Journal of Computational Science 63 (2022) 101754

Contents lists available at ScienceDirect

Journal of Computational Science

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

Shallow Neural Hawkes: Non-parametric kernel estimation for Hawkes processes

Sobin Joseph[1], Lekhapriya Dheeraj Kashyap[1], Shashi Jain *,[1]

Department of Management Studies, Indian Institute of Science, Bangalore, India

ARTICLE INFO

Keywords:
Multidimensional hawkes process
Non-parametric estimation
Log-likelihood estimator
Shallow neural hawkes
BTC-USD market microstructure

ABSTRACT

The Multi-dimensional Hawkes Process (MHP) is a class of self and mutually exciting point processes that find many applications–from predicting earthquakes to modelling order books in high-frequency trading. This paper makes two significant contributions; we first find an unbiased estimator for the gradient of the Hawkes process's log-likelihood estimator. The estimator enables the efficient implementation of the stochastic gradient descent method for the maximum likelihood estimation. The second contribution is that we propose a specific neural network for the non-parametric estimation of the underlying kernels of the MHP. We evaluate the proposed model on synthetic and natural datasets and find the method has comparable or better performance than existing estimation methods. The use of neural networks for modelling the excitation kernel ensures that we do not compromise on the Hawkes model's interpretability. At the same time, the proposed algorithm has the flexibility to estimate any non-standard Hawkes excitation kernel.

1. Introduction

Development in technology has revolutionised the course and mechanism of financial trading. Analysis of market variations at a microstructure level facilitates constructive interpretation of events, thereby making valuable decisions and recommendations in buying or selling stocks. Combining historical data, one can seek correlations between asset interactions to analyse how market processes affect its variables and potentially project the asset performance into the future. An extensive body of empirical literature employs point processes to describe high-frequency data and trade arrival dynamics. In this paper, we are concerned with discovering correlations among event streams that cause or excite future events. A specialised case of the self-exciting point process, called the Hawkes process, is thus a natural mathematical model to govern the trade arrival activity.

Hawkes processes (Hawkes [1]) are temporal point processes in which the intensity depends on the process history with an excitation mechanism. It is well known for studying seismic events (Ogata [2]), financial analysis (Filimonov and Sornette [3], and Bacry et al. [4]) and modelling social interactions (Crane and Sornette [5], Blundell et al. [6], and Zhou et al. [7]). Hawkes process is employed as an intensity-based model for car accident losses and thereby computes automobile insurance premia in Errais [8]. This self-affecting point process captures the feedback effects of events and is computationally tractable. In the field of biology, it is used to study genomic events along with DNA sequences (Reynaud-Bouret et al. [9]). The MHP has also been used to model crime (Mohler et al. [10]) and studies the pattern of civilian deaths in Iraq (Lewis et al. [11]). The primary concern in modelling the Hawkes process is estimating the link function or the excitation kernel. A common practice has been to assume a parametric form of the excitation kernel, the most common being exponential and power-law decay kernels, and then use maximum likelihood estimation (Ozaki [12]) to determine the optimal values of the parameters.

Formally, the multi-dimensional Hawkes process is defined by a D-dimensional point process N_t^d , $d = 1, \dots, D$, with the conditional intensity for the dth dimension expressed as,

$$\lambda_d(t) = \mu_d + \sum_{j=1}^{D} \int_0^t \phi_{dj}(t-\tau) d N_\tau^j, \tag{1}$$

where μ_d is the exogenous base intensity for the dth node and is independent of the history. ϕ_{dj}, $1 \leq d, j \leq D$ are called the excitation kernels that quantify the magnitude of excitation of the base intensity μ_d of the dth node over time due to the past events from node j. These kernel functions are positive and causal (their support is within $\mathbb{R}^+$). Inferring a Hawkes process requires estimating the base intensity μ_d and its kernels functions ϕ_{dj}, either by assuming a parametric form for the kernels or in a non-parametric fashion. Recent developments focus on data-driven, non-parametric estimations of MHP to capture the general shape of the kernel and increase the flexibility of the model.

* Corresponding author.
E-mail address: shashijain@iisc.ac.in (S. Jain).

[1] All three authors have made an equal contribution to the reported study.

https://doi.org/10.1016/j.jocs.2022.101754
Received 27 December 2021; Received in revised form 1 June 2022; Accepted 27 June 2022
Available online 4 July 2022

In general, the kernels ϕ_{dj} and the base intensity μ_d can be estimated by maximising the associated log-likelihood function. However, as will be discussed further in Section 3, the challenge is that the log-likelihood function contains the integral of the intensity function, $\lambda_d(t)$, that depends on the values of the kernels over the whole time interval. In this paper, we present an unbiased estimator for the gradient of the log-likelihood function of the MHP, which makes the application of the Stochastic Gradient Descent (SGD) for maximum likelihood estimation straightforward. As the log-likelihood function for the MHP is usually non-convex in the parameter space, even for the basic exponential kernels, the SGD or other optimisation methods do not guarantee a global maximum. However, in our experiments, we observe that SGD, with ADAM (Kingma and Ba [13]) used for the adaptive learning rates, gets sufficiently close to the optimal parameters in a few iterations.

The paper's main contribution is developing a feed-forward neural network-based non-parametric approach to estimate the kernels of the MHP. Specifically, each excitation kernel of the MHP is modelled as a separate feed-forward network with a single hidden layer. The weights of the different networks are coupled in the likelihood function. The optimal weights are then determined using the batch SGD to maximise the log likelihood. A shallow network is used to allow a closed-form expression for the time-integrated value of the excitation kernels. At the same time, by the universal approximation theorem, the network can approximate any excitation kernel with compact support to arbitrary precision. In this paper, we only consider the excitation effect of new arrivals (instead of inhibition effects), i.e., the excitation kernel's output ranges in $\mathbb{R}^+$, and a fixed base intensity. We test our model against a few state-of-the-art non-parametric estimation methods for MHP. The method is tested against both synthetic as well as a real datasets. We consider the high-frequency data of buy and sell market orders from the Binance crypto exchange for the real dataset. We find that our method's performance, which we call the *Shallow Neural Hawkes* (SNH), is comparable to or better than benchmark models. We observe a clear self-excitation behaviour in the buy and sell BTC-USD trades at the Binance exchange. The observed kernel for the self-excitation process in our BTC-USD trade dataset differs from the commonly used parametric kernels, i.e. exponential and power-law kernels. A distinct advantage of our approach compared to recurrent neural networks used to model the MHP is that we do not lose the interpretability of MHP by recovering the underlying excitation kernels. Another advantage is that a closed-form expression for the integrated kernel functions is obtained, which for instance, histogram-based non-parametric methods would require discrete-time approximations.

2. Related work

In many real-world applications, the Hawkes process's flexibility is enhanced by using non-parametric models. The first non-parametric model of the one dimensional Hawkes processes was proposed in Lewis and Mohler [14], based on an ordinary differential equation (ODE). The first extension of non-parametric kernels to the multi-dimensional case was provided in Zhou et al. [7]. They developed an algorithm to learn the decay kernels using Euler–Lagrange equations to optimise infinite-dimensional functional space. Determined to model a large amount of data, a non-parametric method based on solving the Wiener–Hopf equation using a Gaussian quadrature method was introduced in Bacry and Muzy [15]. Motivated by the branching property of the Hawkes process (Zhuang et al. [16]), an Expectation–Maximisation (EM) algorithm was developed in Marsan and Lengline [17] for non-parametric estimation of decay kernel and background intensity.

The methods close to our approach include the MEMIP (Markovian Estimation of Mutually Interacting Processes) Lemonnier and Vayatis [18] that uses polynomial approximation theory and self-concordant analysis to learn the kernels and the base intensities. While the non-parametric models in Lemonnier and Vayatis [18] and Zhou et al. [7] represent excitation functions as a set of basis functions, a guidance for the selection process of basis functions is provided in Xu et al. [19]. Both Xu et al. [19] and Salehi et al. [20] express the excitation kernels as the sum of Gaussian basis kernels; the former uses a sparse group-lasso regulariser and is suitable for large datasets. In contrast, the latter uses variational expectation–maximisation and is suitable for a handful of datasets. The approach presented in this paper is similar, as the excitation function is expressed as a non-parametric function, specifically as the exponential of the sum of rectified linear units (ReLUs).

