−0.061	+164.1
4	2058.01	−0.057	+95.4
8	2059.06	−0.004	+89.9
16	2059.14	−0.001	+90.9
32	2059.16	0.00	0.00
64	2061.29	+0.10	+6.7
128	2061.63	+0.12	+77.3
256	2061.62	+0.12	+76.1

Table 3
Percentage change in negative log-likelihood (NLL) values for different learning rates.

Learning rate	NLL	% change
0.01	2064.54	+0.00
0.005	2064.19	−0.02
0.001	2065.49	+0.05
0.0005	2067.87	+0.12

The sensitivity analysis is conducted for the negative exponential kernel, which is discussed in Section 4.3. The study focuses on the variation in the number of neurons employed in the neural networks, and the corresponding impact on the negative log-likelihood values. The results are presented in Table 2. The findings indicate that alterations in the number of neurons have an insignificant effect on the minimum negative log-likelihood values obtained, but they do impact the computational time required to achieve optimal parameters. Table 2 demonstrates that using a smaller number of neurons results in increased computational time. This can be attributed to the larger number of iterations required for the method to reach the stopping criteria. In other words, as the number of neurons decreases, the optimization process takes longer to converge, leading to higher computational time. The neural network with thirty two neurons has the lowest computational time. Therefore, we use a neural network with 32 neurons in our experiments to model the kernels.

Table 3 presents the percentage variation in the negative log-likelihood values linked to the optimal parameters obtained by varying the learning rate for the output layer of the neural network. The learning rate for the hidden layer remains unchanged, and is taken as ten percent of the learning rate for the output layer.

We find that the optimal network parameters obtained using varying learning rates results in similar negative log-likelihood values and therefore use a learning rate of 0.01 for all our experiments.

4.6. Real data

4.6.1. Financial dataset

In this study, we evaluate the efficacy of the NNNH method on high-frequency order book data pertaining to two of the most frequently traded cryptocurrencies, namely bitcoin and Ethereum. The data comprises of buy and sell trade records for the BTC-USD (bitcoin-US dollar) and ETH-USD (Ethereum-US Dollar) pairs. We streamed the Binance exchange order book data, as several popular cryptocurrencies are traded in this exchange, and the exchange has high trade volumes. We obtain the tick-by-tick arrival times for both buy and sell trades from the exchange. The market is composed of makers and takers, with makers generating buy or sell orders that are not carried out immediately, thereby creating liquidity for that cryptocurrency. In contrast, takers place market orders that are executed instantly, thereby taking away the liquidity.

The Binance exchange furnishes two streams of tick-by-tick data: trade arrival data and trade stream data. The trade arrival data comprises of limit orders, which are orders placed with a specified price limit, and market orders, which are orders executed at the prevailing market price. Limit orders are executed when the best available market price reaches the set limit, while market orders are executed instantly at the current best limit order. The Binance trade-stream data provides the timestamp of these order arrivals, along with price and volume features, and a unique identifier for the buyer/seller. Since a single market order may necessitate multiple limit orders to fulfil the requested volume, several trades are recorded with a common identifier. Therefore, we cleaned the dataset by filtering out the data with common IDs and retaining only the unique trade events. Finally, based on whether the buyer was a market maker or taker, the trades were marked as either a buy or a sell market order.

Our analysis focuses exclusively on the trades conducted for the BTC-USD and ETH-USD pairs, which account for the predominant trading volumes in the cryptocurrency exchange. The arrival time for market orders for these two pairs can be modelled as a four dimensional non-linear Hawkes process, i.e.,

First Dimension: Intensity process for the sell market orders for the BTC-USD pair
Second Dimension: Intensity process for the buy market orders for the BTC-USD pair

Table 4
Summary of the crypto-currency market orders recorded on December 7 2021 between 12:00 and 12:10 UTC.

Market order type	BTC-USD	ETH-USD	Aggregate
Sell	4219	3480	7699
Buy	4374	2999	7373
Total	8593	6479	15 072

