Comparative Analysis among NHP,THP and Extended Cox-type Hazards Model for Recurrent Data

A Project Report Submitted to the
Department of Computer Science of
Ramakrishna Mission Vivekananda Educational and Research Institute, Belur,
in partial fulfilment of the requirements for the degree of
MSc in Big Data Analytics.

Submitted by
UDDALAK MUKHERJEE
ID No. B2330042

Supervisor:
Dr. Sudipta Das
Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute

Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, Howrah 711202, West Bengal, India
December 26, 2025

Comparative Analysis among NHP,THP and Extended Cox-type Hazards Model for Recurrent Data

By

UDDALAK MUKHERJEE

Declaration by student:

"I hereby declare that the present dissertation is the outcome of my project work under the guidance of Dr. Sudipta Das and I have properly acknowledged the sources of materials used in my project report."

_____

(Uddalak Mukherjee, ID No. B2330042)

A project report in the partial fulfilment of the requirements of the degree of MSc in Big Data Analytics

Examined and approved on

_____

by

_____

Dr. Sudipta Das(supervisor)
Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute

Countersigned by

_____

Registrar
Ramakrishna Mission Vivekananda Educational and Research Institute

Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, Howrah 711202, West Bengal, India

December 26, 2025

Acknowledgement

The present project work is submitted in partial fulfilment of the requirements for the degree of Master of Science of Ramakrishna Mission Vivekananda University (RKMVU). I express my deepest gratitude to my supervisor Prof. Dr. Sudipta Das of Ramakrishna Mission Vivekananda Educational and Research Institute for his inestimable support, encouragement, profound knowledge, largely helpful conversations and also for providing me a systematic way for the completion of my project work. His ability to work hard inspired me a lot. I am also extremely grateful to the Vice-Chancellor of this University for his encouragement and support throughout the course. Last but not the least, this work would not have been possible without support of my fellow classmates.

Belur
December 26, 2025

(Uddalak Mukherjee)
Department of Computer Science
Ramakrishna Mission Vivekananda Educational and Research Institute

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modeling recurrent events is pervasive across domains such as healthcare, finance, and system monitoring. Traditional approaches primarily rely on stochastic process models to characterize and predict the temporal occurrence of events, typically using statistical estimation techniques such as Maximum Likelihood Estimation (MLE). Among the earliest and most fundamental of these models is the Homogeneous Poisson Process (HPP), which assumes a constant event intensity over time, an assumption that often oversimplifies real-world dynamics. Other models include the Non-homogeneous Poisson Process (NHPP) and Renewal process which model the event rate as a deterministic function of time. While these methods offer valuable insights, they often rely on restrictive assumptions like a non-random, time-parametrized intensity function. Hawkes et al.[2] introduced the Hawkes process, a family of doubly stochastic models for recurrent event data, with two variants: the self-exciting and self-correcting process. The self-exciting Hawkes process gained prominence for its ability to model events where past occurrences of certain events increase the likelihood of future events. Subsequent works, such as [13, 5, 12], build upon this idea by leveraging neural networks to model the intensity function, while preserving key structural properties of the Hawkes process framework. These models are often referred to as the Neural Temporal Point Process (NTPP) models. In a parallel line of research, the extended Cox-type hazards model for recurrent events [4] adapts the classical Cox-proportional hazards model to predict failure times of recurrent events in Survival Analysis and Reliability, with or without censoring. In this research report, we make three notable contributions: (a) We introduce a Cox-type hazard model for recurrent events that closely mirrors the Hawkes framework (b) We propose a covariate-dependent variant of the NTPP model - the Transformer Hawkes Process (THP) [13] (c) We conduct an extensive empirical comparison of NTPP models with the extended Cox model on a highly informative yet underexplored dataset.

Chapter 2

Literature Survey

Modern literature has made significant strides in enhancing the classic self-exciting Hawkes process by integrating it with neural network frameworks. These advancements enable the modeling of more intricate recurrent event data, incorporating temporal dependencies that are otherwise difficult to capture with traditional statistical approaches. Notable implementations in this domain include the Recurrently Marked Temporal Point Process (RMTPP) [1], the Neural Hawkes Process (NHP) [5], the Self-Attentive Hawkes Process (SAHP) [12], and the Transformer Hawkes Process (THP) [13]. These methods address the fundamental limitations of the classic Hawkes process, particularly its inability to effectively handle large-scale datasets and complex event patterns. By leveraging the representational power of neural networks, these models provide improved scalability, enabling them to process datasets with high dimensionality and varied structures. Moreover, their capacity to model intricate temporal dependencies has positioned these approaches as indispensable tools in fields such as finance, healthcare, and social network analysis, where recurrent events often exhibit non-trivial correlations over time. By embedding neural architectures, these models offer a more consistent and robust framework, making them suitable for applications demanding sophisticated temporal dynamics modeling.

Kalbfleisch and Prentice [4] (Chapter 9), provides foundational knowledge on survival analysis and discusses extensions to the Cox proportional hazards model. The extended Cox-type hazards model, allows for the analysis, estimation, and prediction of recurrent event data under specific restrictions and assumptions. Despite its utility, traditional implementations of this model often fall short in capturing the nuances of recurrent event sequences, particularly when temporal dependencies between events or interactions with covariates become complex. Building upon this framework, we propose a novel approach to model recurrent event data by integrating dynamic covariates and incorporating a flexible baseline hazard function. This enhance-

ment enables our model to address the limitations of the extended Cox model, offering greater flexibility in modeling diverse recurrent event scenarios.

To validate the proposed model, we applied it to a informative dataset introduced by [7]. This dataset comprises profile vectors for over 10,000 pipes located in the Netherlands, accompanied by a detailed failure record for each pipe. The failure records represent the recurrent event history, with each failure characterized by its timestamp and failure type. The dataset provides an ideal testbed to evaluate models for recurrent event prediction, given its richness in covariate information and the availability of longitudinal event data. We compared the performance of our model with two state-of-the-art neural implementations: the Neural Hawkes Process (NHP) implemented by [11] and our custom implementation of the Transformer Hawkes Process (THP). Both NHP and THP are designed to capture the temporal dependencies in event data, but they differ in their architectural foundations and mechanisms for modeling temporal dynamics. While NHP employs recurrent neural network-based embeddings to model event sequences, THP leverages the self-attention mechanism of transformers, providing superior scalability and robustness to long-term dependencies.

Our experimental results demonstrate that the extended Cox-type model, despite its reliance on simpler assumptions, can achieve competitive performance when appropriately adapted to incorporate covariate effects and recurrent event structures. The comparisons with NHP and THP highlight the strengths and weaknesses of each approach, providing insights into the trade-offs between model complexity, computational efficiency, and predictive accuracy. This study underscores the potential of hybrid approaches that blend classical statistical methods with modern machine learning frameworks to address the challenges of recurrent event prediction. The findings open avenues for further exploration, particularly in enhancing the Cox model to better align with neural architectures and leveraging its interpretability alongside the flexibility of neural methods.

Chapter 3

Self-Exciting Linear Hawkes Process

Problems of estimation, filtering and smoothing of point process have been discussed by many authors (Vere-Jones [10], Snyder [9], Segal [8], Ozaki [6]). However, it was observed by Vere-Jones in [10] that no satisfactory solution for the parameter estimation problem has emerged. In 1979, [6] proposed the maximum likelihood estimation technique for the general hawkes process. Our objective in this chapter is to re-produce the results of [6] for (a) A simpler version of the hawkes model and (b) A re-parameterized version of the hawkes model discussed in the paper. This chapter gives key insights for the classic Hawkes self-exciting model which is essentially for getting a grasp on subsequent models which are more complex, like THP and NHP. We present a slightly toned down version of the analysis and simulation experiments presented in [2, 3] to address a more general audience.

Let $N(t)$ be a point process such that,

$$P\{N(t + \Delta t) - N(t) = 1\} = \lambda(t)\Delta t + o(\Delta t)$$
$$P\{N(t + \Delta t) - N(t) > 1\} = o(\Delta t)$$

Hawkes [3], [2] introduced a general point process model whose intensity function is given by

$$\lambda(t) = \lambda_0 + \int_{-\infty}^{t} g(t - u)dN(u)$$

where $g(\cdot) \geq 0$ and $\int_0^\infty g(u)du < 1$. We call $g(t)$ the response function. Alternatively, the model can also be stated as,

$$\lambda(t) = \lambda_0 + \sum_{j:t_j < t} \psi(t - t_j)$$

where λ_0 is called the baseline intensity and $\psi(\cdot)$ is a pre-specified decaying function.

In this chapter, we will be dealing with only the exponentially decaying function with two different representations for the intensity model:

- $\lambda(t) = \lambda_0 + \sum_{t_i < t} \exp\left(-(t_i - t)\right)$

- $\lambda(t) = \lambda_0 + \alpha \sum_{t_i < t} \exp\left(-(t_i - t)/\sigma\right)$

where $\lambda_0, \alpha, \sigma$ are all positive quantities.

3.1 Likelihood Estimation

In this section, we construct the conditional likelihood function for both models based on their respective intensities.