In a relatively recent study of temporal point processes, the authors in Du et al. [21] develop a recurrent neural network to model point processes and learn influences from event history. The authors in Mei and Eisner [22] develop a novel continuous-time LSTM to model the self-modulating Hawkes processes. This setting can capture the exciting and inhibiting effects of past events on the future and allow the background intensity to take negative values corresponding to delayed response or inertia of some events. Compared to expressing each excitation kernel as a neural network, LSTM might be less desirable when there is a greater focus on the interpretability of the MHP, for instance, for learning the Granger causality graph. We also significantly simplify the SGD formulation compared to Mei and Eisner [22], where one has to rely on simulations to obtain the gradients, while in Du et al. [21] numerical integration is needed to get the necessary gradients of the log-likelihood.

The compelling rise in the cryptocurrency asset prices has naturally gained consideration among investors, economists and market researchers. Notably, Bitcoin, the world's most renowned digital currency, has seen an unprecedented rise. In Koutmos [23], the author investigates the extent to which market risk factors can explain Bitcoin's price behaviour. In another study on Bitcoin price dynamics (Giudici and Polinesi [24]), the authors learn about the correlation of Bitcoin prices from different exchanges. The authors in Akyildirim et al. [25] compare four machine learning classification algorithms to predict the direction of Bitcoin returns (upward or downward price moves). Recently Goodell and Goutte [26] studied the co-movement between COVID-19 levels and Bitcoin prices. Philippas et al. [27] study the dependence between media attention and Bitcoin prices and find that Bitcoin prices are partially driven by the momentum of media attention on social networks.

Similarly, a growing literature is dedicated to studying point processes to high-frequency financial data. In particular, due to the trading activity's correlated and clustered nature, the Hawkes processes are used to model market order arrival dynamics. Bowsher [28] used a continuous-time bivariate Hawkes process for modelling the arrival times of market sell and buy orders. Recently, a bivariate Hawkes process was proposed in Bacry et al. [29] to model the variations of the asset prices and study the signature plot and the Epps effect.

3. Preliminary definitions

A D-dimensional MHP is a collection of D univariate counting processes $N_d(t), d = 1, \dots, D$. The realisation of the MHP over an observation period $[0, T)$ consists of a sequence of discrete events $\mathcal{S} = \{(t_n, d_n)\}$, where $t_n \in [0, T)$ is the time-stamp of the nth event and $d_n \in \{1, \dots, D\}$ is the label of corresponding dimension in which the event occurred. The conditional intensity process for the d−th dimension is given by Eq. (1).

Often, the Hawkes kernels are assumed to have a parametric form and the base intensity μ_d is assumed to be constant. Two widely used parametric kernels include,

$$\text{the exponential kernel:} \quad \phi_{dj}(t) = \alpha_{dj} e^{-\beta_{dj} t}, \tag{2}$$

and

$$\text{the power-law kernel:} \quad \phi_{dj}(t) = \alpha_{dj} (\delta_{dj} + t)^{-\beta_{dj}}, \tag{3}$$

where $d = 1, \dots, D$, $j = 1, \dots, D$, and α_{dj}, β_{dj}, $\delta_{dj} \in \mathbb{R}^+$ are the adjacency, decay, and the lag parameters respectively. In this paper, we assume that $\phi_{dj} : \mathbb{R}^+ \to \mathbb{R}^+$ can be an arbitrary continuous function with compact support in $\mathbb{R}^+$. We also assume that μ_d is a positive constant.

We denote the parameters of the multi-dimensional Hawkes process in a matrix form as $\boldsymbol{\mu} = [\mu_1, \dots, \mu_D]^\top$ for the base intensity, and $\boldsymbol{\Phi} = (\phi_{dj})$ for the excitation kernels. These parameters can be estimated by optimising the log-likelihood over the observed events that are sampled from the process. The log-likelihood for model parameters $\Theta = \{\boldsymbol{\Phi}, \boldsymbol{\mu}\}$ of the Hawkes process can be derived from its intensity function (see for instance Rubin [30], Daley and Vere-Jones [31]) and is given by,

$$\begin{aligned}\mathcal{L}(\Theta) &= \sum_{d=1}^{D}\left(\int_0^T \log\left(\lambda_d(u)\right)\, dN_d(u) - \int_0^T \lambda_d(s)\, ds\right)\\ &= \sum_{d=1}^{D}\left(\sum_{(t_n,d_n)\in\mathcal{S}}\left(\log\left(\lambda_d(t_n)\right)\mathbb{1}\{d_n = d\}\right) - \int_0^T \lambda_d(s)\, ds\right). \qquad (4)\end{aligned}$$

For the application of the SGD, we need an unbiased estimator for the gradient of $\mathcal{L}$ with respect to model parameters. Obtaining an unbiased gradient estimator for $\int_0^T \lambda_d(s)\,ds$ is challenging. Mei and Eisner [22] use a simulation-based approach for an unbiased estimate, while Yang et al. [32] work with a time-discretised version of $\mathcal{L}$. Both these approaches are computationally intensive. We propose the following as an unbiased estimator for the gradient of the log-likelihood function $\mathcal{L}$.

Let $\{t_1^d, \dots, t_{N_d(T)}^d\}$, be the ordered arrival times for the nodes $d = 1, \dots, D$. We first focus on the integral of the intensity with respect to time.

$$\begin{aligned}\int_0^T \lambda_d(s)\, ds &= \int_0^T \left(\mu_d + \sum_{t_n<s} \phi_{dd_n}(s - t_n)\right) ds\\ &= \int_0^T \mu_d\, ds + \int_0^T \left(\sum_{t_n<s} \phi_{dd_n}(s - t_n)\right) ds \qquad (5)\end{aligned}$$

We can write the first part of the integral as

$$\int_0^T \mu_d\, ds = \sum_{t_n^d<T} \int_{t_{n-1}^d}^{t_n^d} \mu_d\, ds \qquad (6)$$

The second part of the expression in Eq. (5) can be written as follows,

$$\begin{aligned}\int_0^T \sum_{t_n<s} \phi_{dd_n}(s - t_n)\, ds &= \sum_{(t_n,d_n)\in\mathcal{S}} \int_{t_n}^{t_{n+1}} \sum_{t_m<t_n} \phi_{dd_m}(s - t_m)\, ds,\\ &= \sum_{(t_n,d_n)\in\mathcal{S}} \int_{t_n}^{T} \phi_{dd_n}(s - t_n)\, ds,\\ &= \sum_{(t_n,d_n)\in\mathcal{S}} \int_0^{T-t_n} \phi_{dd_n}(s)\, ds,\\ &= \sum_{j=1}^{D}\sum_{t_i^j<T} \int_0^{T-t_i^j} \phi_{dj}(s)\, ds. \qquad (7)\end{aligned}$$

The first equality above is from partitioning the interval $[0, T)$ by the arrival times; the second equality comes from the fact that the term $\phi_{dd_n}(s - t_n)$ will appear in all integral partitions greater than t_n, while a basic change of variable obtains the third equality. The final equality is the outcome of the rearrangement of terms.

Finally, the following relation is obtained by rearranging the terms

$$\sum_{d=1}^{D}\sum_{j=1}^{D}\sum_{t_i^j<T} \int_0^{T-t_i^j} \phi_{dj}(s)\, ds = \sum_{d=1}^{D}\sum_{j=1}^{D}\sum_{t_i^d<T} \int_0^{T-t_i^d} \phi_{jd}(s)\, ds. \qquad (8)$$

Substituting Eq. (6) and (8) into Eq. (3) gives us:

$$\mathcal{L}(\Theta) = \sum_{d=1}^{D}\left(\sum_{t_n^d<T}\left(\log\left(\lambda_d(t_n^d)\right) - \int_{t_{n-1}^d}^{t_n^d} \mu_d\, ds - \sum_{j=1}^{D}\int_0^{T-t_n^d} \phi_{jd}(s)\, ds\right)\right).$$

Table 1
The mean values of μ_{11}, α_{11}, and β_{11} estimated from the simulated data of the univariate Hawkes process. The actual parameters used for simulation are in the order [μ_{11}, α_{11}, β_{11}]. s.e. is the standard error computed using the outcome of 30 independent runs.