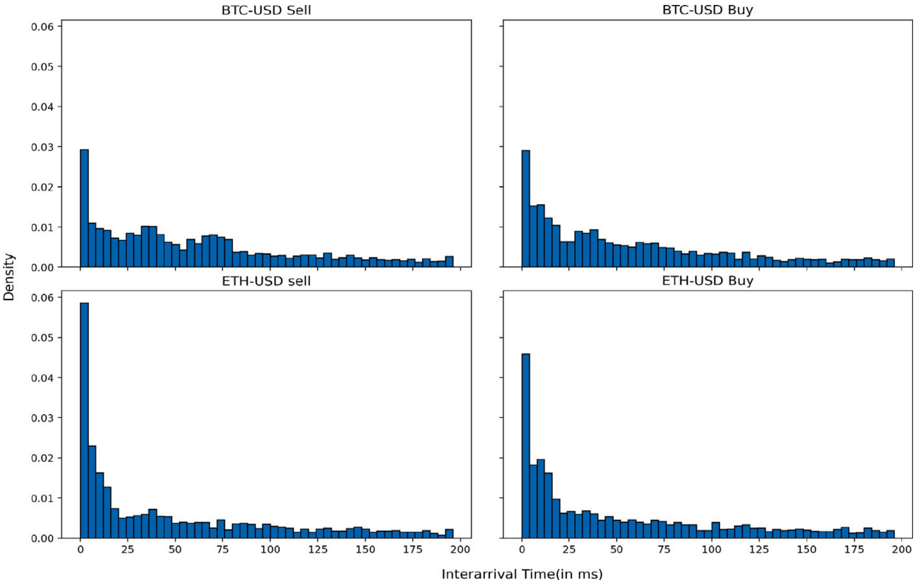


Fig. 13. Histogram for the inter event time arrivals of the BTC-USD sell trade.

Third Dimension: Intensity process for the sell market orders for the ETH-USD pair
Fourth Dimension: Intensity process for the buy market orders for the ETH-USD pair.

The analysis was performed for the market orders arrival times on December 7 2021, between 12:00 to 12:10 (UTC) on the Binance exchange. Table 4 summarizes the data after classifying the trades as buy or sell market order events.

Fig. 13 provides the histogram of the inter-arrival time of the sell BTC-USD market orders. The timestamps for order arrival are in milliseconds.

Our aim is to employ the NNNH method to jointly model the intensity process of buy and sell market orders for BTC-USD and ETH-USD currency pairs. This modelling approach enables us to gain insights into the causal relationships between the two pairs, such as whether the arrival of a buy BTC-USD order affects the arrival of a sell ETH-USD market order. Furthermore, identifying the functional form of the self and cross modulation due to the arrival of market orders is of interest. This highlights the necessity of non-parametric estimation techniques, as the true form of the modulation function remains unknown. To achieve our objective, we partition the dataset into train, validation, and test sets. We optimize the parameters of the model using stochastic gradients computed on the train set and use the validation set for applying the early stopping criteria. Specifically, if there is no improvement in negative log-likelihood values computed on the validation set for ten consecutive iterations, we stop training. Finally, we evaluate the goodness of fit using the negative log-likelihood reported on the test set.

We adopt the widely used non-parametric methods, namely the WH and the EM method (Lewis and Mohler [15]), as reference models. In Fig. 14, we compare the obtained kernels using the three methods. The corresponding negative log-likelihood values are 9630, 6732, and 6400 for the EM, WH, and NNNH methods, respectively. The estimated base intensities for BTC-USD sell, BTC-USD buy, ETH-USD sell, and ETH-USD buy, in the order of appearance, using the NNNH method are 0.0028, 0.0032, 0.0021, and 0.0022, respectively.

By examining Fig. 14, we can draw the following observations for the market orders that arrived during the training window:

- Significant self excitation is observed for both BTC-USD and ETH-USD buy and sell market orders.
- ETH-USD buy orders result in excitation of BTC-USD buy and BTC-USD sell orders.
- The self excitation of ETH-USD buy and sell orders is higher compared to their respective BTC-USD orders.
- A certain level of inhibition is observed in BTC-USD sell orders due to buy BTC-USD orders.