3.1.1 MLE of Simple Linear Hawkes Process

Lemma 1 (MLE for the baseline intensity of a simple linear Hawkes process)**.** *Consider a univariate linear Hawkes process with conditional intensity*

$$\lambda(t) = \lambda_0 + \sum_{t_i < t} \exp(-(t - t_i)),$$

and let $0 < t_1 < \cdots < t_k$ denote the observed event times on the interval $[0, t_k]$. The maximum likelihood estimator $\hat{\lambda}_0$ of the baseline intensity satisfies the estimating equation

$$\sum_{j=1}^{k} \frac{1}{\lambda_0 + \sum_{i=0}^{j-1} \exp(-(t_j - t_i))} = t_k.$$

Moreover, the above equation admits a unique solution $\hat{\lambda}_0 > 0$.

Proof. Given the intensity function, we can construct the density function in the following manner:

$$\lambda(t) = \lambda_0 + \sum_{t_i < t} \exp\left(-(t_i - t)\right)$$

Or,

$$\lambda(t \mid t_0, \ldots, t_k, \lambda_0) = \lambda_0 + \sum_{i=0}^{k} \exp\left(-(t - t_i)\right).$$

Hence,

$$f\left(t \mid t_0, \ldots, t_k, \lambda_0\right) = \left[\lambda_0 + \sum_{i=0}^{k} \exp\left(-\left(t - t_i\right)\right)\right] \times$$

$$\exp\left\{-\int_{t_k}^{t} \lambda_0 d\tau - \int_{t_k}^{t} \sum_{t_i < \tau} \exp\left(-\left(\tau - t_i\right)\right) d\tau\right\}.$$

Hence, the conditional likelihood function is,

$$\mathcal{L}\left(t_1, \ldots, t_k \mid t_0, \lambda_0\right)$$

$$= f\left(t_k \mid t_0, \ldots, t_{k-1}, \lambda_0\right) \times f\left(t_{k-1} \mid t_0, \ldots, t_{k-2}, \lambda_0\right) \times \cdots \times f\left(t_1 \mid t_0, \lambda_0\right)$$

$$= \prod_{j=1}^{k} \left[\lambda_0 + \sum_{i=0}^{j-1} \exp\left(-\left(t_j - t_i\right)\right)\right] \quad \times$$

$$\prod_{j=1}^{k} \exp\left[-\int_{t_{j-1}}^{t_j} \lambda_0 \, d\tau - \int_{t_{j-1}}^{t_j} \sum_{t_i < \tau} \exp\left(-\left(\tau - t_i\right)\right) d\tau\right]$$

Therefore, the conditional log likelihood function is,

$$l\left(t_1, \ldots, t_k \mid t_0, \lambda_0\right) = \sum_{j=1}^{k} \log\left[\lambda_0 + \sum_{i=0}^{j-1} \exp\left(-\left(t_j - t_i\right)\right)\right] - \lambda_0 \sum_{j=1}^{k} \left(t_j - t_{j-1}\right)$$

$$- \sum_{j=1}^{k} \left[\sum_{i=0}^{j-2} \exp\left(-\left(t_{j-1} - t_i\right)\right) - \sum_{i=0}^{j-1} \exp\left(-\left(t_j - t_i\right)\right)\right]$$

$$= \sum_{j=1}^{k} \log\left[\lambda_0 + \sum_{i=0}^{j-1} \exp\left(-\left(t_j - t_i\right)\right)\right] - \lambda_0 \left(t_k - t_0\right) - \sum_{j=1}^{k-1} \exp\left(-\left(t_k - t_i\right)\right)$$

$$= \sum_{j=1}^{k} \log\left[\lambda_0 + \sum_{i=1}^{j-1} \exp\left(-\left(t_j - t_i\right)\right)\right] - \lambda_0 t_k - \sum_{j=1}^{k-1} \exp\left(-\left(t_k - t_j\right)\right)$$

Hence, the final log-likelihood is given as,

$$l\left(\lambda_0\right) = \sum_{j=1}^{k} \log\left[\lambda_0 + \sum_{i=0}^{j-1} \exp\left(-\left(t_j - t_i\right)\right)\right] - \lambda_0 t_k - \sum_{j=1}^{k-1} \exp\left(-\left(t_k - t_j\right)\right) \qquad (1.1)$$

On differentiating (1.1) with respect to λ_0 we get the following equation,

$$\sum_{j=1}^{k} \frac{1}{\lambda_0 + \sum_{i=0}^{j-1} \exp\left(-\left(t_j - t_i\right)\right)} = t_k \qquad (1.2)$$

13

Solving (1.2) for λ_0 will give us $\hat{\lambda}_0$, the maximum likelihood estimate of the baseline intensity.

$\square$

3.1.2 MLE of General Linear Hawkes Process

Lemma 2 (MLE for the parameters of a generalized linear Hawkes process). *Consider a univariate Hawkes process with conditional intensity*

$$\lambda(t) = \lambda_0 + \alpha \sum_{t_i < t} \exp\left(-\frac{t - t_i}{\sigma}\right),$$

and let $0 < t_1 < \cdots < t_k$ denote the observed event times on $[0, t_k]$. The maximum likelihood estimators $(\hat{\lambda}_0, \hat{\alpha}, \hat{\sigma})$ satisfy the following system of estimating equations:

$$\sum_{j=1}^{k} \frac{1}{\lambda_0 + \alpha \sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_i}{\sigma}\right)} = t_k,$$

$$\sum_{j=1}^{k} \frac{\sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_i}{\sigma}\right)}{\lambda_0 + \alpha \sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_i}{\sigma}\right)} = \sigma \sum_{j=1}^{k-1} \exp\left(-\frac{t_k - t_j}{\sigma}\right) - \sigma,$$

$$\sum_{j=1}^{k} \frac{\alpha \sum_{i=0}^{j-1}(t_j - t_i) \exp\left(-\frac{t_j - t_i}{\sigma}\right)}{\lambda_0 + \alpha \sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_i}{\sigma}\right)} = \alpha \sigma^2 \left[\sum_{j=1}^{k-1} \exp\left(-\frac{t_k - t_j}{\sigma}\right) - 1\right]$$
$$+ \alpha \sigma \sum_{j=1}^{k-1}(t_k - t_j) \exp\left(-\frac{t_k - t_j}{\sigma}\right).$$

Any solution to this system corresponds to a stationary point of the log-likelihood.

Proof. Analogous to the previous lemma, we can construct the conditional likelihood of the process from its intensity function as follows:

$$\lambda(t) = \lambda_0 + \alpha \sum_{t_i < t} \exp\left(-\frac{t - t_i}{\sigma}\right)$$

Or,

$$\lambda(t \mid t_0, \ldots, t_k, \Theta) = \lambda_0 + \alpha \sum_{i=0}^{k} \exp\left(-\frac{t - t_i}{\sigma}\right)$$

14

Where Θ is the parameter set containing $\lambda_0, \alpha, \sigma$. Hence,

$$f\left(t \mid t_0, \ldots, t_k, \Theta\right) = \left[\lambda_0 + \alpha \sum_{i=0}^{k} \exp\left(-\frac{t - t_i}{\sigma}\right)\right] \times$$

$$\exp\left\{-\int_{t_k}^{t} \lambda_0 d\tau - \alpha \int_{t_k}^{t} \sum_{t_i < \tau} \exp\left(-\frac{\tau - t_i}{\sigma}\right) d\tau\right\}.$$

Therefore, the conditional likelihood function will be,

$$\mathcal{L}\left(t_1, \ldots, t_k \mid t_0, \Theta\right) = \prod_{j=1}^{k} \left[\lambda_0 + \alpha \sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_i}{\sigma}\right)\right]$$

$$\times \prod_{j=1}^{k} \exp\left[-\int_{t_{j-1}}^{t_j} \lambda_0 d\tau - \alpha \int_{t_{j-1}}^{t_j} \sum_{t_i < \tau} \exp\left(-\frac{\tau - t_i}{\sigma}\right) d\tau\right]$$

And finally taking log and simplifying this just like in the previous section the final log-likelihood function will be,

$$l\left(\Theta\right) = \sum_{j=1}^{k} \log\left[\lambda_0 + \alpha \sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_i}{\sigma}\right)\right] - \lambda_0 t_k + \alpha\sigma \sum_{j=1}^{k-1} \exp\left(-\frac{t_k - t_j}{\sigma}\right) - \alpha\sigma \quad (2.1)$$

Differentiating (2.1) with respect to λ_0 we get the following equation,

$$\sum_{j=1}^{k} \frac{1}{\lambda_0 + \alpha \sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_j}{\sigma}\right)} = t_k \quad (2.2)$$

Again differentiation (2.1) with respect to α we get,

$$\sum_{j=1}^{k} \frac{\sum_{i=0}^{j-1} \exp\left(-\frac{t_i - t_j}{\sigma}\right)}{\lambda_0 + \alpha \sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_j}{\sigma}\right)} = \sigma \sum_{j=1}^{k-1} \exp\left(-\frac{t_k - t_j}{\sigma}\right) - \sigma \quad (2.3)$$

And finally differentiating (2.1) with respect to σ we get,

$$\sum_{j=1}^{k} \frac{\alpha \sum_{i=0}^{j-1} (t_j - t_i) \exp\left(-\frac{t_j - t_i}{\sigma}\right)}{\lambda_0 + \alpha \sum_{i=0}^{j-1} \exp\left(-\frac{t_j - t_i}{\sigma}\right)} = \alpha\sigma^2 \left[\sum_{j=1}^{k-1} \exp\left(-\frac{t_k - t_j}{\sigma}\right) - 1\right]$$

$$+ \alpha\sigma \sum_{j=1}^{k-1} (t_k - t_j) \exp\left(-\frac{t_k - t_j}{\sigma}\right) \quad (2.4)$$

Solving equations (2.2), (2.3) and (2.4) simultaneously for $\lambda_0, \alpha, \sigma$ we can get the maximum likelihood estimates $\hat{\lambda}_0, \hat{\alpha}, \hat{\sigma}$ of the concerned parameters. □

3.2 Simulation and Empirical Evaluation

In this section, we discuss a technique for simulating samples from a general linear Hawkes process and then apply the methods from the previous section to find the maximum likelihood estimates of the parameters as well as their variances.