μ_{11} (s.e.)	α_{11} (s.e.)	β_{11} (s.e.)	Actual parameters
1.012 (0.044)	0.498 (0.0211)	2.005 (0.208)	[1,0.5,2]
2.05 (0.051)	3.07 (0.037)	9.57 (0.32)	[2,3,10]
0.489 (0.0128)	186.1 (5.68)	585.38 (10.24)	[0.5,200,600]

Therefore gradient of $\mathcal{L}$ is:

$$\nabla_\Theta\mathcal{L}(\Theta) = \sum_{d=1}^{D}\left(\sum_{t_n^d<T} \nabla_\Theta\left(\log\left(\lambda_d(t_n^d)\right) - \int_{t_{n-1}^d}^{t_n^d} \mu_d\, ds - \sum_{j=1}^{D}\int_0^{T-t_n^d} \phi_{jd}(s)\, ds\right)\right),$$

Which gives us the unbiased estimator.

$$\nabla_\Theta\left(\log(\lambda_{d_n}(t_n)) - \int_{t_n^-}^{t_n} \mu_{d_n}\, ds - \sum_{j=1}^{D}\int_0^{T-t_n} \phi_{jd_n}(s)\, ds\right), \qquad (9)$$

where $(t_n, d_n) \in \mathcal{S}$ and $t_n^- := \max_{t_m}\{t_m | t_m < t_n \wedge d_m = d_n\}$, i.e. t_n^- is the timestamp of the event that occurred just prior to the event at t_n for node d_n. A challenge in efficiently utilising Eq. (9) in the SGD method is that we need a closed-form expression for computing $\int \phi_{dj}(s)\,ds$. When a parametric form for the excitation kernel is assumed, usually closed-form expression for this integral exists. In Section 3.1 we present results for parameters inferred using SGD for exponential kernels and find that the results are close to the true parameter values. However, in Section 4, we present a non-parametric approach, which is general enough to infer any continuous excitation kernel and has closed-form expression for the integrated excitation kernel.

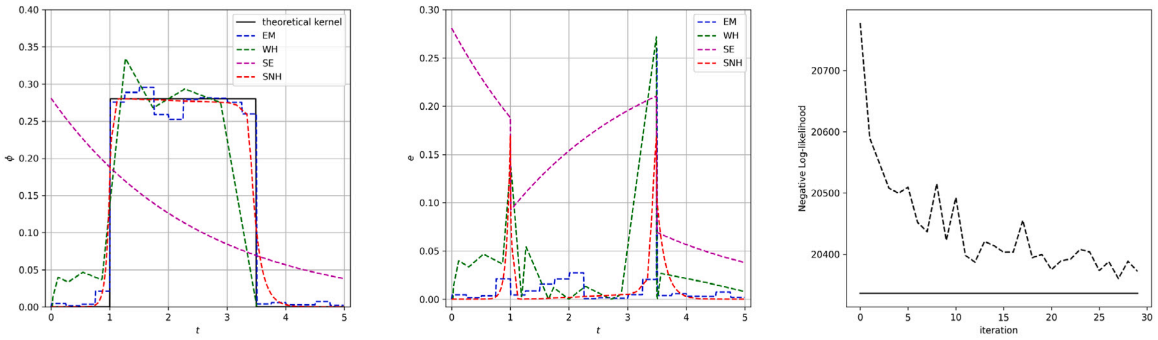
3.1. Parameter estimation for MHP using SGD

The kernels of the Hawkes process are often assumed to have a parametric form, and the parameters are typically estimated by minimising the negative log-likelihood over the observed events. However, even for the basic exponential Hawkes process, whose kernels are given by Eq. (2), the log-likelihood function is non-convex in the parameter space. Most methods fix the value of the decay β_{dj} (where $d = 1, \dots, D$, and $j = 1, \dots, D$) and optimise upon the α_{dj} (see for instance [33]). One then needs to try several fixed values for the decay parameter β_{dj} and finally select the decay and adjacency combination that resulted in the lowest negative log-likelihood.

On the other hand, the SGD method is well suited for non-convex optimisation. We use the batch SGD method to estimate the univariate and the bivariate exponential Hawkes process parameters. The unbiased estimate of the gradient of the log-likelihood function with respect to the parameters, α_{dj}, and β_{dj} is computed following Eq. (9).

We first consider the univariate exponential Hawkes process. The arrival times are simulated using Ogata's thinning algorithm [34] for three choices of parameter values as given in column *Actual Parameters* of the Table 1. We simulate the process for a period of $[0, 5000)$. The initial guess for the parameter values is drawn from a uniform distribution between 0 and 1. We use ADAM for adaptive learning rates, with the learning rate set to 0.01 and a batch size of 32 used for the SGD.

Table 1 shows the mean value and standard errors (from 30 trials) of the estimated parameters using the SGD. We find that the estimated parameters are close to the parameter values used for the simulation. Even when the actual parameters have large adjacency and decay values, the method can accurately estimate them.



(a) (b) (c)

Fig. 1. Synthetic data experiment results for univariate Hawkes processes (a) The estimated rectangular kernel. (b) The estimated error for the rectangular kernel. (c) The convergence plot of negative log-likelihood for the rectangular kernel estimation in the SNH model.

Table 2
Estimated $\boldsymbol{\mu}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ matrix for the exponential MHP. The actual parameters used for simulation are $\mu_d = 0.5$, $\alpha_{dj} = 200$, $\beta_{dj} = 600$, where $1 \le d, j \le 2$.

$\boldsymbol{\mu}$	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$
$\begin{bmatrix} 0.514 \\ 0.528 \end{bmatrix}$	$\begin{bmatrix} 185.03 & 192.35 \\ 186.48 & 195.09 \end{bmatrix}$	$\begin{bmatrix} 589.24 & 561.29 \\ 555.92 & 572.37 \end{bmatrix}$

Table 2 shows the corresponding parameter values estimated for a bivariate exponential Hawkes Process using the batch SGD to minimise the negative log-likelihood in the parameter space. The choice of hyper-parameters is the same as that for the 1-D case, and the initial guess for the parameter values is again drawn from uniform random between 0 and 1. We see that using the unbiased gradient estimator, given by Eq. (9), both the decay and adjacency parameter values can be simultaneously estimated using the SGD method. These experiments also validate that we have obtained the correct expression for the unbiased estimator.

4. Proposed model

A feed-forward network with a single hidden layer, a sufficiently large number of neurons, and an appropriate choice of the activation function is known to be a universal approximator Hornik et al. [35]. We, in the proposed method, model each excitation kernel $\phi_{dj}(t)$, $1 \le d, j \le D$ of the MHP using a separate feed-forward network with a single hidden layer. As we consider only excitation kernels, the output of each of these neural networks should be in $\mathbb{R}^+$. The weights of the different networks are coupled together in the likelihood function. We use the batch stochastic gradient descent to maximise the log-likelihood over the parameter space. The unbiased estimates of the gradient of the log-likelihood are obtained using Eq. (9). For efficient calculation of the gradient, as discussed in Section 3 ideally, there should be a closed-form expression for the time-integrated value of the approximated excitation kernel. Based on these criteria, a) a positive output for the approximated excitation kernel and b) a closed-form expression for its integral, we developed a specific architecture for our neural network.