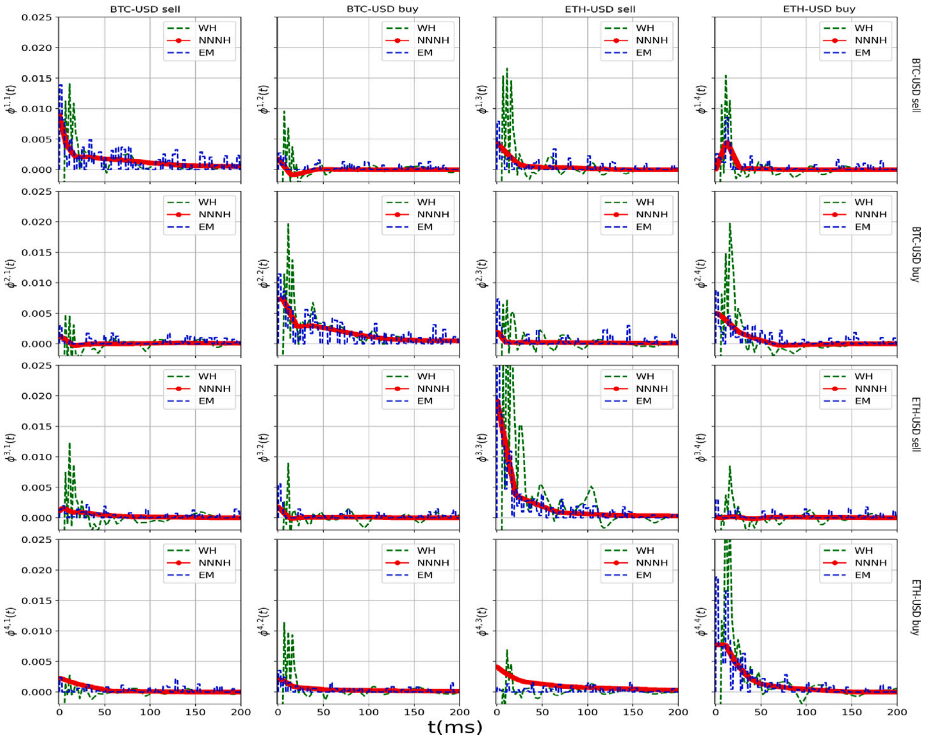


Fig. 14. Fitted kernels for BTC-USD and ETH-USD sell-buy market orders estimated using the NNNH, the WH, and the EM methods.

Notably, all three methods effectively capture a lagged cross-excitation effect in BTC-USD sell orders caused by buy ETH-USD buy orders.

Fig. 15 presents the QQ plot for the BTC-USD and ETH-USD order arrivals based on the EM, WH, and NNNH models. The sample quantiles for the QQ plot are obtained from the test dataset. The results indicate that all three models make reasonably accurate predictions. Notably, the NNNH method appears to have more precise predictions, particularly for BTC-USD and ETH-USD sell orders, as evidenced by the visual assessment.

4.6.2. *Neuron spike train dataset*

In this study, we utilized the NNNH method to analyse a dataset of neuron spikes, which was obtained from an experiment conducted by Engelhard et al. [48]. The primary objective of the experiment was to examine the relationship between unit recordings from the motor cortex of monkeys and the position and grip strength of their hands as they utilized a joystick to manipulate a robotic arm. Consistent with the approach taken by Aljadeff et al. [49], we analysed the data obtained from the nine simultaneous electrode recordings. Although the dataset included additional information regarding hand motion, such as cursor trajectory and grip force, our analysis was focused exclusively on the neuron spike train.

To provide a brief biological background about neuron spikes, we refer to Duval et al. [50]. Neurons employ an electrical wave of short duration, known as the action potential or spike, to reliably transmit signals from one end to the other (Castelfranco and Hartline [51]). Action potentials are all-or-none events and are triggered when the neuron's membrane potential, which is the electrical potential difference between the inside and the outside of the neuron, is sufficiently large. Between two successive action potentials, the neuron adds up its inputs, causing changes in its membrane potential. When this potential reaches a sharp threshold, the neuron fires a spike that propagates (Luo et al. [52]).

The histogram in Fig. 16 presents the inter-arrival times of spikes generated by a single neuron. Notably, the peak of the histogram is marginally displaced from the origin, in contrast to the histogram of a conventional exponential distribution. This observation may suggests the presence of an inhibitory process. To model the neuron spike train, we employed a non-linear Hawkes process that is nine-dimensional. Each dimension of the process is associated with the intensity of neuron spike arrivals at an individual electrode. We use the NNNH method to estimate the kernel functions and the base intensities for this process.

Table 5 gives the number of spike events for each neurons used for the analysis.

Table 6 reports the base intensity values for the arrival of neuron spikes at each electrode obtained from applying NNNH to the neuron spike events. Furthermore, Fig. 17 visually presents a network diagram depicting the outcomes derived from the NNNH estimation. This figure offers insights into both self-dependencies and mutual interactions among individual neurons. Additionally,