3.2.1 Simulating from a General Hawkes Process

The algorithm for the generation of n-samples from Hawkes' self-exciting process data is thus described as follows:

Algorithm 1 Simulation Algorithm

1: Sample $U \sim \text{Uniform}(0,1)$
2: Set $t_1 \leftarrow -\log(U)/\lambda_0$
3: Initialize $S(1) \leftarrow 1$, $k \leftarrow 1$
4: **while** $k < n$ **do**
5:     Sample $U \sim \text{Uniform}(0,1)$
6:     Compute u as the solution of

$$\log U + \lambda_0(u - t_k) + \alpha\sigma S(k)\left(1 - \exp\left(-\frac{u - t_k}{\sigma}\right)\right) = 0$$

7:     Set $t_{k+1} \leftarrow u$
8:     Update

$$S(k+1) \leftarrow \exp\left(-\frac{t_{k+1} - t_k}{\sigma}\right) S(k) + 1$$

9:     $k \leftarrow k + 1$
10: **end while**

Lemma 3 (Correctness of Algorithm 1). *Algorithm 1 generates samples from the General Linear Hawkes Process.*

Proof. Suppose that data $t_1, t_2, \ldots, t_k$ are given, Let $F(t \mid t_1, \ldots t_k, \theta)$ be conditional distribution of random variable of interval between t_k and the next event t $(t \geq t_k)$ of the process, and let $f(t \mid t_1, \ldots, t_k, \theta)$ be its probability density function. The conditional intensity function is given by,

$$\lambda(t \mid t_1, \ldots, t_k, \theta) = \frac{f(t \mid t_1, \ldots, t_k, \theta)}{1 - F(t \mid t_1, \ldots, t_k, \theta)}$$

And we have,

$$\log\{1 - F(u \mid t, \dots, t_k, \theta)\} = -\int_{t_k}^{u} \lambda\left(t \mid t_1, \dots, t_k, \theta\right) dt$$

$$= -\int_{t_k}^{u} \lambda_0 + \alpha \sum_{i=1}^{k} \exp\left(-\frac{t - t_i}{\sigma}\right) dt$$

Assuming, $1 - F\left(u \mid t_1, \dots, t_k, \theta\right)$ is distributed uniformly on $[0, 1]$, we generate t_{k+1} by first generating a uniform random number U in $[0, 1]$ & solving the following equation with respect to u,

$$\log U + \int_{t_k}^{u} \left\{\lambda_0 + \alpha \sum_{i=1}^{k} \exp\left(-\frac{t - t_i}{\sigma}\right)\right\} dt = 0$$

$$\Rightarrow \log U + \lambda_0\left(u - t_k\right) + \alpha\sigma\left\{\sum_{i=1}^{k} \exp\left(-\frac{t_k - t_i}{\sigma}\right) - \sum_{i=1}^{k} \exp\left(-\frac{u - t_i}{\sigma}\right)\right\} = 0$$

This equation can be solved recursively as,

$$\log U + \lambda_0\left(u - t_k\right) + \alpha\sigma S(k)\left\{1 - \exp\left(-\frac{u - t_k}{\sigma}\right)\right\} = 0 \tag{3.1}$$

Where,

$$S(1) = 1,$$
$$S(i) = \exp\left(-\frac{t_i - t_{i-1}}{\sigma}\right) S(i-1) + 1 \quad (i \geq 2)$$

$\square$

3.2.2 Estimation for Simple Linear Hawkes Process

One hundred samples were generated from the general Hawkes process using the algorithm discussed above for three different values of the baseline intensity λ_0, with α and σ set to 1. The maximum likelihood estimate $\hat{\lambda}_0$ of λ_0 was calculated by numerically maximizing equation (1.1). Alternatively, it could also be calculated by numerically solving equation (1.2). A total of 5000 iterations were performed, each with varying samples generated from the same baseline intensity, and the arithmetic mean of $\hat{\lambda}_0$ was taken as the final estimate. The corresponding standard error and relative bias were also calculated. This process was repeated for all three different values of baseline intensity, with sample sizes of 500 and 1000. The table below highlights the results.

Table 3.1: Estimated paramter for Simple Hawkes Process

λ_0	k	$\overline{\hat{\lambda}_0}$	$\overline{\text{se}(\hat{\lambda}_0)}$	Relative Bias
	100	0.7166	0.4639	0.4332
0.5	500	0.6780	0.3776	0.3561
	1000	0.6636	0.3418	0.3271
	100	1.2700	0.6554	0.2700
1	500	1.2305	0.5439	0.2305
	1000	1.2082	0.5242	0.2082
	100	2.3448	1.0025	0.1724
2	500	2.2710	0.8218	0.1355
	1000	2.2448	0.7818	0.1224

Table 3.1 demonstrates that, for a specified baseline intensity, the estimate approaches the actual value with increasing numbers of simulated samples used in the estimation process. Similarly, for a fixed sample size, the relative bias (defined as $\frac{\hat{\theta}-\theta}{\theta}$, where $\hat{\theta}$ is the estimate and θ is the actual value) diminishes as the baseline intensity increases. This indicates that, within a fixed time interval, a higher number of arrivals leads to a more accurate estimate in terms of reduced bias.

3.2.3 Estimation for General Linear Hawkes Process

One hundred samples were generated from the general Hawkes process using the algorithm discussed above for two different combinations of the parameters α, σ and λ_0. The maximum likelihood estimates $\hat{\alpha}$, $\hat{\sigma}$, and $\hat{\lambda}_0$ were calculated by numerically maximizing the log-likelihood given by equation (2.1). Alternatively, these estimates could be obtained by solving equations (2.2), (2.3), and (2.4) simultaneously. This process was repeated over 5000 iterations with varying samples generated from the same parameter combinations, and the arithmetic means of $\hat{\alpha}$, $\hat{\sigma}$, and $\hat{\lambda}_0$ were taken as the final estimates. Their corresponding standard errors and relative biases were also calculated. The same procedure was conducted for larger sample sizes of 500 and 1000. The generalized variance (denoted using $\det(\hat{\Sigma})$ - determinant of the covariance matrix) and the condition number (denoted with $\kappa(\hat{\Sigma})$ - ratio of the maximum and minimum eigen values of the covariance matrix) were computed in the order of (α, σ, λ_0) for both rows and columns for each entry. The following table demonstrates the results which were found. Here Θ is the parameter set containing (α, σ, λ_0) and **k** is the number of samples.

The Table 3.2 demonstrates that increasing the number of samples in the general Hawkes process leads to estimates that more closely align with the true parameter values for both pa-

Table 3.2: Estimated Parameters for General Hawkes Process

Θ	k	$\bar{\hat{\alpha}}$	$\bar{\hat{\sigma}}$	$\bar{\hat{\lambda}}_0$	$\mathbf{det}(\hat{\Sigma})$	$\kappa(\hat{\Sigma})$
$(\alpha, \sigma, \lambda_0)$ $= (4, 0.2, 0.5)$	100	4.0666	0.2062	0.5764	1.84×10^{-3}	111.7387
	500	4.0172	0.2005	0.5109	6.18×10^{-8}	2744.2989
	1000	4.0025	0.2005	0.5054	7.23×10^{-9}	2820.1885
$(\alpha, \sigma, \lambda_0)$ $= (0.8, 1.0, 0.5)$	100	0.7834	1.0963	0.7091	1.35×10^{-2}	4.5004
	500	0.8011	1.0042	0.5458	1.54×10^{-6}	22.6789
	1000	0.8021	0.9989	0.5221	1.15×10^{-7}	28.5099

rameter sets $(\alpha, \sigma, \lambda_0) = (4, 0.2, 0.5)$ and $(\alpha, \sigma, \lambda_0) = (0.8, 1.0, 0.5)$. Specifically, estimates of $\hat{\alpha}$, $\hat{\sigma}$, and $\hat{\lambda}_0$ become more accurate as the sample size increases from 100 to 1000. The generalized variances indicate that the variance of the estimates decreases with larger sample sizes, reflecting greater precision, while the increasing condition number with sample size indicates stronger dependence among the estimates, although the first set is highly ill-conditioned as compared to the second set.

The Table 3.3 demonstrates that for a fixed value of baseline intensity and sample size, the relative biases of both $\bar{\hat{\alpha}}$ and $\bar{\hat{\sigma}}$ tend to decrease as the value of the original parameter increases. This indicates that the accuracy of the parameter estimates improves with lower values of the original parameters under these conditions. True to the nature of the intensity function of the general hawkes model, if both α and σ are low then only the contribution of the baseline intensity, λ_0 is significant to the overall intensity at a particular time point. As the value of both the former parameter increases their contribution becomes increasingly significant.