In order to approximate $\phi_{dj}(t)$, $1 \le d, j \le D$ we use a feed-forward network $\hat{\phi}_{dj} : \mathbb{R} \to \mathbb{R}^+$ of the form

$$\hat{\phi}_{dj} := \psi \circ A_2 \circ \varphi \circ A_1$$

where $A_1 : \mathbb{R} \to \mathbb{R}^p$ and $A_2 : \mathbb{R}^p \to \mathbb{R}$ are affine functions of the form,

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

and

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

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

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

while $\psi : \mathbb{R} \to \mathbb{R}+$, is exponential function

$$\psi(x) := e^x$$

With a choice of p neurons for the hidden layer, the dimension of the parameter space for the network will be $3p + 1$. For a D-dimensional Hawkes process we would need D^2 networks. Writing $\mathbf{W}_1 := [a_1^1, \dots, a_1^p]^\top$, $\mathbf{W}_2 := [a_2^1, \dots, a_2^p]$, $\mathbf{b}_1 := [b_1^1, \dots, b_1^p]^\top$, the approximate kernel can be written as:

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

The choice of exponential function for the output layer is to ensure that the output is in $\mathbb{R}^+$ as required by excitation kernels. As the ReLU activation function is not a polynomial everywhere, the network will be a universal approximator (Leshno et al. [36]). The other advantage of this particular choice of network architecture is that a closed-form expression for $\int_0^t \hat{\phi}_{dj}(u)\, du$ can be readily evaluated. It turns out that it is a linear combination of $\hat{\phi}_{dj}$ as discussed in Section 4.1. The optimal parameters for the MHP, i.e. $\Theta = \{\boldsymbol{\Phi}, \boldsymbol{\mu}\}$, where $\boldsymbol{\Phi}$ is the set of weights of all the D^2 networks, are obtained using batch SGD, where we use ADAM for the adaptive learning rates.

4.1. Integrated shallow excitation kernel

The SNH models each excitation kernel $\phi_{dj}(t)$, as

$$\hat{\phi}_{dj}(t) = \exp\left(b_2 + \sum_{i=1}^{p} a_2^i \max\left(a_1^i x + b_1^i, 0\right)\right), \tag{10}$$

Where p is the number of neurons used in the hidden layer. The unbiased estimator in Eq. (9) requires us to compute the gradient of the integrated excitation kernel, i.e.

$$\int_0^t \phi_{dj}(s)\, ds.$$

Integrating $\hat{\phi}_{dj}(t)$ involves integrating a function of max functions (as given in Eq. (10)). Let $\mathbf{X} \equiv \{x_1, \dots, x_p\}$ be the sequence of inflection points obtained for the p neurons, where the inflection point of the ith neuron is obtained by solving,

$$\begin{aligned} a_1^i x_i + b_1^i &= 0 \\ x_i &= -\frac{b_1^i}{a_1^i}. \end{aligned}$$

We re-index the sequence $\mathbf{X}$ in monotone increasing order as $\mathbf{S} \equiv \{s_1 \leq s_2 \leq \cdots \leq s_p\}$. Let $0 \leq s_l \leq \cdots \leq s_u \leq T$, where $1 \leq l \leq u \leq p$ be the largest subsequence of the sorted sequence $\mathbf{S}$ of inflection points, i.e. all the inflection points that lie in the range $[0, T]$. Between any two adjacent inflection points, $\widehat{\phi}_{dj}(t)$, is an exponential of a linear function in t, which can be readily integrated. Therefore, we write the above integral as,

$$\int_0^t \widehat{\phi}_{dj}(s)\, ds = \int_0^{s_l} \widehat{\phi}_{dj}(s)\, ds + \cdots + \int_{s_u}^T \widehat{\phi}_{dj}(s)\, ds, \tag{11}$$

where the solution of the definite sub-integral between consecutive sorted inflection points, $0 < s_m < s_n < T$, is given by:

$$\int_{s_m}^{s_n} \widehat{\phi}_{dj}(s)\, ds = \frac{1}{\sum_{i=1}^p a_1^i a_2^i \mathbb{1}\left\{\lim_{x \to s_n^-} a_1^i x + b_1^i > 0\right\}} \times \left(\widehat{\phi}_{dj}(s_n) - \widehat{\phi}_{dj}(s_m)\right).$$

5. Experiments and results

5.1. Synthetic data

In this section, we demonstrate the performance of the Shallow Neural Hawkes model by fitting various forms of kernels and by comparing it against the state-of-the-art non-parametric models, including the EM method given in Lewis and Mohler [14] and the Wiener–Hopf (WH) model described in Bacry and Muzy [15]. All simulations are performed using the thinning algorithm described in Ogata [34]. We also use a large set of tools from the tick library, Bacry et al. [33], that facilitates efficient parametric and non-parametric estimations. We first conduct numerical experiments for the univariate case of the Hawkes process and then report the findings for the bivariate case.

5.1.1. Univariate case

We consider the univariate Hawkes process with the following kernels:

1. the exponential kernel, given by Eq. (2), with parameters $[\alpha, \beta, \mu] = [1, 4, 0.05]$,
2. the power-law kernel, given by Eq. (3), with parameters $[\alpha, \beta, \delta, \mu] = [1, 4, 1, 0.05]$,
3. and the rectangular kernel, given by Eq. (12), with parameters $[\alpha, \beta, \delta, \mu] := [0.7, 0.4, 1, 0.05]$,

for our study.

$$\phi(t) = \begin{cases} \alpha\beta, & \text{if } \delta < t < \delta + \dfrac{1}{\beta} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

The arrival times are simulated from $[0, 60000]$ for all the cases. We get $N_T = 3972$, 4442, and 10196 events for the above three cases. Once the arrival times have been simulated, we use the SNH to infer the excitation kernel for each case.

Experiment setup: We use 100 neurons to model each kernel, the initial weights are drawn from a uniform distribution in the range of $[0, 0.5]$. In all our initialisations, we find that positive weights for the inner layer and negative weights for the outer layer help faster convergence of the method. We use the ADAM optimiser (Kingma and Ba [13]) for adaptive learning rates, with a learning rate of 10^{-2} and 5×10^{-3} used for the inner and outer layer, respectively. We use a learning rate of 10^{-3} for determining the optimal μ value. A batch size of 50 is used to compute the gradient for the SGD, and we train the network up to 30 epochs. To avoid over-fitting, early-stopping criteria can be used, where the algorithm terminates when the updated parameters have not improved the best-recorded validation error for some pre-specified number of iterations [37].

The learned kernels from the Shallow Neural Hawkes model are then compared to kernels determined by the parametric sum of the exponential kernels (SE) method, the non-parametric EM, and the WH model. The reference models are implemented using the tick library (Bacry et al. [33]). We choose the kernel support for the non-parametric EM estimation as 5 and a kernel size of 20. For the WH method, we set the number of quadratures as 50 and use linear sampling for the exponential kernels. The linear sampling method performs poorly in the power-law kernel and the rectangular kernel. Hence we use the semi-log sampling approach with the maximum kernel support of 1000 and a maximum lag of 100.

Experiment results: Using the sampled arrival times, the SNH method can closely recover all the three cases of the excitation kernels. The recovered kernels from the SNH method are closest to the true kernels for the exponential and the power-law Hawkes process. The results for these two cases are discussed in the Appendix. Fig. 1 reports the results for the univariate Hawkes process with a rectangular kernel. First, we compare the performance of the models based on the kernel estimation. The EM model is a histogram-based estimator with a discrete function kernel, whose performance critically depends on the bins' choice. The WH model also has a strong dependency on the grid's choice in the kernel estimation process Morzywolek [38]. We find that while the non-parametric methods, SNH, WH, and the EM, can reasonably recover the kernel, the sum of the exponential approach has the poorest fit. Fig. 1(b) reports the L1 error of the recovered kernels for the four methods. The SNH errors are lowest except at the edges, where the EM method performs better. Fig. 1(c) shows the negative log-likelihood values achieved by the SNH at different epochs. Within a few epochs, the negative likelihood values obtained by the SNH are reasonably close to the value obtained using the true kernel.