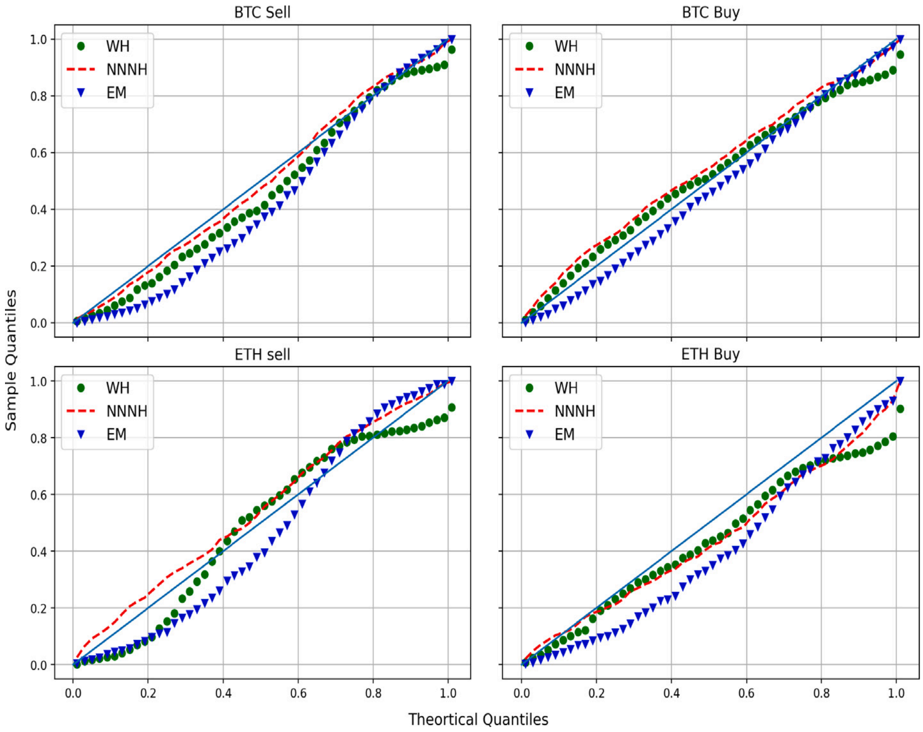


Fig. 15. QQ plot comparing order arrivals of BTC-USD and ETH-USD using EM, WH, and NNNH methods.

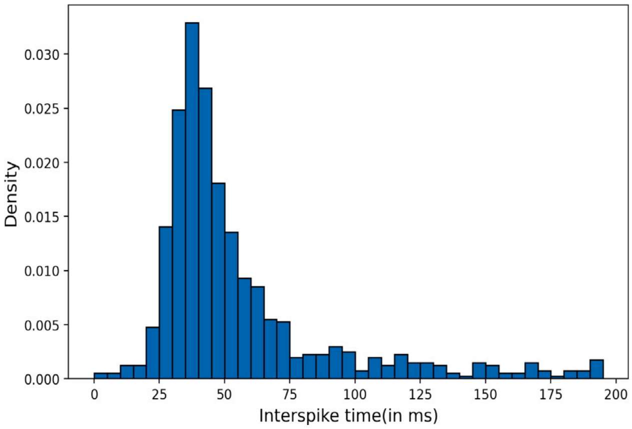


Fig. 16. Histogram of inter-spike time intervals for neuron 1 in grip-and-reach task.

Table 5
Summary statistics of neuron spike data.

Neuron	1	2	3	4	5	6	7	8	9
No. of events	4121	1895	912	2509	221	2784	149	3854	1360

the figure includes a depiction of the kernel norm, denoted as $|\phi_{dj}|_1$, associated with each individual kernel. The kernel norm, defined as $|\phi_{dj}|_1 = \int_0^\infty \phi_{dj} dt$, denotes the average count of events attributed to type d that are triggered or decayed due to the influence of type j (Bacry et al. [53]). Also Fig. C.18 in Appendix C illustrates the interdependency among neurons derived from the NNNH estimation. The observations drawn from the Fig. C.18 and network diagram in Fig. 17 suggest that the neurons not

Table 6
Base intensities of neuron spike data obtained using the NNNH for the grip-and-reach task.

μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9
0.009	0.035	0.001	0.046	0.001	0.043	0.001	0.076	0.022

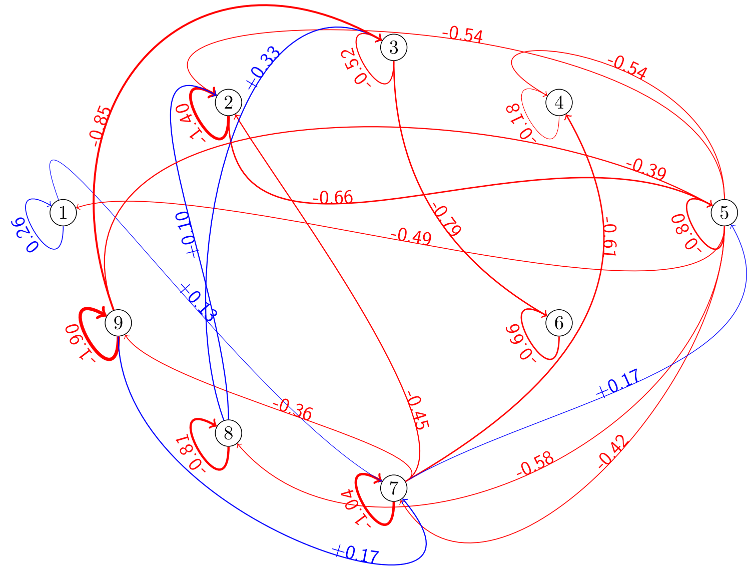


Fig. 17. A network representation of the inter-neuronal dependency generated through the NNNH method in a monkey grip-and-reach task is presented. Each node within the graph corresponds to an individual neuron, while the edges signify the directional relationships between neurons, including self-dependency. Notably, the numerical values associated with the edges denote the corresponding kernel norm values. Excitatory connections are denoted by a blue colour, while inhibitory connections are highlighted in red.

only demonstrate self-dependency but also establish inter-dependency connections with each other. These self or mutual neuronal dependencies manifest with varying degrees across different neurons. While the majority of neurons exhibit a decaying behaviour in their dependencies, a subset of neurons showcases an excitatory relationship with others. This phenomenon indicates that a spike occurring in a single neuron can trigger either an excitatory or inhibitory response in neighbouring neurons.