Table 3.3: Relative Biases of Parameters for General Hawkes Process

Θ	k	$\mathbf{RB}(\bar{\hat{\alpha}})$	$\mathbf{RB}(\bar{\hat{\sigma}})$	$\mathbf{RB}(\bar{\hat{\lambda}}_0)$
$(\alpha, \sigma, \lambda_0)$ $= (4, 0.2, 0.5)$	100	0.01665	0.0310	0.1528
	500	0.0043	0.0025	0.0218
	1000	0.000625	0.0025	0.0108
$(\alpha, \sigma, \lambda_0)$ $= (0.8, 1.0, 0.5)$	100	-0.02075	0.0963	0.4182
	500	0.001375	0.0042	0.0916
	1000	0.002625	-0.0011	0.0442

Chapter 4

Cox-Proportional Hazards Model

In this chapter we explore in detail the premise for the classic Cox-proportional hazards model as described by [4]. Unlike in Chapter 3, we did not explicitly show the estimation procedure for the Cox model since it is a standard procedure covered in detail in a number of text books and covering all details would far exceed the purpose of this report. We also did not conduct any empirical evaluation of the Cox model for the same purpose, as multiple experiments and problems have already been provided in [4].

Suppose we observe n independent subjects. For subject i let

$$T_i = \text{true event time}, \qquad C_i = \text{censoring time},$$

and we observe

$$t_i = \min(T_i, C_i), \qquad \delta_i = \mathbf{1}\{T_i \leq C_i\},$$

and a p-vector of covariates $X_i \in \mathbb{R}^p$. Write the observed dataset as $\{(t_i, \delta_i, X_i) : i = 1, \ldots, n\}$.

Define the hazard function for the event time T given covariates X by

$$\lambda(t \mid X) = \lim_{\Delta t \downarrow 0} \frac{P(t \leq T < t + \Delta t \mid T \geq t, X)}{\Delta t}.$$

4.1 Model specification and likelihood construction

The Cox proportional hazards model assumes the hazard of subject i at time t has the form

$$\lambda(t \mid X_i) = \lambda_0(t) \exp(\beta^\top X_i), \tag{4.1}$$

where

- $\beta \in \mathbb{R}^p$ is an unknown regression vector of interest (log hazard ratios),

- $\lambda_0(t)$ is an unspecified (nonparametric) baseline hazard function common to all subjects.

Key implications:

$$\frac{\lambda(t \mid X_i)}{\lambda(t \mid X_j)} = \exp\{\beta^\top (X_i - X_j)\},$$

so covariate effects are multiplicative on the hazard and constant over time (proportional hazards).

4.1.1 Partial likelihood

A full likelihood would require specifying $\lambda_0(t)$. Cox proposed the *partial likelihood* that allows estimation of β without specifying $\lambda_0(t)$.

Lemma 4 (Cox partial log-likelihood)**.** *Consider the Cox proportional hazards model*

$$\lambda(t \mid X) = \lambda_0(t) \exp(\beta^\top X),$$

and assume independent censoring conditional on covariates. Let $(t_i, \delta_i, X_i)_{i=1}^n$ denote the observed data, and let $\mathcal{R}(t) = \{j : t_j \geq t\}$ be the risk set at time t. Then the partial log-likelihood for the regression parameter β is given by

$$\ell_P(\beta) = \sum_{i=1}^n \delta_i \left[\beta^\top X_i - \log \left(\sum_{j \in \mathcal{R}(t_i)} \exp(\beta^\top X_j) \right) \right].$$

Proof. Order distinct observed event times as $t_{(1)} < t_{(2)} < \cdots < t_{(m)}$ where events occur (ties handled below). For each event time $t_{(k)}$ denote the risk set

$$\mathcal{R}(t_{(k)}) = \{j : t_j \geq t_{(k)}\}$$

(i.e. subjects still at risk just before $t_{(k)}$). If the event at $t_{(k)}$ is a single subject i_k (no tie), the conditional probability that subject i_k fails at $t_{(k)}$ given one failure at that time and given the risk set is

$$\frac{\lambda_0(t_{(k)}) \exp(\beta^\top X_{i_k})}{\sum_{j \in \mathcal{R}(t_{(k)})} \lambda_0(t_{(k)}) \exp(\beta^\top X_j)} = \frac{\exp(\beta^\top X_{i_k})}{\sum_{j \in \mathcal{R}(t_{(k)})} \exp(\beta^\top X_j)}.$$

Multiplying these conditional probabilities over event times yields the (profiled) partial likeli-

hood for β:

$$L_P(\beta) \;=\; \prod_{k=1}^{m} \frac{\exp(\beta^\top X_{i_k})}{\sum_{j \in \mathcal{R}(t_{(k)})} \exp(\beta^\top X_j)}. \tag{4.2}$$

Equivalently, using indicator δ_i and the convention that the product runs over observed events,

$$L_P(\beta) = \prod_{i=1}^{n} \left[\frac{\exp(\beta^\top X_i)}{\sum_{j \in \mathcal{R}(t_i)} \exp(\beta^\top X_j)} \right]^{\delta_i}. \tag{4.3}$$

hence, the log-partial-likelihood is

$$\ell_P(\beta) = \sum_{i=1}^{n} \delta_i \left\{ \beta^\top X_i - \log\left(\sum_{j \in \mathcal{R}(t_i)} \exp\{\beta^\top X_j\} \right) \right\}. \tag{4.4}$$

$\square$

4.1.2 Score function and observed information

Lemma 5 (Score function and information matrix of the Cox partial likelihood). *Consider the Cox proportional hazards model*

$$\lambda(t \mid X) = \lambda_0(t) \exp(\beta^\top X),$$

and assume independent censoring conditional on covariates. Let $\ell_P(\beta)$ denote the partial log-likelihood defined in Lemma 4, and let $\mathcal{R}(t) = \{j : t_j \geq t\}$ be the risk set at time t. Then the score function and the observed information matrix are given by

$$U(\beta) \;=\; \sum_{i=1}^{n} \delta_i \left[X_i - \bar{X}(t_i; \beta) \right], \qquad\qquad I(\beta) = \sum_{i=1}^{n} \delta_i \operatorname{Var}_\beta \big(X \mid \mathcal{R}(t_i) \big).$$

where $\bar{X}(t; \beta) = \dfrac{\sum_{j \in \mathcal{R}(t)} X_j e^{\beta^\top X_j}}{\sum_{j \in \mathcal{R}(t)} e^{\beta^\top X_j}}.$

Proof. Differentiate $\ell_P(\beta)$ to obtain the score vector $U(\beta)$ and observed information $I(\beta)$.

Define the conditional (at time t) risk-weighted mean of covariates:

$$\bar{X}(t; \beta) \;=\; \frac{\sum_{j \in \mathcal{R}(t)} X_j e^{\beta^\top X_j}}{\sum_{j \in \mathcal{R}(t)} e^{\beta^\top X_j}}.$$

Then the score is

$$U(\beta) = \frac{\partial \ell_P(\beta)}{\partial \beta} = \sum_{i=1}^{n} \delta_i \{X_i - \bar{X}(t_i; \beta)\}. \tag{4.5}$$

And the observed (negative Hessian) information matrix is

$$I(\beta) = -\frac{\partial^2 \ell_P(\beta)}{\partial \beta \, \partial \beta^\top} = \sum_{i=1}^{n} \delta_i \left\{ \frac{\sum_{j \in \mathcal{R}(t_i)} X_j X_j^\top e^{\beta^\top X_j}}{\sum_{j \in \mathcal{R}(t_i)} e^{\beta^\top X_j}} - \bar{X}(t_i; \beta) \bar{X}(t_i; \beta)^\top \right\}$$

$$= \sum_{i=1}^{n} \delta_i \, \mathrm{Var}_\beta \big(X \mid \mathcal{R}(t_i) \big). \tag{4.6}$$

$\square$

4.2 Estimation and Inference

4.2.1 Estimation of β

An estimator $\hat{\beta}$ of β is obtained by solving the score equations

$$U(\hat{\beta}) = 0.$$

Because the score equations are nonlinear in general, numerical methods (typically Newton–Raphson or Fisher scoring) are used. The Newton–Raphson update is

$$\beta^{(r+1)} = \beta^{(r)} + \big[I(\beta^{(r)}) \big]^{-1} U(\beta^{(r)}),$$

iterated until convergence. Under regularity conditions and non-informative censoring, $\hat{\beta}$ is consistent and asymptotically normal:

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow{d} N(0, \Sigma), \qquad \Sigma = I(\beta)^{-1},$$

and in practice the estimated covariance of $\hat{\beta}$ is $\widehat{\mathrm{Var}}(\hat{\beta}) = I(\hat{\beta})^{-1}$.