5.1.2. Bivariate case

For the bivariate analysis, we simulate using the tick library a bivariate Hawkes process with non-standard kernels. While not reported here, we also study the performance of the SNH in fitting the bivariate exponential and power-law kernel. We get similar outcomes as the univariate case, where the SNH outperforms the EM, WH, and SE. We focus here on the non-standard kernels that test the versatility of the proposed method

Experiment setup : The non-standard kernels are simulated using the class *TimeFunction* from the tick library, which uses several types of interpolation to determine the function value between two points on $[0, \infty)$ Bacry et al. [33]. The kernel function $\phi_{(0,0)}(t)$ is defined using

$$x = \begin{bmatrix} 0.0 & 1.0 & 1.5 & 2.0 & 3.5 \end{bmatrix}, \quad y = \begin{bmatrix} 0.0 & 0.2 & 0.0 & 0.1 & 0.0 \end{bmatrix}$$

and the y-values are extended to the right. Next, we have $\phi_{(0,1)}(t) = \frac{sin(t)}{4}$ for $0 < t < T$. We then generate a zero kernel $\phi_{(1,0)}(t) = 0$. Finally, we simulate a non-standard form kernel for $\phi_{(1,1)}(t)$ using

$$x = \begin{bmatrix} 0.0 & 0.7 & 2.5 & 3.0 & 4.0 \end{bmatrix}, \quad y = \begin{bmatrix} 3.0 & 0.03 & 0.03 & 0.2 & 0.0 \end{bmatrix}.$$

The baseline intensity is set to

$$\mu = \begin{bmatrix} 0.05 \\ 0.05 \end{bmatrix}.$$

Fig. 2 illustrates the four kernels used in the bivariate Hawkes process. The arrival times are simulated in the sample period $[0, 60000)$. To recover the non-standard kernels using the SNH model, we use the same initialisation technique for the network weights as used in the univariate case. Also the same learning rates are used as the univariate case for the ADAM algorithm. We train the network for 100 epochs and use a batch size of 50 for computing the necessary gradients.

Experiment results : The SNH model's performance in recovering the non-standard kernels is shown in Fig. 2. We see that the kernel setting in $\phi_{0,0}(t)$ is advantageous to the EM (as the underlying histograms are rectangular). As expected, the EM reports the best recovery for $\phi_{0,0}(t)$.

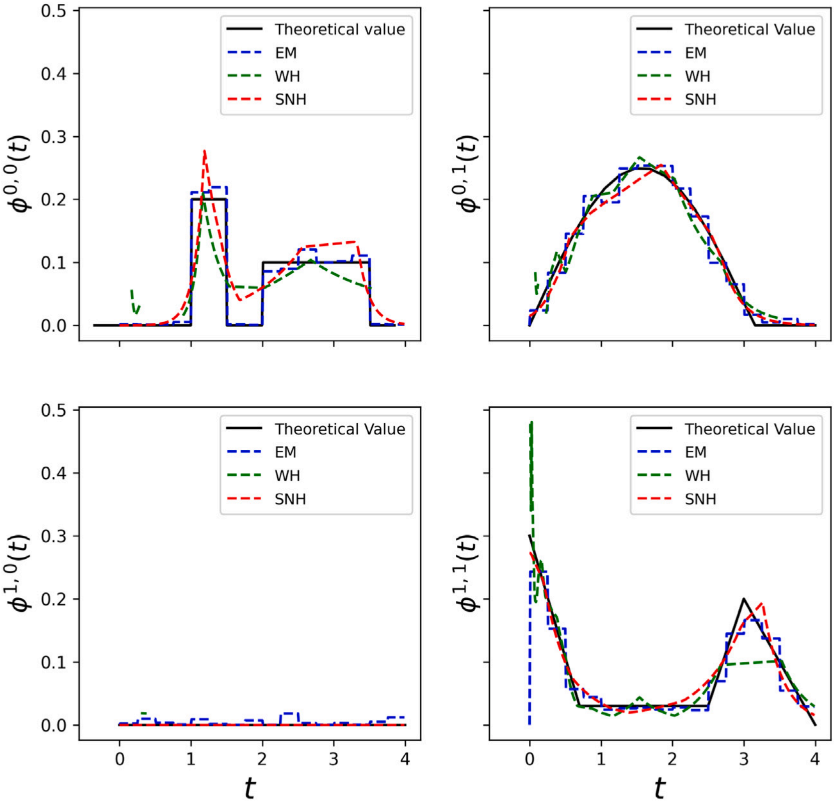
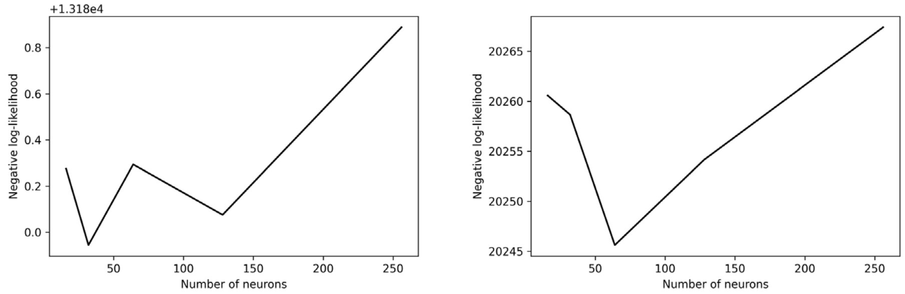


Fig. 2. Synthetic data experiment results for bivariate Hawkes processes with the non-standard kernels. Theoretical value corresponds to the kernel values used for simulating the arrival times. The recovered kernel values from the SNH, WH, and the EM are reported for the four kernels.



(a)

(b)

Fig. 3. Plot of estimated negative log-likelihood for varied number of neurons (a) The case of exponential kernels (b) The case of rectangular kernels.

In the case of kernels $\phi_{0,1}(t), \phi_{1,0}(t), \phi_{1,1}(t)$, the SNH model achieves better results when compared to other models (see Fig. A.9 for the corresponding errors in recovering the kernels). This experiment proves that the SNH can simultaneously recover the kernels of an MHP, the errors in the recovered kernels are lower or comparable to the other state-of-the-art methods, and it can recover non-standard kernels.

5.1.3. *Choice of hyper-parameters*

We extend our analysis for the univariate Hawkes process described in Section 5.1 to study the effect of hyper-parameter choices for the SNH. We first study the impact on the performance of the SNH model with a varied number of neurons. Fig. 3 shows the estimated negative log-likelihood values with an increasing number of neurons used in the SNH for the exponential form of the kernel. As expected, we find that fewer neurons are sufficient to achieve convergence. Next, we perform a similar analysis on the rectangular kernel described in Section 5.1, and Fig. 3 illustrates the corresponding results. In this case, it is evident that optimal performance is achieved by using neurons in the range 32 to 128 in the SNH architecture.

We investigate the SNH model's performance based on different initial learning rates. We use the univariate Hawkes process with a rectangular kernel as described in Section 5.1 for this analysis. From the study, we conclude that a higher initial learning rate for the outer layer compared to the inner layer helps in faster convergence (see Tables 3 and 4).

5.2. *Self and cross excitation effects in the bitcoin market order arrivals at the binance exchange*

This paper investigates the Shallow Neural Hawkes model's performance on BTC-USD order book data (from the Binance exchange) to understand this critical cryptocurrency's microstructure. We streamed the Binance exchange order book data, as several popular cryptocurrencies are traded in this exchange, and the exchange has high trade volumes. The data includes the tick-by-tick buy and sell market order

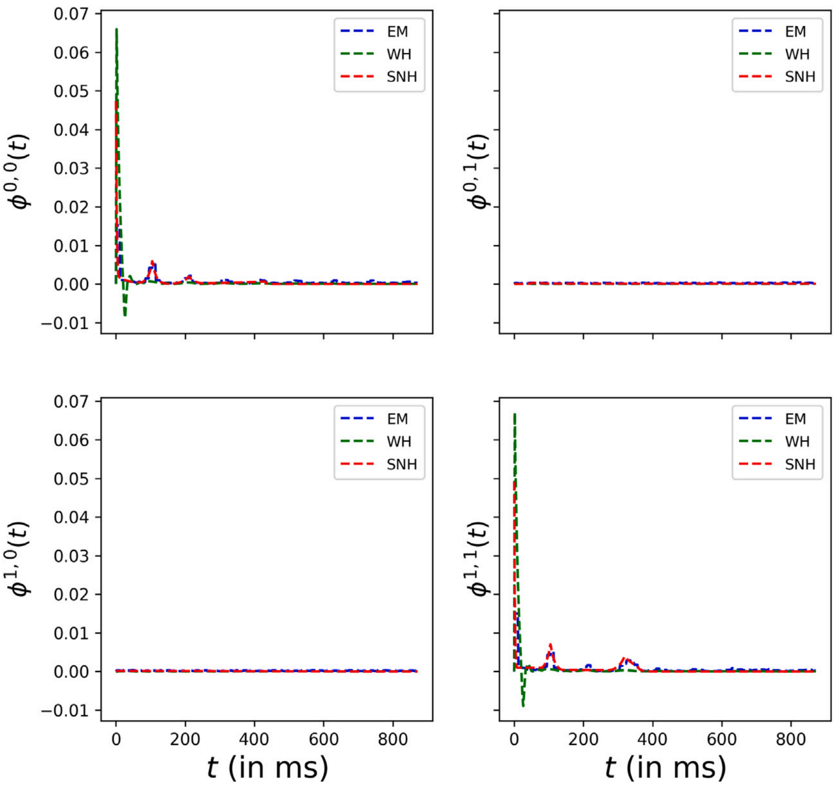


Fig. 4. Experiment results of Bitcoin data as bivariate Hawkes processes. Estimated μ_1 (base rate for sell trade) is 1.2×10^{-3}, 9×10^{-4}, 2×10^{-4} for the SNH, the EM, and the WH respectively. Estimated μ_2 (base rate for buy trade) is 1.1×10^{-3}, 7×10^{-4}, 2×10^{-4} for the SNH, the EM, and the WH respectively. The negative log-likelihood values are 5.2×10^6, 5.4×10^6, and 7.5×10^6 for the SNH, the EM, and the WH respectively.