From Fig. C.18, it is observed that a majority of the neurons exhibit a self-inhibitory characteristic. Specifically, after firing, neurons typically undergo a refractory period during which they cannot generate additional spikes. The inhibitory pattern evident in the plots likely corresponds to this refractory period. Based on the figures, it is reasonable to assume that the refractory period for the recorded motor neurons falls within the 20–60 ms range.

5. Conclusion

This paper proposes a non-parametric method for estimating non-linear Hawkes processes. We call the method Neural Network for Non-linear Hawkes processes or the NNNH method. The NNNH method involves modelling the kernels and the base intensity functions of a non-linear Hawkes process as individual neural networks. The parameters of the neural networks involved are jointly estimated by maximizing the log-likelihood function using the batch stochastic gradient descent with Adam for adaptive learning. To apply the SGD method, an unbiased estimator for the gradients with respect to the network parameters is proposed. The paper provides an efficient scheme to evaluate the integrated intensity and its gradients, which can be a computational bottleneck.

The effectiveness of the NNNH method in learning the underlying process is investigated through numerical experiments. Unlike many recent neural network-based models that predict conditional intensities, the NNNH model retains the interpretability of the Hawkes process by enabling the inference of the kernels rather than just direct prediction of the conditional intensities. The ability to recover the kernels is desirable for understanding causal relationships between the arrivals in different dimensions, for instance. The NNNH method performed satisfactorily for the diverse set of problems considered. We also see that the NNNH can be used to estimate non-homogeneous Poisson process. The NNNH method is able to recover non-smooth kernels, kernels with negative intensities, and estimate other variants of the non-linear Hawkes processes.

The NNNH approach is applied to examine the process of order arrivals for buy and sell transactions on the Binance exchange in relation to the BTC-USD and ETH-USD currency pairs. We find evidence of both self and cross excitation in the order arrivals for the two currency pairs. Notably, self excitation is observed to be significant for both currency pairs, while the cross dependence exhibits an asymmetry whereby ETH-USD order arrivals are more strongly influenced by BTC-USD order arrivals than vice versa.

We applied the NNNH model to a publicly accessible dataset of neuron spikes obtained from the motor cortex of monkeys engaged in a grip-and-reach task. As anticipated, the kernels obtained from the analysis exhibit self-inhibition, which could be associated with the refractory period of the neurons. Our findings provide evidence of the NNNH method's efficacy in analysing high-dimensional datasets.

The NNNH method has a disadvantage in that the estimation of model parameters relies on the use of stochastic gradient descent (SGD), which updates the network parameters iteratively using first-order derivatives. Convergence of SGD-based methods can be slow, often requiring several iterations before the stopping criterion is met. However, the use of SGD makes the NNNH method well-suited for online learning, enabling the model to be trained on new data points that are continuously streaming in, as opposed to a static dataset. Moreover, the NNNH method is amenable to parallel computing of the gradients for batch SGD. A potential avenue for future research is to extend the NNNH method to model marked non-linear Hawkes processes.

Data availability

I have shared the link of the data and the code on the manuscript.

Appendix A. Gradients of the neural network parameters

As both $\mu_d(x)$ and $\phi_{dj}(x)$ has similar network, the gradients will also be similar. The gradient with respect to each of the parameterθ_p is given by,

$$\nabla_{\theta_p}\widehat{\phi}_{dj}(x) = \nabla_{\left[b_2, a_2^p, a_1^p, b_1^p\right]}\left(b_2 + \sum_{i=1}^{P} a_2^i \max\left(a_1^i x + b_1^i, 0\right)\right)$$

The gradients corresponding to each of the neural parameters were provided as,

$$\nabla_{b_2}\phi_{dj}(x) = 1$$

$$\nabla_{a_2^p}\phi_{dj}(x) = \max\left(a_1^p x + b_1^p, 0\right)$$

$$\nabla_{a_1^p}\phi_{dj}(x) = \left(a_2^p x\right) \mathbb{1}_{\left(a_1^p x + b_1^p, 0\right) > 0}$$

$$\nabla_{b_1^p}\phi_{dj}(x) = \left(a_2^p\right) \mathbb{1}_{\left(a_1^p x + b_1^p, 0\right) > 0}$$