4.2.2 Inference

Using the estimated covariance, one can compute Wald-type tests and confidence intervals:

$$\hat{\beta}_j \pm z_{\alpha/2} \sqrt{\widehat{\mathrm{Var}}(\hat{\beta}_j)}.$$

Also available are the score test (based on $U(\beta_0)$ and information at β_0) and the partial-likelihood ratio test comparing nested models:

$$2\{\ell_P(\hat{\beta}_{\text{full}}) - \ell_P(\hat{\beta}_{\text{reduced}})\} \overset{approx}{\sim} \chi^2_{df}.$$

4.2.3   Treatment of tied event times

The partial likelihood formulas above assume no ties (no two events at exactly the same observed time). Two common approaches when ties occur:

Breslow's approximation

If at time t there are d events and the set of failed subjects is $\mathcal{D}(t)$, Breslow approximates the contribution to the partial likelihood by

$$\prod_{i \in \mathcal{D}(t)} \frac{\exp(\beta^\top X_i)}{\sum_{j \in \mathcal{R}(t)} \exp(\beta^\top X_j)} = \frac{\prod_{i \in \mathcal{D}(t)} \exp(\beta^\top X_i)}{\left(\sum_{j \in \mathcal{R}(t)} \exp(\beta^\top X_j)\right)^d}.$$

This is simple but can be biased if many ties exist.

Efron's approximation

Efron gives a better approximation for moderate numbers of ties. Let

$$S(\beta) = \sum_{j \in \mathcal{R}(t)} \exp(\beta^\top X_j), \qquad S_{\mathcal{D}}(\beta) = \sum_{i \in \mathcal{D}(t)} \exp(\beta^\top X_i),$$

and let $d = |\mathcal{D}(t)|$. Efron's contribution to the partial likelihood is approximated by

$$\frac{\prod_{i \in \mathcal{D}(t)} \exp(\beta^\top X_i)}{\prod_{l=0}^{d-1} \left[S(\beta) - \frac{l}{d} S_{\mathcal{D}}(\beta)\right]}.$$

Efron's method is commonly used in software as a compromise between exact discrete-likelihood computation and simplicity.

(An "exact" conditional likelihood for ties can be computed by considering all orders of failures among tied observations; this is usually computationally expensive when d is large.)

4.2.4 Estimation of the baseline cumulative hazard and survival

After $\hat{\beta}$ is obtained, the (profile) baseline cumulative hazard $H_0(t) = \int_0^t \lambda_0(u)\, du$ can be estimated using Breslow's estimator:

$$\widehat{H}_0(t) = \sum_{k:t_{(k)} \leq t} \frac{d_k}{\sum_{j \in \mathcal{R}(t_{(k)})} \exp(\hat{\beta}^\top X_j)}, \tag{4.7}$$

where d_k is the number of events at time $t_{(k)}$. (For tied times one replaces the denominator appropriately depending on tie method.) The estimated baseline survival function is

$$\widehat{S}_0(t) = \exp\{-\widehat{H}_0(t)\},$$

and the survival function for a subject with covariate X is estimated by

$$\widehat{S}(t \mid X) = \widehat{S}_0(t)^{\exp(\hat{\beta}^\top X)}.$$

Chapter 5

Proposed Models

In this section, we describe the mathematical frameworks of the models used for comparison in the subsequent sections.

5.1 Extended Cox-type model for Recurrent Data

For the i-th pipe, the hazard function for recurrent events is modeled differently based on the number of failures up to that point. Thus, the hazard/intensity function for the m-th failure is defined as follows:

$$\lambda_{i,m}(t \mid H_t) = \begin{cases} \lambda_0(t) \, \exp\left\{\sum_{k=1}^p \beta_{mk} X_{ik} + \sum_{j=1}^{m-1} \alpha_j(t - t_j)\right\}, & \text{if } m > 1, \\ \lambda_0(t) \, \exp\left\{\sum_{k=1}^p \beta_{mk} X_{ik}\right\}, & \text{if } m = 1. \end{cases}$$

where $\lambda_0(t)$ denotes the baseline intensity function, $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the matrix of baseline covariates, β_{mk} are the corresponding covariate coefficients, and $\alpha_j(\cdot)$ parameterizes the contribution of past events to the conditional intensity. It differs from the classic Cox-model in two fundamentals ways: (1) the effect of the covariates on the intensity changes overtime with each failure as captured by $\beta_{\cdot,k}$. (2) It partially violates the classic Cox assumption of the exponent term being entirely time-independent by incorporating the temporal differences with the previous failures as additional covariates.

This framework provides a dynamic hazard model for recurrent events, with the hazard rate adapting based on the history of past failures. The integration of time-dependent covariates $(t - t_j)$ ensures that the model captures the temporal dependencies between consecutive failures.

Other processes, such as the Neural Hawkes Process (NHP) [5] and the Transformer Hawkes Process (THP) [13], employ a similar setup for modeling recurrent event data but define the intensity (hazard) function as a neural network-driven function of hidden representations. Additionally, they also include the event types in their setup which we also plan to incorporate to our model in future endeavors. These representations are generated based on the event history, including timestamps and event types, enabling the models to capture complex temporal dependencies and patterns.

5.2 Neural Hawkes Process (NHP)

The intensity function for the Neural Hawkes Process is modeled as:

$$\lambda_k(t \mid H_t) = \sum_{k=1}^{K} f_k \left(\mathbf{w}_k^\top h(t) \right),$$

where K is the no. of distinct event types, $h(t)$ represents the hidden state at time t, $\mathbf{w}_k$ are learnable parameters, and f_k are non-linear functions, often implemented as neural networks. The NHP uses a continuous-time Long Short-Term Memory (LSTM) architecture to update the hidden state $h(t)$ dynamically as events occur. The model integrates time intervals explicitly into its hidden state updates, making it well-suited for capturing non-linear dependencies and complex temporal dynamics in the data. [11] improves on NHP by incorporating the base covariates of the data into the hidden representations $h(t)$ to capture more context - an idea we will extend in our novel implementation of the Transformer Hawkes Process.

5.3 Transformer Hawkes Process (THP)

The intensity function for the Transformer Hawkes Process is defined as:

$$\lambda \left(t \mid \mathcal{H}_t \right) = \sum_{k=1}^{K} \lambda_k \left(t \mid \mathcal{H}_t \right)$$

$$\lambda_k \left(t \mid \mathcal{H}_t \right) = f_k(\underbrace{\alpha_k \frac{t - t_j}{t_j}}_{\text{current}} + \underbrace{\mathbf{w}_k^\top \mathbf{h}(t)}_{\text{history}} + \underbrace{b_k}_{\text{base}}), t \in [t_j, t_{j+1})$$

where $h(t)$ is the hidden representation generated by a transformer-based architecture and the common choice of f_k is the softplus function, $f_k(x) = \beta_k \log(1 + \exp(x/\beta_k))$. Unlike NHP,

THP leverages self-attention mechanisms to model temporal dependencies, capturing long-range interactions between past events. The input to the transformer includes the event timestamps and types, encoded using positional encodings and embeddings, respectively. The self-attentive architecture allows THP to handle larger datasets and events with more intricate temporal patterns, outperforming traditional recurrent architectures in many scenarios.

By utilizing neural network architectures, both NHP and THP can overcome the limitations of traditional Hawkes processes, such as the difficulty in modeling non-linear dependencies and scalability issues. These approaches represent a significant advancement in the modeling of recurrent event data.

However, a noteworthy point is that neither the Neural Hawkes Process (NHP) nor the Transformer Hawkes Process (THP) initially incorporated the original covariate values of the base data. To address this limitation, [11] introduced a baseline term to the NHP intensity function to account for the covariates. Motivated by this, we modified the THP intensity function to include a linear combination of the base covariate values of the pipes, weighted by learnable parameters. The revised intensity function is given by:

$$\lambda_k \left(t \mid \mathcal{H}_t \right) = f_k \left(\underbrace{\alpha_k \frac{t - t_j}{t_j}}_{\text{current}} + \underbrace{\mathbf{w}_k^\top \mathbf{h}(t)}_{\text{history}} + \underbrace{\boldsymbol{\beta}_k^\top \mathbf{X}}_{\text{base covariates}} + \underbrace{b_k}_{\text{base}} \right), \quad t \in [t_j, t_{j+1}),$$

where: - $\alpha_k \frac{t-t_j}{t_j}$: Captures the current time interval information relative to the last event t_j, - $\mathbf{w}_k^\top \mathbf{h}(t)$: Encodes the historical dependency using the hidden representation $\mathbf{h}(t)$ generated by the transformer architecture, - $\boldsymbol{\beta}_k^\top \mathbf{X}$: Represents the contribution of the base covariates $\mathbf{X}$, with $\boldsymbol{\beta}_k$ as learnable weights and b_k is the base/bias term present in the original model.

This modification integrates the covariate information directly into the model, improving its capacity to leverage static and dynamic features for recurrent event prediction.

Chapter 6

Results

6.1 Data Description And Exploratory Data Analysis

The dataset comprises information on 10,203 pipes installed between 1860 and 2018, divided into two sections: the **installation profile** and the **failure history**. The installation profile contains 12 covariates, such as diameter, length, age, material, soil type, and other relevant attributes, which provide a detailed characterization of each pipe. The failure history records the timestamps of failures (in years since installation) along with the types of failures, categorized into seven types: Ageing, Soil Subsidence, Corrosion, Unknown Cause, Material Defect, Tree Roots, and Pressure. Each failure event is represented as a pair (t_j, k_j), where t_j is the failure timestamp, and k_j is the numerically encoded failure type. For instance, an exemplary failure sequence is $\{(76.41, 3), (80.42, 3), (80.97, 5), (81.12, 4)\}$.