Table 3
Estimated negative log-likelihood for different learning rates of the outer layer of SNH model performed for 30 epochs (Lr of inner layer = 5×10^{-3}).

Lr of outer layer	Neg loglik
0.0001	37113
0.005	20244
0.001	20562
0.05	20765
0.01	20201

Table 4
Estimated negative log-likelihood for different learning rates of the inner layer of SNH model performed for 30 epochs (Lr of outer layer = 10^{-2}).

Lr of inner layer	Neg loglik
0.00001	21369
0.00005	20483
0.0005	20350
0.0001	20321
0.005	20587
0.01	22346

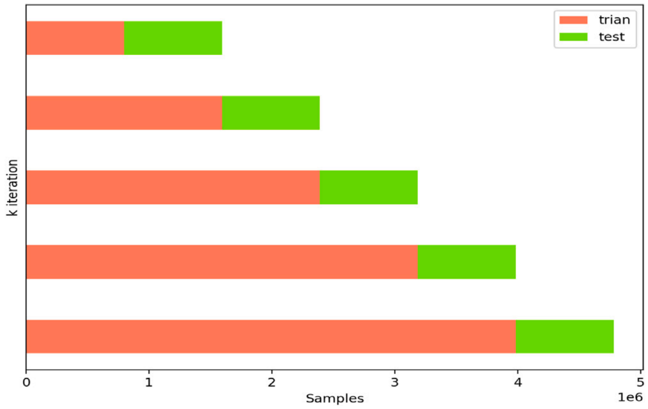


Fig. 5. TimeSeriesSplit function on bivariate Bitcoin dataset (buy and sell data) with $N_T = 4,785,363$ and number of splits = 5.

arrival times for the Bitcoins (BTC-USD pair). Markets are made of makers and takers. As the Binance APIs explain, the makers create buying or selling orders that are not carried out immediately, thereby creating liquidity for that cryptocurrency. On the other hand, people that buy or sell instantly are called the takers. The trade arrival data consists of limit orders, which are orders placed with a limit price and market orders, which are orders placed to buy or sell at the current available price. The limit orders are executed when the market prices reach the set limit, and on the other hand, market orders are executed instantly at the current best market price. The Binance trade-stream data displays the timestamp of these orders' arrival, with price and volume features and a unique ID for the buyer/seller. A particular market order might require several limit orders to fill the demanded volume, resulting in several trades recorded with a common ID. Hence we cleaned the dataset by filtering the data for common IDs to include only the unique trade events. Finally, depending upon whether the buyer is a market maker or taker, the dataset was marked as buy or sell market orders.

Our goal is to examine the dependence between these events, suggesting self-excitation or cross excitation or both between the arrival of buy and sell market orders for the BTC-USD pair. Unlike the experiments in the previous synthetic data section, the shape of the excitation function of BTC trades is unknown and such is the case for all real-world events. This necessitated developing a non-parametric approach to estimate the underlying Hawkes model.

5.2.1. Experiment setup

We use the BTC-USD pair data traded in the Binance cryptocurrency exchange. The full dataset consists of nearly 7,002,171 intraday market orders, as recorded from 12 May 2020 at 12:00 AM to 21 May at 11:00 PM (UTC). Identical sell and buy trade orders were removed from the dataset. We performed univariate analysis separately on the arrival times of buy orders ($N = 2,485,932$) and the arrival times of sell orders ($N = 2,295,554$). Using the univariate analysis, we can conclude whether the arrival of a market order results in self-excitation.

Bivariate analysis is performed jointly on the buy and sell trade data to learn their interactions, specifically if there is any cross-excitation between the buy and the sell market orders. For the SNH network architecture, we use the same initial settings as in the synthetic data instance. With $N_{T_{\text{sell}}} + N_{T_{\text{buy}}} = 4,781,486$, we train the network in 30 epochs.

We perform non-parametric analysis on the Bitcoin dataset using the EM and the WH models to facilitate comparison. We choose the kernel support for the EM estimation as 6 and a kernel size of 100. We set the number of quadratures for the WH method at 200.

5.2.2. Experiment results

In Fig. 4, we plot the kernels estimated by the SNH, the EM, and the WH methods. It is evident that the two events are not mutually exciting but exhibit self-exciting behaviour. The negative log-likelihood values recorded from the SNH, the EM and the WH models are 5.2×10^6, 5.4×10^6, and 7.5×10^6, respectively. The SNH model achieves competitive negative log-likelihood compared to the EM and the WH models. The WH method exhibits consistently poor results, while the performances of the EM and the SNH methods are comparable. To further validate the SNH method, we perform the k-fold test on the arrival data as described below.

5.2.3. K-fold cross validation of real data

Cross-validation is one of the most widely used methods for evaluating learning algorithms. Ideally, we divide the dataset into a training set, cross-validation set, and test set to optimise the parameters, evaluate each algorithm and finally test the successful algorithm with the least error. However, when the data is scarce or limited, we are left with fewer samples in the training set. As a solution to this problem, we use the k-fold cross-validation method Friedman et al. [39] to test our model's performance. In this method, we divide the dataset into k-groups, and for each group, we split the training and test set to evaluate the score. The performance measure is the average of the evaluated scores of the k-groups, given as,

$$CV(\Theta) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(\Theta) \tag{13}$$

For the dataset in our experiment, we use the *TimeSeriesSplit* function provided by Scikit-learn [40]. Unlike non-time series data where the dataset is randomly split, this function divides the dataset along the sequence. Successive training sets are supersets of those that come before them. Due to the dependence on history in the Hawkes processes, we modify the split function to evaluate the negative log-likelihood collectively on training and test samples (rather than just test samples). The Fig. 5 demonstrates the time-series cross-validation split on bivariate Bitcoin data, for K = 5 groups. Training sets and their corresponding test dataset sizes are given in Table 5. The estimated score, i.e. the negative log-likelihood values for the SNH, the EM and the WH models are 3.5×10^6, 3.8×10^6 and 4.8×10^6, respectively.

5.2.4. Why is it essential to accurately recover the kernels of the Hawkes process?

Accurate estimation of the underlying kernels of the Hawkes process is desirable as it provides:

- an accurate description of the self and cross interactions between arrival processes,
- it helps in the development of better predictive models.

Table 5
Sizes of train-test dataset.

	k = 1	k = 2	k = 3	k = 4	k = 5
Train	796,916	1, 593, 830	2, 390, 744	3, 187, 658,	3, 984, 572
Test	1, 593, 830	2, 390, 744	3, 187, 658	3, 984, 572	4, 781, 486

We explain the above points in the context of the market order arrival times for the BTC-USD pair, previously discussed in Section 5.2. The recovered kernels in Fig. 4 provide some descriptive information about the trade order arrival process. The first obvious inference is that the arrival of a buy (sell) market order increases the intensity of the arrival of the next buy (sell) market order; in other words, there is an observable self-excitation behaviour. One can also clearly infer that a buy (sell) market order arrival has no impact on the intensity of the sell (buy) order arrival; i.e., there is no observable cross excitation between the buy and sell market orders. The self-excitation kernel, $\phi^{0,0}(t)$, shows that a sell order arrival rapidly increases the intensity of the next sell order arrival. This increased intensity quickly decays but is followed by subsequent minor peaks at an interval of roughly 100 ms. We make similar observations about the self-excitation behaviour of the buy market orders. Such insights could be valuable for algorithmic trading as it gives information on possibly the required latency for executing the market orders.