Appendix B. Integrated intensity function — network representation

Given the splitting criteria in Section 3.2,

$$\int_{s_{q-1}}^{s_q} \widehat{\lambda}_d^*(s)ds = \int_{s_{q-1}}^{s_q} \max\left(\widehat{\lambda}_d(s), 0\right) ds,$$

If there are no zero crossings for $\max\left(\widehat{\lambda}_d(s), 0\right)$ within the interval $[s_{q-1}, s_q]$, then $\widehat{\lambda}_d^*(s)$ is linear in the interval. Also if $\widehat{\lambda}_d(s_{q-1}) > 0$, then the network expression for the integrated intensity is given as,

$$\begin{aligned}
\int_{s_{q-1}}^{s_q} \max\left(\widehat{\lambda}_d(s), 0\right) ds &= \left[\int_{s_{q-1}}^{s_q} \widehat{\lambda}_d(s)ds\right] \mathbb{1}_{\widehat{\lambda}_d(s_{q-1})>0} \\
&= \left[\int_{s_{q-1}}^{s_q} \widehat{\mu} + \sum_{\{k,i,j\}} a_2^i \max\left(a_1^i(s - t_k^j) + b_1^i, 0\right)\right] \mathbb{1}_{\widehat{\lambda}_d(s_{q-1})>0} \\
&= \left[\int_{s_{q-1}}^{s_q} \widehat{\mu} + \sum_{\{k,i,j\}} a_2^i \left[a_1^i(s - t_k^j) + b_1^i\right] \mathbb{1}_{a_1^i(s-t_k^j)+b_1^i>0}\right] \mathbb{1}_{\widehat{\lambda}_d(s_{q-1})>0} \\
&= \left[\widehat{\mu}s + \sum_{\{k,i,j\}} a_2^i \left[a_1^i\left(\frac{s^2}{2} - t_k^j s\right) + b_1^i s\right] \mathbb{1}_{a_1^i(s-t_k^j)+b_1^i>0}\right]_{s_{q-1}}^{s_q} \mathbb{1}_{\widehat{\lambda}_d(s_{q-1})>0}
\end{aligned}$$

Appendix C. Neuron spike train dataset — plots

See Fig. C.18.

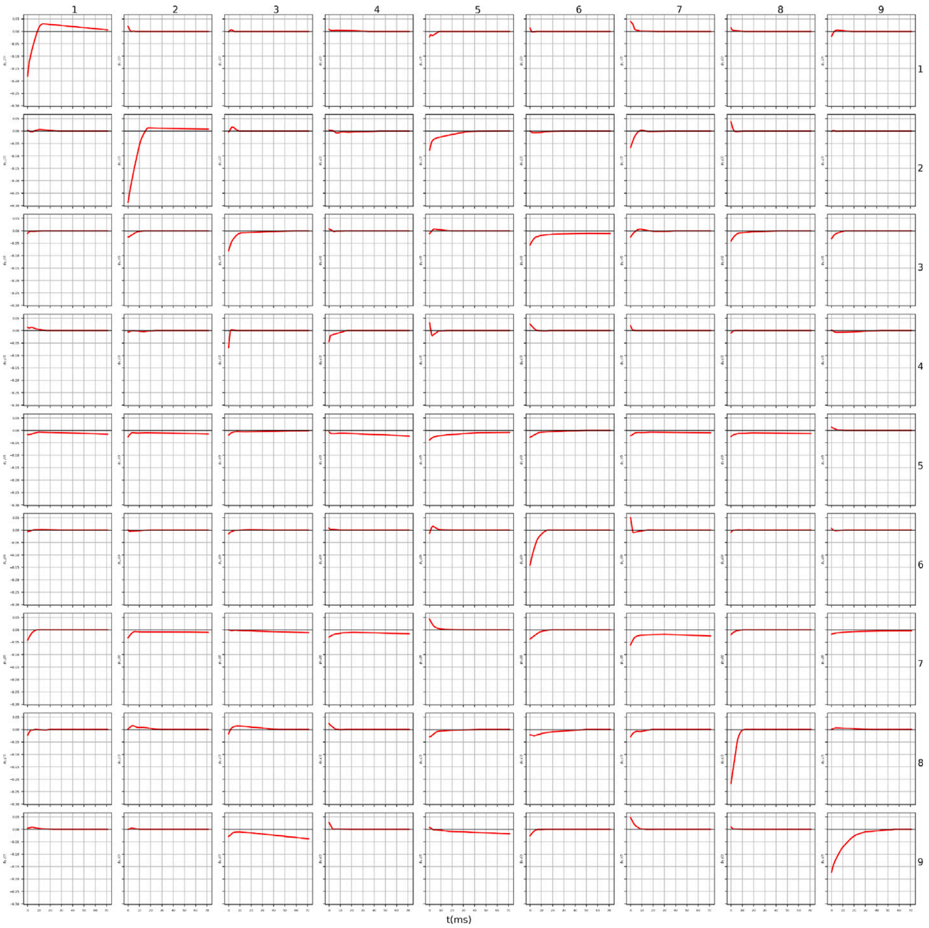


Fig. C.18. Kernels of motor neurons demonstrating the self-dependency and mutual dependency of neurons for the grip-and-reach task estimated by NNNH.