The profile attributes are used to estimate the **baseline intensity** for each pipe, providing a static risk assessment. On the other hand, the failure history is employed to model the **historical effects** on the intensity or hazard rate, which evolves over time based on past failures, using a Hawkes process-based approach.

6.1.1 Dataset Splits and Characteristics

The dataset is divided into training and testing sets. The training set includes the complete profile information and the entire failure event sequence for each pipe until it is deemed unusable. In contrast, the testing set contains the profile data, but the failure event sequence is truncated, including only the first one or a few failure times. This setup simulates a real-world scenario where limited failure history is available for prediction.

The failure-event sequences are characterized by varying lengths, with a minimum of 1 event,

an average of 3 events, and a maximum of 5 events per pipe. Table 6.1 summarizes the sequence characteristics.

Table 6.1: Characteristics of failure event-time sequence

Dataset	# Event Types	Sequence length		
		Min	Average	Max
Pipe-Failure	7	1	3	5

6.1.2 Existing Prediction and Evaluation

In the testing set, future failure times are predicted until the hazard rate exceeds a specified threshold, indicating that the pipe has reached its economic end-of-life. This allows for estimating the **remaining useful lifetime (RUL)** of the pipe based on its failure history. The last predicted failure time is compared to the actual last failure time, and the prediction accuracy is assessed using **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)**.

Reported results from [11] indicate MSE and MAE values of 2.15 and 1.59 for the training set, and 2.17 and 1.64 for the testing set, respectively. These metrics demonstrate the model's efficacy in predicting failure events and estimating the RUL of pipes, which is crucial for maintenance planning and resource allocation.

6.1.3 EDA and Pre-Processing of Profile Features

- **Dataset Overview:**

  - The dataset contains **10,203** entries and **14** columns.

  - Features include both numeric values (e.g., diameter, age, length) and categorical features (e.g., material, vegetation).

- **Feature Details:**

  - **UITWENDIGE** (External Diameter): Represents the external diameter of the unit in millimeters.

  - **INWENDIGE** (Internal Diameter): Represents the internal diameter of the unit.

  - **NOMINALE_D** (Nominal Diameter): Nominal diameter or dimension of the unit.

  - **Age**: Represents the age or year of construction of the unit.

  - **LENGTE_GIS** (Length from GIS): Length of the unit obtained from GIS data in meters.

- **FUNCTIE_LE** (Function Type): Indicates the function or role of the unit, e.g., *Distributieleiding* (Distribution Pipe).

- **NETWERK** (Network Type): Represents the type of network, e.g., DN stands for Diameter Nominal.

- **MATERIAAL** (Material): Indicates the construction material of the unit, e.g., PVC (Polyvinyl Chloride).

- **Aansluiting** (Connection): Represents the number of connections or connections per unit length.

- **distance**: Distance between this unit and another reference point, possibly in meters.

- **Soil_type** (Soil Type): Type of soil where the unit is situated, which could influence risk or installation requirements.

- **Vegetation**: Type of vegetation surrounding the unit, e.g., *High green* or *Rural*.

- **Covariate Types:**

  - **Categorical Variables:**

    — **FUNCTIE_LE** (Function Type): Categories include *Distribution Pipe, Connection Pipe,* etc.

    — **NETWERK** (Network Type): Categories describe the type of network such as DN.

    — **MATERIAAL** (Material): Construction material types, e.g., PVC, Steel.

    — **Soil_type**: Types of soil such as *Sand, Clay,* etc.

    — **Vegetation**: Descriptions of vegetation around the unit like *High green, Urban, Rural*.

- **Numeric Feature Distribution:**

  - Histograms were plotted for numeric features to understand their distributions.

  - The figure below illustrates the distributions for selected numeric features.

- **Categorical Feature Analysis:**

  - Countplots were utilized to visualize the distribution of categorical features.

  - The plot below depicts the counts for the categorical features.

  - Pie charts were created to visualize the proportion of categories within each categorical feature.
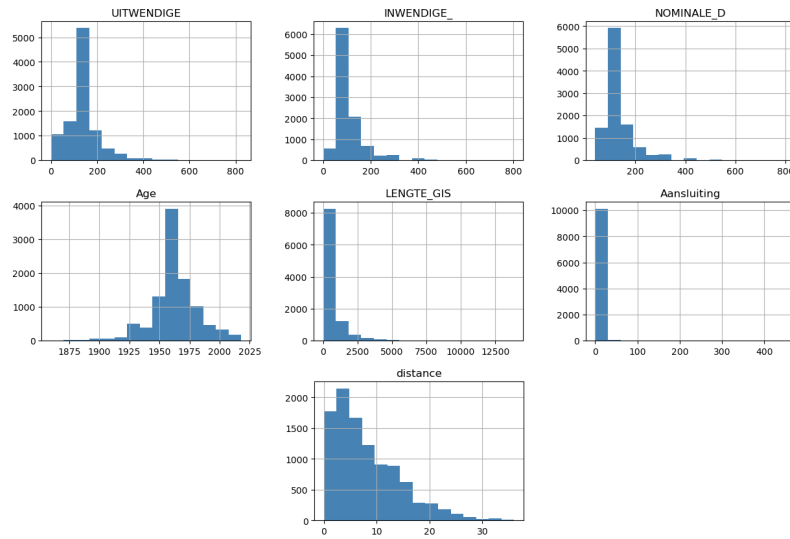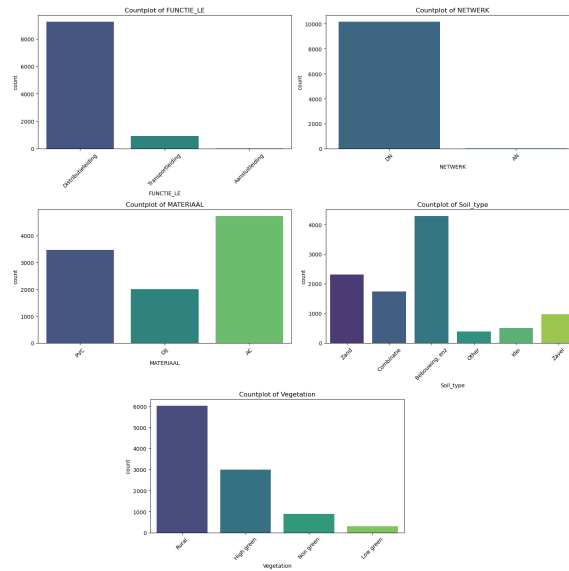
Figure 6.1: Distribution of Numeric Features

Figure 6.2: Countplot of Categorical Variables

- **Correlation Analysis:**

  - A correlation heatmap was generated to visualize relationships between numeric features.

  - Strong correlations were observed between diameter-related features.
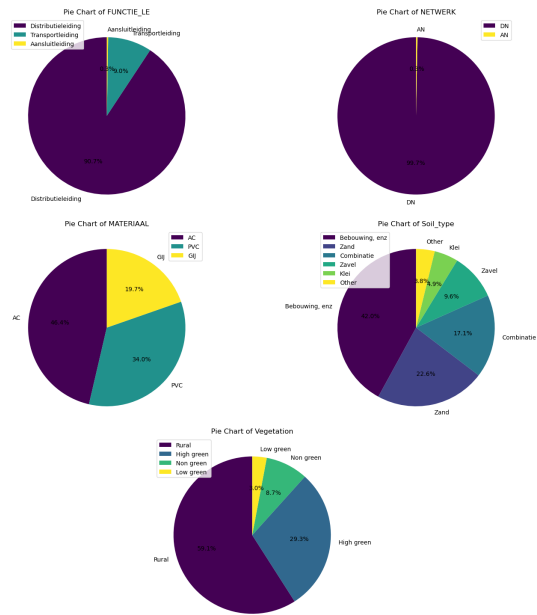
- **Pairwise Relationships:**
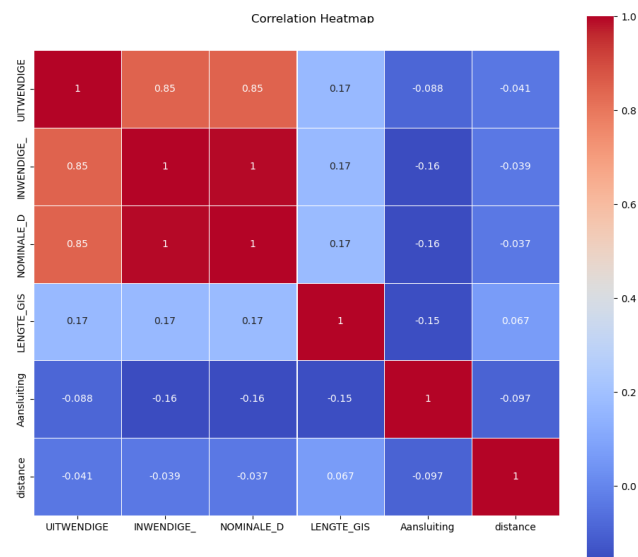
Figure 6.3: Combined Pie Charts of Categorical Features

Figure 6.4: Correlation Heatmap

- A pairplot was employed to examine relationships between the top numeric features.

- The scatter plots highlight potential trends or clusters due to diameter-related factors.