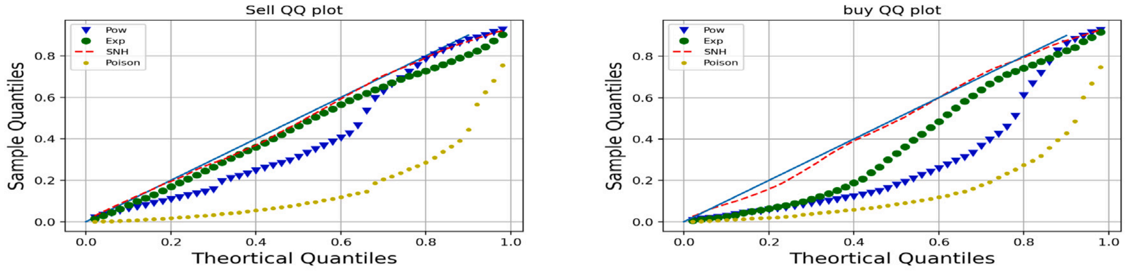
Accurate inference of the kernels is essential for developing a better predictive model for the arrival process. For instance, a trading algorithm is based on predicting, with a 90% confidence, the time *before* which the next market order will arrive. If the prediction model is accurate, one would expect nearly 90% of the predictions would be correct, and for roughly 10% of the cases, the order would arrive *after* the predicted time. If the model always makes an accurate prediction (does not fail for 10% of the cases), then the predictions are overly conservative, i.e., the predicted time is set too far in the future. On the other hand, if more than 10% of the predictions fail, the predicted time is closer than expected. Therefore, if the trading algorithm requires a prediction with Q% certainty, an accurate prediction model should have Q% of correct outcomes. The accuracy of such a prediction model can be visualised using the QQ plot.

For the BTC-USD market order arrival times, we would first like to compare the predictive ability of a fitted parametric (exponential and power-law) Hawkes process with a fitted homogeneous Poisson process. We also study if a non-parametric estimation of the Hawkes process results in better predictions when compared with the parametric Hawkes process. To evaluate the accuracy of the predictions in the test dataset, we make the QQ plots for each model. Fig. 6 illustrates the QQ plots for the sell and buy market orders for the BTC-USD pair. We see that the homogeneous Poisson process has the lowest accuracy and can conclude that the model's predictions for the next arrivals are sooner than that from an ideal model. Between the parametric models, the exponential Hawkes process performs better than the Hawkes process with the power-law kernels. Also, the SNH performs better when compared with the parametric models, as its QQ plot is closest to the 45-degree line (the QQ plot for the ideal model).

We can also compare the three non-parametric models, the SNH, the EM, and the WH using the QQ plots. Fig. 7 clearly illustrates that the three models are comparable, with a slightly poor outcome for the WH method.

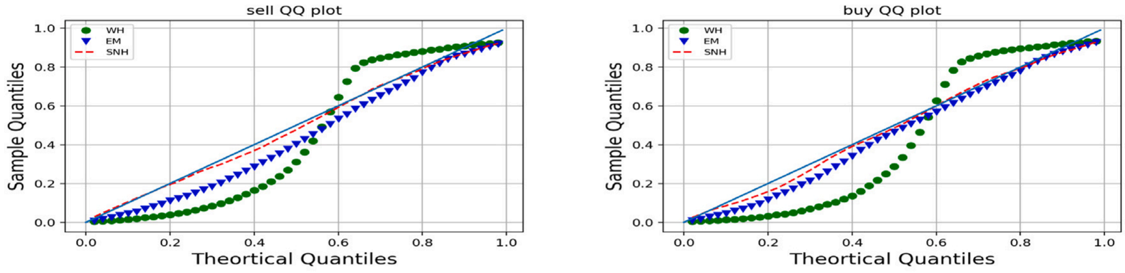
6. Conclusion, limitation and future directions

We have developed a non-parametric kernel estimation method for the MHP, called the Shallow Neural Hawkes. The SNH models the



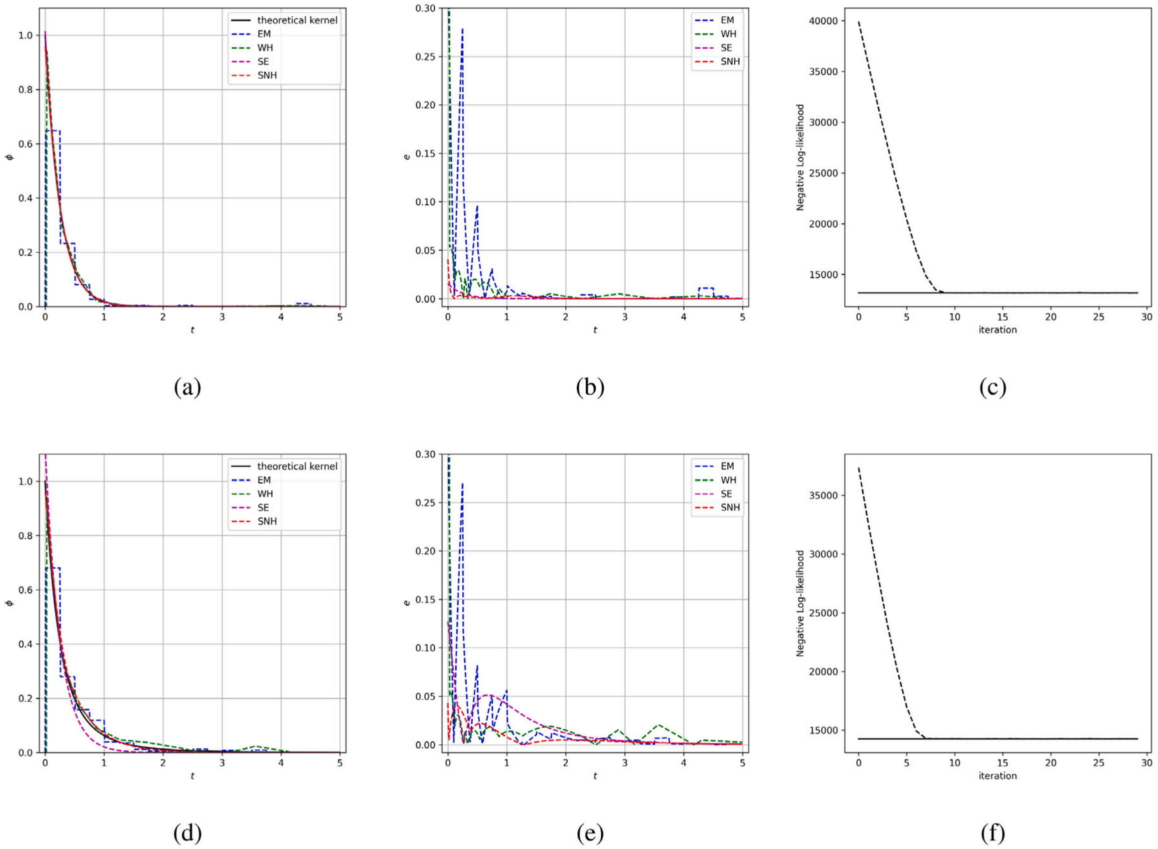
(a) (b)

Fig. 6. The QQ plots for the arrival time of (a) sell and (b) buy market orders for the BTC-USD pair. The prediction models are the homogeneous Poisson process, the exponential, the power-law, and the Shallow Neural Hawkes process.



(a) (b)

Fig. 7. The QQ plots for the arrival time for (a) sell and (b) buy market orders for the BTC-USD pair. The non-parametric prediction models used are the SNH, EM, and the WH method.