References

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

[2] Y. Ogata, Seismicity analysis through point-process modeling: A review, in: Seismicity Patterns, their Statistical Significance and Physical Meaning, Springer, Basel, 1999, pp. 471–507.

[3] D. Marsan, O. Lengline, Extending earthquakes' reach through cascading, Science 319 (5866) (2008) 1076–1079.

[4] V. Filimonov, D. Sornette, Quantifying reflexivity in financial markets: Toward a prediction of flash crashes, Phys. Rev. E 85 (5) (2012) 056108.

[5] E. Bacry, I. Mastromatteo, J.-F. Muzy, Hawkes processes in finance, Mark. Microstruct. Liquid. 1 (01) (2015) 1550005.

[6] R. Crane, D. Sornette, Robust dynamic classes revealed by measuring the response function of a social system, Proc. Natl. Acad. Sci. 105 (41) (2008) 15649–15653.

[7] C. Blundell, J. Beck, K.A. Heller, Modelling reciprocating relationships with hawkes processes, in: Advances in Neural Information Processing Systems, 2012, pp. 2600–2608.

[8] K. Zhou, H. Zha, L. Song, Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes, in: Artificial Intelligence and Statistics, 2013, pp. 641–649.

[9] G.O. Mohler, M.B. Short, P.J. Brantingham, F.P. Schoenberg, G.E. Tita, Self-exciting point process modeling of crime, J. Amer. Statist. Assoc. 106 (493) (2011) 100–108.

[10] P. Reynaud-Bouret, S. Schbath, et al., Adaptive estimation for hawkes processes; application to genome analysis, Ann. Statist. 38 (5) (2010) 2781–2822.

[11] L. Carstensen, A. Sandelin, O. Winther, N.R. Hansen, Multivariate hawkes process models of the occurrence of regulatory elements, BMC Bioinformat. 11 (1) (2010) 1–19.

[12] J. Park, A.W. Chaffee, R.J. Harrigan, F.P. Schoenberg, A non-parametric hawkes model of the spread of ebola in west africa, J. Appl. Stat. (2020) 1–17.

[13] W.-H. Chiang, X. Liu, G. Mohler, Hawkes process modeling of COVID-19 with mobility leading indicators and spatial covariates, Int. J. Forecast. 38 (2) (2022) 505–520.

[14] T. Ozaki, Maximum likelihood estimation of hawkes' self-exciting point processes, Ann. Inst. Statist. Math. 31 (1) (1979) 145–155.

[15] E. Lewis, G. Mohler, A nonparametric EM algorithm for multiscale hawkes processes, J. Nonparametr. Stat. 1 (1) (2011) 1–20.

[16] K. Zhou, H. Zha, L. Song, Learning triggering kernels for multi-dimensional hawkes processes, in: International Conference on Machine Learning, PMLR, 2013, pp. 1301–1309.

[17] E. Bacry, J.-F. Muzy, Second order statistics characterization of hawkes processes and non-parametric estimation, 2014, arXiv preprint arXiv:1401.0903.

[18] M. Achab, E. Bacry, S. Gaïffas, I. Mastromatteo, J.-F. Muzy, Uncovering causality from multivariate hawkes integrated cumulants, J. Mach. Learn. Res. 18 (1) (2017) 6998–7025.

[19] Y. Yang, J. Etesami, N. He, N. Kiyavash, Online learning for multivariate hawkes processes, in: Advances in Neural Information Processing Systems, 2017, pp. 4937–4946.

[20] H. Xu, M. Farajtabar, H. Zha, Learning granger causality for hawkes processes, in: International Conference on Machine Learning, 2016, pp. 1717–1726.

[21] S. Joseph, L.D. Kashyap, S. Jain, Shallow neural hawkes: Non-parametric kernel estimation for hawkes processes, J. Comput. Sci. (2022) 101754.

[22] P. Reynaud-Bouret, V. Rivoirard, C. Tuleau-Malot, Inference of functional connectivity in neurosciences via hawkes processes, in: 2013 IEEE Global Conference on Signal and Information Processing, IEEE, 2013, pp. 317–320.

[23] P. Brémaud, L. Massoulié, Stability of nonlinear hawkes processes, Ann. Probab. (1996) 1563–1588.

[24] P. Reynaud-Bouret, V. Rivoirard, F. Grammont, C. Tuleau-Malot, Goodness-of-fit tests and nonparametric adaptive estimation for spike train analysis, J. Math. Neurosci. 4 (2014) 1–41.