- **Boxplots by Category:**
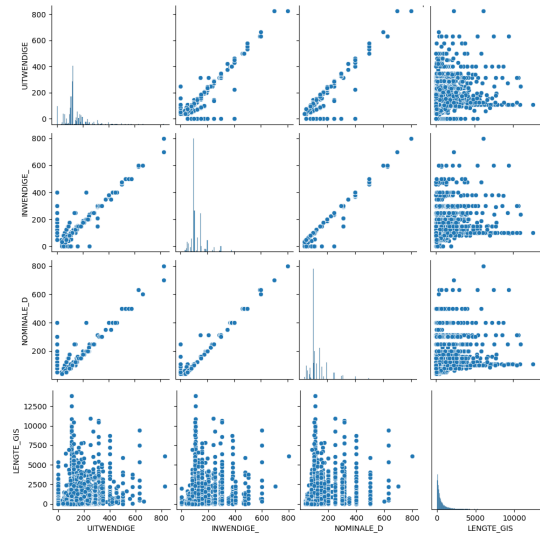
Figure 6.5: Pairplot of Selected Numeric Features

– Boxplots were used to explore the distribution of numeric features across different categorical features.

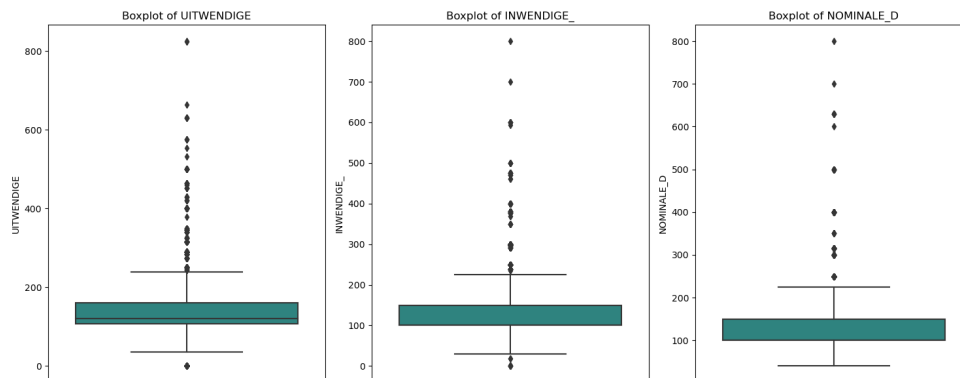– The plots below show the distribution of numerical variables, highlighting possible outliers.

Figure 6.6: Boxplots of Numeric Covariates

• **Preprocessing:**

  – **Standardization:**

    — Performed standardization on numerical covariates to ensure they have a mean of 0 and a standard deviation of 1. This step is crucial for algorithms that are sensitive to the scale of data.
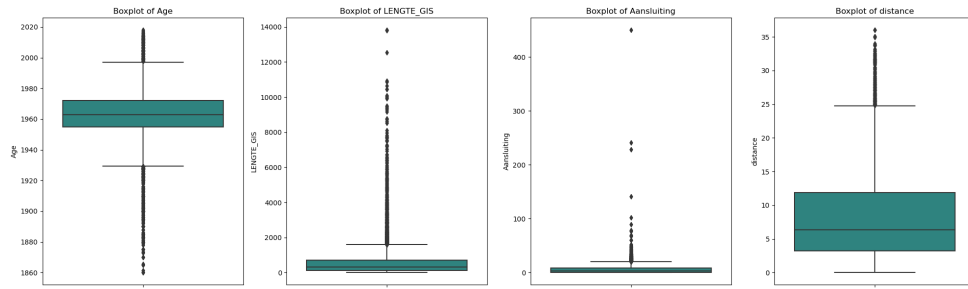
Figure 6.7: Boxplots of Numeric Covariates (2)

- **Encoding Categorical Variables:**

  — Categorical covariates were encoded using appropriate techniques such as one-hot encoding for nominal variables and label encoding for ordinal variables.

- **Dropping Unnecessary Covariates:**

  — `pipe_id` and `date` were removed from the dataset as they were not relevant for the analysis and modeling.

  — Features exhibiting high correlation with each other were also dropped to reduce redundancy and mitigate multicollinearity issues. This helps in improving the performance of predictive models. e,g., UITWENDIGE, INWENDIGE and NOMINALED all represent some form of diameter of the pipes and hence are higly correlated so only NOMINALED was taken. Also NETWERK (Network Type) was highly imbalanced and was hence dropped.

6.2 Analysis of Survival Curves

The Kaplan-Meier survival curves provide an estimate of the survival probability over time. These plots are grouped by vegetation type, soil type, and material to highlight differences in survival probabilities based on covariates. These plots suggest that the covariates proportionally effect the survival rate and hence justifies our choice of the Cox Proportional Hazards Model.
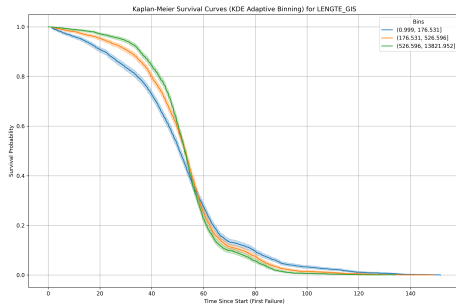

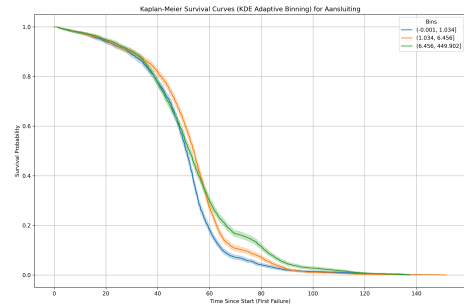Figure 6.8: Kaplan-Meier Survival Curves (KDE Adaptive Binning) for `LENGTE_GIS`

Figure 6.9: Kaplan-Meier Survival Curves (KDE Adaptive Binning) for `Aansluiting`
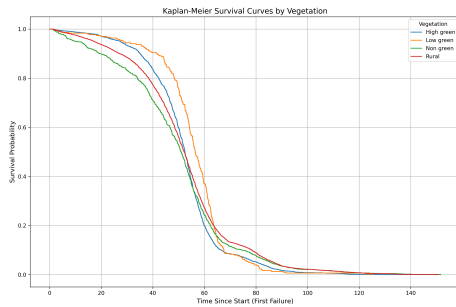

Figure 6.10: Kaplan-Meier Survival Curves by Vegetation Type
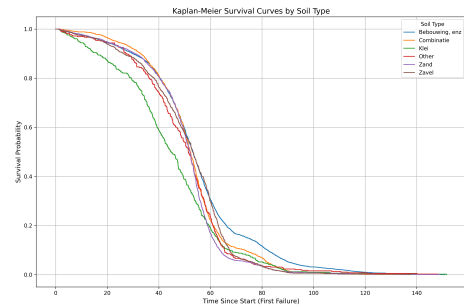

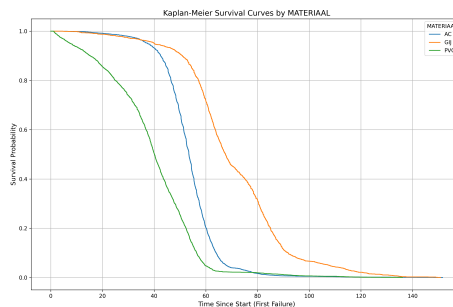Figure 6.11: Kaplan-Meier Survival Curves by Soil Type

Figure 6.12: Kaplan-Meier Survival Curves by Material Type

The following hazard curves for the categorical variables further justify our cause, and the plots for the numerical variables using kernel density adaptive binning are quite similar (using a Gaussian kernel).
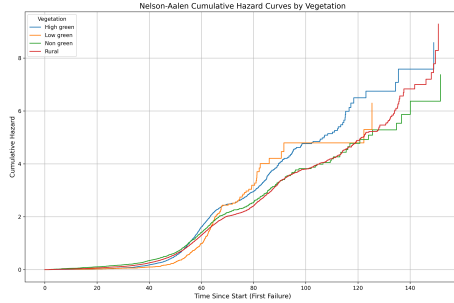

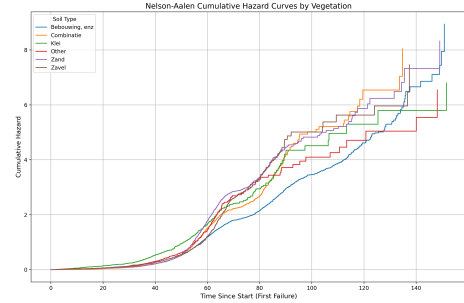Figure 6.13: Nelson-Aalen Cumulative Hazard Curves by Vegetation Type

Figure 6.14: Nelson-Aalen Cumulative Hazard Curves by Soil Type
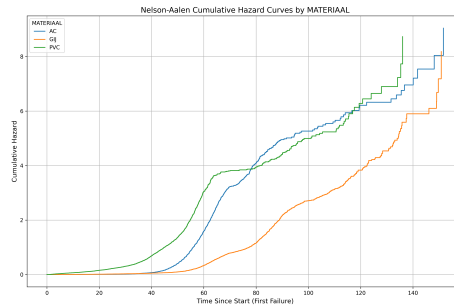
Figure 6.15: Nelson-Aalen Cumulative Hazard Curves by Material Type