(a) (b) (c)

(d) (e) (f)

Fig. A.8. Synthetic data experiment results for univariate Hawkes processes.
(a) The estimated Exponential kernel. (b) The estimated error for Exponential kernel. (c) The negative log-likelihood values for the Exponential kernel estimation in the SNH model for increasing number epochs. (d) The estimated Power Law kernel. (e) The estimated error for the Power Law kernel. (f) The convergence plot of negative log-likelihood for the Power Law kernel estimation in the SNH model.

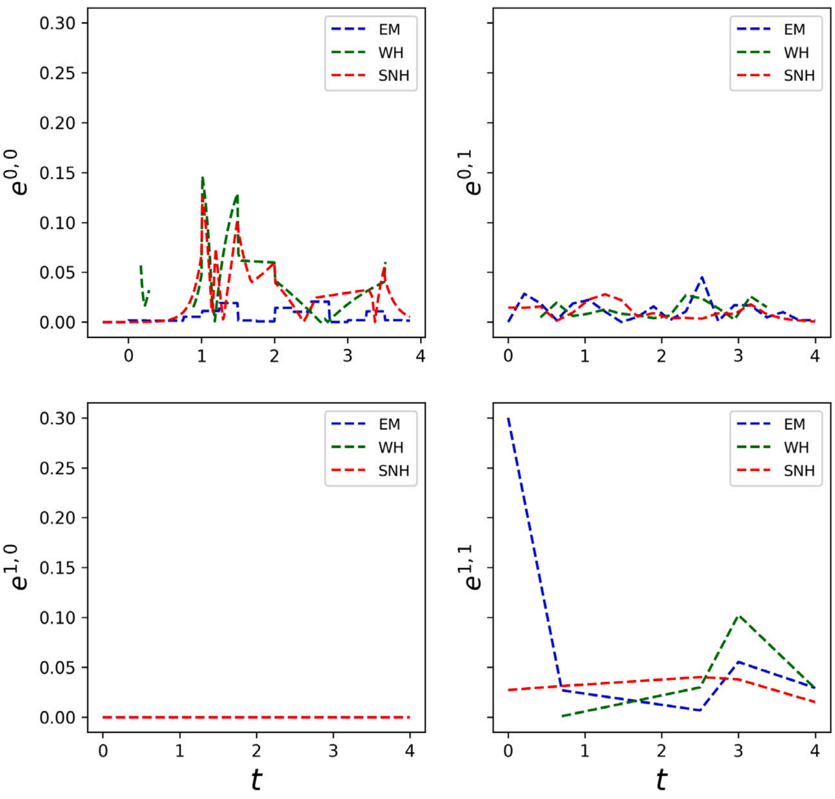


Fig. A.9. Synthetic data experiment results for bivariate Hawkes processes with the non-standard kernels: the estimated error for the kernels.

excitation kernel as a feed-forward network with a single hidden layer. We arrive at a specific network architecture to ensure that we can efficiently determine the optimal parameters using the SGD and that the kernels are excitation kernels. The excitation kernel then translates to an exponential of the sum of the ReLU functions. The network parameters are obtained using a batch SGD with log-likelihood as the objective to maximise. We provide an unbiased estimator for the gradient of the log-likelihood function required for the efficient application of the SGD. The method is tested with both synthetic and real dataset. The real dataset consists of tick-by-tick buy and sell market orders for BTC-USD pairs traded on the Binance cryptocurrency exchange. The performance of our method is consistently compared with the best in all the examples considered.

Interestingly, the cryptocurrency exchange's buy and sell trade order arrival process has a self-excitation feature. The observed self-excitation kernel does not follow the commonly used exponential or power-law based parametric kernels. In our dataset, we observed no cross-excitation effects between the buy and the sell orders, i.e. arrival of a buy order would not affect the intensity of arrival of a sell order (or vice-a-versa).

A future extension to our approach would be incorporating time-varying base intensities into the model. The desired extension would also include kernels that capture the negative dependencies of market events, i.e. inhibitive effects. Overall, the results presented here encourage further research on using neural networks in self-exciting point processes for modelling and making efficient statistical inferences of trading events.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Additional results

Fig. A.8 compares the kernel fits obtained using the sum of the exponential, the EM and the WH method with the SNH method for the exponential and the power-law Hawkes process. While all the methods can accurately recover the true kernel shapes, the most significant error is observed for the EM method. While the sum-of-exponential can accurately estimate the kernel for the exponential Hawkes process, it does not perform equally well for the power-law case. The SNH method achieves the best fit within ten epochs for both cases.

Fig. A.9 reports the L1 error, $|\phi_{dj}(t) - \widehat{\phi_{dj}}(t)|$, in the recovery of the bivariate Hawkes kernels for the case considered in Section 5.1.2.

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, 1999, pp. 471–507.

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

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

[5] 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.

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

[7] 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.

[8] E. Errais, Pricing insurance premia: a top down approach, Ann. Oper. Res. (2019) 1–16.

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

[10] 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.

[11] E. Lewis, G. Mohler, P.J. Brantingham, A.L. Bertozzi, Self-exciting point process models of civilian deaths in Iraq, Secur. J. 25 (3) (2012) 244–264.

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

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

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

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

[16] J. Zhuang, Y. Ogata, D. Vere-Jones, Stochastic declustering of space-time earthquake occurrences, J. Amer. Statist. Assoc. 97 (458) (2002) 369–380.

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

[18] 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.

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

[20] F. Salehi, W. Trouleau, M. Grossglauser, P. Thiran, Learning Hawkes Processes from a handful of events, in: Advances in Neural Information Processing Systems, 2019, pp. 12694–12704.

[21] 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.

[22] H. Mei, J.M. Eisner, The neural hawkes process: A neurally self-modulating multivariate point process, in: Advances in Neural Information Processing Systems, 2017, pp. 6754–6764.

[23] D. Koutmos, Market risk and Bitcoin returns, Ann. Oper. Res. (2019) 1–25.

[24] P. Giudici, G. Polinesi, Crypto price discovery through correlation networks, Ann. Oper. Res. (2019) 1–15.

[25] E. Akyildirim, A. Goncu, A. Sensoy, Prediction of cryptocurrency returns using machine learning, Ann. Oper. Res. (2020) 1–34.

[26] J.W. Goodell, S. Goutte, Co-movement of COVID-19 and Bitcoin: Evidence from wavelet coherence analysis, Finance Res. Lett. (2020) 101625.

[27] D. Philippas, H. Rjiba, K. Guesmi, S. Goutte, Media attention and Bitcoin prices, Finance Res. Lett. 30 (2019) 37–43.

[28] C.G. Bowsher, Modelling security market events in continuous time: Intensity based, multivariate point process models, J. Econometrics 141 (2) (2007) 876–912.

[29] E. Bacry, S. Delattre, M. Hoffmann, J.-F. Muzy, Modelling microstructure noise with mutually exciting point processes, Quant. Finance 13 (1) (2013) 65–77.

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

[31] D.J. Daley, D. Vere-Jones, An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure, Springer Science & Business Media, 2007.

[32] 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.

[33] E. Bacry, M. Bompaire, S. Gaïffas, S. Poulsen, Tick: a Python library for statistical learning, with a particular emphasis on time-dependent modelling, 2017, arXiv preprint arXiv:1707.03003.

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

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

[36] 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.

[37] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[38] P. Morzywolek, Non-Parametric Methods for Estimation of Hawkes Process for High-Frequency Financial Data (ETH Zürich Master thesis), 2015.

[39] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, Vol. 1, No. 10, Springer Series in Statistics New York, 2001.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

Sobin Joseph is currently pursuing his Ph.D. degree at the Department of Management Studies, Indian Institute of Science, Bangalore. Sobin holds a Master's degree in Industrial engineering from the College of Engineering, Trivandrum. His current research is focused on the area of application of point processes in finance.

Lekhapriya Kashyap is currently pursuing her Ph.D. degree at Texas A&M University in data science. Lekhapriya has a Master's degree in Industrial Management from the New Jersey Institute of Technology. Lekhapriya performed the current study at the Department of Management Studies, Indian Institute of Science, Bangalore.

Shashi Jain is an Assistant Professor at the Department of Management Studies, Indian Institute of Science, Bangalore. He has a Ph.D. in Applied Mathematics from the TU Delft, Netherlands. His research is focused on the area of computational finance.