[25] A. Bonnet, M. Herrera, M. Sangnier, Maximum likelihood estimation for hawkes processes with self-excitation or inhibition, 2021, arXiv preprint arXiv:2103.05299.

[26] A. Bonnet, M.M. Herrera, M. Sangnier, Inference of multivariate exponential hawkes processes with inhibition and application to neuronal activity, 2022, arXiv preprint arXiv:2205.04107.

[27] R. Lemonnier, N. Vayatis, Nonparametric Markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate hawkes processes, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2014, pp. 161–176.

[28] R. Lemonnier, K. Scaman, A. Kalogeratos, Multivariate hawkes processes for large-scale inference, 2016, arXiv preprint arXiv:1602.08418.

[29] Y. Wang, B. Xie, N. Du, L. Song, Isotonic hawkes processes, in: International Conference on Machine Learning, PMLR, 2016, pp. 2226–2234.

[30] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, L. Song, Recurrent marked temporal point processes: Embedding event history to vector, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1555–1564.

[31] H. Mei, J.M. Eisner, The neural hawkes process: A neurally self-modulating multivariate point process, Adv. Neural Inf. Process. Syst. 30 (2017).

[32] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[33] M. Costa, C. Graham, L. Marsalle, V.C. Tran, Renewal in hawkes processes with self-excitation and inhibition, Adv. Appl. Probab. 52 (3) (2020) 879–915.

[34] Á. Cartea, S.N. Cohen, S. Labyad, Gradient-based estimation of linear hawkes processes with general kernels, 2021, arXiv preprint arXiv:2111.10637.

[35] Y. Lee, K.W. Lim, C.S. Ong, Hawkes processes with stochastic excitations, in: International Conference on Machine Learning, PMLR, 2016, pp. 79–88.

[36] F. Zhou, S. Luo, Z. Li, X. Fan, Y. Wang, A. Sowmya, F. Chen, Efficient em-variational inference for nonparametric hawkes process, Stat. Comput. 31 (4) (2021) 46.

[37] I. Rubin, Regular point processes and their detection, IEEE Trans. Inform. Theory 18 (5) (1972) 547–557.

[38] D.J. Daley, D. Vere-Jones, An introduction to the theory of point processes: Volume I: Elementary Theory and Methods, Springer Science & Business Media, Springer, 2007.

[39] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.

[40] M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Netw. 6 (6) (1993) 861–867.

[41] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, MIT, 2016.

[42] S.E. Rigdon, A.P. Basu, The power law process: a model for the reliability of repairable systems, J. Qual. Technol. 21 (4) (1989) 251–260.

[43] S. Lee, J.R. Wilson, M.M. Crawford, Modeling and simulation of a nonhomogeneous Poisson process having cyclic behavior, Comm. Statist. Simulation Comput. 20 (2–3) (1991) 777–809.

[44] L.M. Leemis, Nonparametric estimation of the cumulative intensity function for a nonhomogeneous Poisson process, Manage. Sci. 37 (7) (1991) 886–900.

[45] X. Xiao, T. Dohi, Wavelet shrinkage estimation for non-homogeneous Poisson process based software reliability models, IEEE Trans. Reliab. 62 (1) (2013) 211–225.

[46] P.W. Lewis, G.S. Shedler, Simulation of nonhomogeneous Poisson processes by thinning, Naval Res. Logist. Q. 26 (3) (1979) 403–413.

[47] Y. Ogata, On lewis' simulation method for point processes, IEEE Trans. Inform. Theory 27 (1) (1981) 23–31.

[48] B. Engelhard, N. Ozeri, Z. Israel, H. Bergman, E. Vaadia, Inducing gamma oscillations and precise spike synchrony by operant conditioning via brain-machine interface, Neuron 77 (2) (2013) 361–375.

[49] J. Aljadeff, B.J. Lansdell, A.L. Fairhall, D. Kleinfeld, Analysis of neuronal spike trains, deconstructed, Neuron 91 (2) (2016) 221–259.

[50] C. Duval, E. Luçon, C. Pouzat, Interacting hawkes processes with multiplicative inhibition, Stochastic Process. Appl. 148 (2022) 180–226, http://dx.doi.org/10.1016/j.spa.2022.02.008.

[51] A.M. Castelfranco, D.K. Hartline, Evolution of rapid nerve conduction, Brain Res. 1641 (2016) 11–33.

[52] C. Luo, X. Zheng, D. Zeng, Inferring social influence and meme interaction with hawkes processes, in: 2015 IEEE International Conference on Intelligence and Security Informatics, ISI, IEEE, 2015, pp. 135–137.

[53] E. Bacry, T. Jaisson, J.-F. Muzy, Estimation of slowly decreasing hawkes kernels: application to high-frequency order book dynamics, Quant. Finance 16 (8) (2016) 1179–1201.