The survival curves for the 1st to 5th failures show crossing patterns, indicating that the baseline hazard function changes over time. This justifies the use of the Weibull baseline intensity in the extended Cox-type model, as it is flexible enough to capture these dynamic changes in hazard rates effectively.
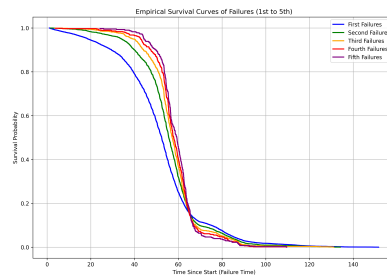

Figure 6.16: Survival Curves for the 1st to 5th Failures

6.3 Estimation and Prediction

The log-likelihood function for the Hawkes process and the extended Cox-type model is given by:

$$\ell(\mathcal{S}) = \underbrace{\sum_{j=1}^{L} \log \lambda\left(t_j \mid \mathcal{H}_j\right)}_{\text{event log-likelihood}} - \underbrace{\int_{t_1}^{t_L} \lambda\left(t \mid \mathcal{H}_t\right) dt}_{\text{non-event log-likelihood}} .$$

This function consists of two terms: the event log-likelihood, which captures the contribution of observed events, and the non-event log-likelihood, which accounts for the intervals where no events occur. The non-event log-likelihood is approximated using either Monte Carlo Integration, providing an unbiased estimate, or Numerical Integration techniques such as the trapezoid rule. Similarly, the gradient of the log-likelihood can be approximated using these methods for efficient computation.

The extended Cox-type model leverages this same log-likelihood formulation, treating the intensity function $\lambda(t \mid \mathcal{H}_t)$ as a parameterized hazard function. Chapter 3 describes in detail how to derive the conditional likelihood of a Hawkes process based on its conditional intensity function, further linking the statistical framework of the two models.

The predicted next timestamp and the corresponding event type are calculated using the following expressions:

$$\widehat{t}_{j+1} = \int_{t_j}^{\infty} t \cdot p\left(t \mid \mathcal{H}_t\right) dt, \quad \widehat{k}_{j+1} = \underset{k}{\operatorname{argmax}} \frac{\lambda_k\left(t_{j+1} \mid \mathcal{H}_{j+1}\right)}{\lambda\left(t_{j+1} \mid \mathcal{H}_{j+1}\right)}.$$

Here, $\widehat{t}_{j+1}$ is computed numerically using integration techniques to estimate the expected time of the next event, while $\widehat{k}_{j+1}$ is determined by maximizing the normalized conditional intensity function over all possible event types. This approach ensures that the model captures both temporal and categorical dynamics of recurrent events effectively. However, for our Cox-type setup, the event type $\widehat{k}_{j+1}$ is irrelevant as we do not include the seperate intensities/hazard rates for the different failure types even though we plan to add them in later works.

6.3.1 Experimental Results using Extended Cox-type Model

The model was trained on the training sets, and its predictive accuracy was evaluated on the test set by calculating the root mean squared error (RMSE) between the actual and predicted failure times. A Weibull baseline hazard rate as used for our model as it felt more appropriate given the

prior discussion on the data. Additionally, the log-likelihood values were recorded during the training process to monitor model convergence. The RMSEs and log-likelihoods were averaged across 10 seeds to provide a robust measure of model performance.

The results recorded for the test data were an average RMSE of 2.6327 and an average log-likelihood of 0.3267, reflecting the model's ability to generalize to unseen data. For the training data, the model achieved an average log-likelihood of 0.5378 and an average RMSE of 2.5986, indicating good fit on the training samples while maintaining a reasonable level of complexity.

6.3.2 Experimental Results using THP

The experiment was conducted using the specified parameters to train the model. The data was sourced from the directory `data/data_pipe/`, with a batch size of 4. The model architecture consisted of 4 layers, 4 attention heads, and a model dimension (d_{model}) of 512. The transformer hidden size ($d_{\mathrm{transformer}}$) was set to 64, while the inner dimension of the feed-forward network (d_{inner}) was 1024. The key and value dimensions (d_k and d_v) were both set to 512. A dropout rate of 0.1 was applied, and the learning rate (lr) was fixed at 1×10^{-4}. Label smoothing was applied with a smoothing factor of 0.1. The training was run for 100 epochs, and logs were saved in `log.txt`. The experiment utilized device 1 for computations. This model was also evaluated across 10 seeds.

The best performance was recorded during the 99th epoch, achieving a test set RMSE of 1.7719 and an event type accuracy of 74.86%, with the corresponding log-likelihood value being 1.55501. On the training set, the RMSE was lower at 1.5601, log-likelihood of 1.5961 with an accuracy of 78.88%. These results demonstrate the model's strong capability to predict recurrent events while maintaining a high level of accuracy, particularly on the training data.

6.3.3 Comparative Analysis

The tables summarize the comparative performance of the NHP, THP, and the extended Cox-type models based on Log-Likelihood, RMSE, and Accuracy metrics. Our novel extension of THP significantly outperforms both the NHP implementation by [11] on this data as well the extended Cox-model.

As shown in Table 6.2, the test performance of the models varies significantly based on the RMSE, Log-Likelihood, and Accuracy metrics. For NHP, the log likelihood and Accuracy values was not reported by the paper [11] and due to time constraints, their results were not reproduced or recoded.

Table 6.2: Test Results for NHP, THP, and Cox-type Models

Model	Log-Likelihood	RMSE	Accuracy
NHP [11]	-	2.17	-
THP	1.56	1.77	74.86%
Cox-type	0.33	2.63	-

The training results in Table 6.3 highlight the differences in model performance on the training data, with the THP showing higher RMSE compared to NHP and Cox model. The results are as expected better than the test results

Table 6.3: Training Results for NHP, THP, and Cox-type Models

Model	Log-Likelihood	RMSE	Accuracy
NHP [11]	-	2.15	-
THP	1.60	1.56	78.88%
Cox-type	0.54	2.60	-

Chapter 7

Conclusion & Future Work

This project explored three major advancements in the modeling of recurrent event data: (1) extending the Transformer Hawkes Process (THP) to incorporate covariate effects, thereby enabling the model to capture both static and dynamic influences on event occurrences, (2) developing a novel Cox-type extended hazards model specifically designed for recurrent event prediction and (3) comparing these models in a highly informative dataset. The enhanced THP model integrates baseline covariates to improve its applicability to real-world datasets, while the extended Cox-type model offers a flexible approach to predict failure times by leveraging both static profile data and historical event patterns.

A comprehensive comparison of these newer implementations was conducted against the Neural Hawkes Process (NHP) reported by [11]. The results demonstrate the potential of both models, with the THP excelling in its ability to handle complex temporal dependencies and the Cox-type model providing interpretable predictions with competitive accuracy. However, key limitations remain: the lack of extensive ablations for the THP model and the Cox-type model's insufficient theoretical background as compared to other extended Cox-type models. While building strong theory for the covariate augmented THP model is rather difficult due to its inherent transformer based architecture, the extended Cox-type model shows more promise in this regime. Future work will focus on addressing these limitations by systematically optimizing the THP model's parameters and constructing more scalable estimators theoretically for the extended Cox model. Additionally, we also plan to incorporate the event classification for the Cox-type model as a major part of our future research. These improvements aim to further enhance the performance and scalability of the proposed approaches in recurrent event modeling

Bibliography

[1] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

[2] A. G. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443, 1971.

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

[4] John D. Kalbfleisch and Ross L. Prentice. *The Statistical Analysis of Failure Time Data*. John Wiley & Sons, New York, 1980.

[5] Hongyuan Mei and Jason Eisner. Neural hawkes process: A neurally self-modulating multivariate point process. *Advances in Neural Information Processing Systems*, 30:6754–6764, 2017.

[6] Tohru Ozaki. Maximum likelihood estimation of hawkes' self-exciting point processes. *Annals of the institute of statistical mathematics*, 31(1):145–155, 1979.

[7] P. D. Rogers and N. S. Grigg. Failure assessment model to prioritize pipe replacement in water utility asset management. In *Water Distribution Systems Analysis Symposium 2006*, pages 1–17, 2008.

[8] Adrian Segall. Recursive estimation from discrete-time point processes. *IEEE transactions on Information Theory*, 22(4):422–431, 2003.

[9] D Snyder. Smoothing for doubly stochastic poisson processes. *IEEE Transactions on Information Theory*, 18(5):558–562, 1972.

[10] D Vere-Jones. On updating algorithms and inference for stochastic point processes. *Journal of Applied Probability*, 12(S1):239–259, 1975.

[11] J. Verheugd, P.R.D.O. da Costa, R.R. Afshar, Y. Zhang, and S. Boersma. Predicting water pipe failures with a recurrent neural hawkes process model. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2628–2633. IEEE, October 2020.

[12] Qingyun Zhang, Hongteng Zheng, and Hongyuan Zha. Self-attentive hawkes process. In *Advances in Neural Information Processing Systems*, volume 33, pages 20745–20757, 2020.

[13] Simiao Zuo, Hao Jiang, Yichen Huang, Yitao Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International Conference on Machine Learning*, pages 11692–11702, 2020.