

Modeling Recurrent Events On A Network: Network Poisson Process (*NPP*)

JACKSON WATSON

May 2025

Contents

1 Abstract	4
2 Introduction	5
2.1 Literature Review	5
2.2 Motivation	8
3 Network Poisson Process (<i>NPP</i>) Model Formulation	9
3.1 Poisson Process	9
3.2 Network Lasso	10
4 Parameter Estimation: ADMM	14
5 Network Homogenous Poisson Process + Static Covariates	
Case Study: US Oil And Gas Well Dataset	16
5.1 Dataset Description (Liu and Pan 2020)	16
5.2 Summary Statistics	17
5.3 Modeling	20
5.4 Forecasting	21
5.5 Forecasting Performance Metrics	22
5.6 Forecasting Results	23
5.7 Inference On Existing Nodes	24
5.8 Inference On New Nodes	26
5.9 Inference On New Nodes: Results	27
6 Network Homogenous Poisson Process + Static Covariates	
Case Study: Scotland Pipe Dataset	29
6.1 Dataset Description (Xiao, Ye and Revie, 2024)	29
6.2 Summary Statistics	31
6.3 Forecasting Results	33

6.4	Inference	34
7	Network <i>NHPP</i> + Dynamic Covariates	36
7.1	Further Model Formulation	36
8	Network NHPP + Dynamic Covariates Case Study: US Oil And Gas Well Dataset	37
8.1	Summary Statistics	37
8.2	Modeling	39
8.3	Modeling Complexities	40
8.3.1	Complexity 1: Unstable $\mu = 0$ Parameter Estimates .	40
8.3.2	Complexity 2: Exponential Function Asymmetry . .	41
8.4	Results	42
9	Conclusion	43
10	Appendix	44
10.1	Convexity of the Negative Log-Likelihood (1)	44
10.2	Why the L2 Norm of Differences Leads to Stratification	46
10.3	ADMM Derivation for Network Lasso	48
10.4	Rolling Expectation for a Non-Homogeneous Poisson Process with Dynamic Covariate	52

1 Abstract

A method of modelling recurrent events on a network called Network Poisson Process (*NPP*) is proposed. It fits a separate Poisson process at each node using both its historical data and that of its neighbours. *NPP* thus determines the extent to which data should be shared between related processes on a network. To fit such models, Hallac, Leskovec and Boyd's 2015 distributed optimisation model *network lasso* and The Alternating Direction Method of Multipliers (ADMM) are used. It is argued that *NPP* presents a robust and effective way to model data with static covariates in which the recurrent events at each node resemble a homogenous poisson process. 2 case studies are carried out on water pipes (332,570 nodes) and oil wells (8,232 nodes) in support of this argument as well as the claim that *NPP* works on massive datasets. Extending this framework to a NHPP and dynamic covariate context is discussed but not ultimately solved.

2 Introduction

2.1 Literature Review

Networks of interconnected systems, such as telecommunication base stations and repairable engineering systems, are integral to society’s infrastructure. These systems often experience recurrent failures, posing significant challenges for maintenance and reliability management. Predictive models for recurrent event data are essential to anticipate and mitigate these failures, ensuring uninterrupted service and operational efficiency (Meeker and Hong, 2014; Hong et al., 2018).

Prior to the proliferation of current day computational power, common ways of forecasting recurrent event data on a network included the Cox proportional hazards model and the Kaplan-Meier estimator. These methods were widely used due to their simplicity and ability to handle censored data. However, they oversimplify network recurrent event processes in some cases (Kalbfleisch and Prentice, 2002; Fleming and Harrington, 1991). For example, some such basic models did not capture dependencies between event (Lawless, 1995; Peña, Strawderman, and Hollander, 2001). Others struggled to incorporate covariates that vary over time (dynamic covariates), limiting their ability to reflect dynamic system behavior (Lin, Wei, and Ying, 2000; Diggle et al., 2002). Furthermore, spatial trends were often ignored, as early survival models typically assumed independence across units (Clayton and Cuzick, 1985; Lawson, 2001).

Modern analytical methods have significantly advanced our ability to model ‘Big Data’ (Liu and Pan, 2020) networks with spatially similar recurrent events: Heterogeneity across units or individuals—such as differing baseline risks or frailty—is now regularly modelled using random effects or frailty terms, improving the accuracy of predictions (Diggle et al., 2002). Deep

learning and Bayesian frameworks have also been used to address system heterogeneity (Jin et al., 2023; Sahu and Mardia, 2005; Zhang, Liu, and Pan, 2025).

In parallel, modern statistical and machine learning approaches have improved the integration of static and dynamic covariates into recurrent event models (Nirmalkanna and Cigsar, 2024; Zhang and Shi, 2024; Ke, 2023). The challenge of modelling cascade failures—where one failure triggers a chain of subsequent failures—within complex networks has been of particular interest recently (Wu, Wang, Ye and Chen, 2024; Zhai, Ye, Li, Revie and Brunsen, 2024). These cascading effects are especially relevant in interdependent systems like power grids, communication networks, and transportation systems (Valdez et al., 2020; Zhou et al., 2023).

Beyond big data concerns, covariates, and cascading failures, modern recurrent event analysis also addresses several foundational statistical concerns. One such issue is censoring, particularly informative censoring, where the likelihood of dropout or study termination is related to the event process itself. Here, newer approaches (Wang, Qin, and Chiang, 2001; Andersen and Gill, 1982) incorporate latent variable frameworks and inverse probability weighting to correct for bias. Another concern is the distributional assumption underlying arrival processes. While early models often relied on exponential or Poisson assumptions, modern techniques allow for flexible, semi-parametric, or non-parametric modeling of inter-arrival times (Zhou and Hanson, 2018).

This area of research is rapidly developing (Zhang et al., 2025; Zhou et al., 2025), driven by the increasing availability of Big Data from advanced sensing technologies and the proliferation of IoT devices (Hennerfeind et al., 2006; Li and Ryan, 2002). As networks continue to expand and evolve, the demand for innovative predictive models will only grow, making this field a vibrant

and crucial domain for future research (Li and Lin, 2006; Zhao and Hanson, 2011; Hanson et al., 2012).

2.2 Motivation

This paper aims to establish *NPP* as an approachable yet effective method for addressing 2 topical issues in modelling recurrent events on modern, ‘Big Data’ networks: Incorporating static covariate information and achieving computational efficiency in model fitting. NHPP terms and dynamic covariates can also be added to this framework - The following ‘Model Formulation’ section thus considers the most general possible case of *NPP* which includes NHPP, static covariates and dynamic covariates. *NPP* with ‘homogenous Poisson process + static covariates’ and ‘NHPP + dynamic covariates’ are then considered separately as the latter is not ultimately solved. This may provide motivation for future research.

3 Network Poisson Process (*NPP*) Model Formulation

3.1 Poisson Process

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the vertex set and \mathcal{E} is the set of edges. The y_i arrival process times at each site i observed up until time T , $\mathbf{t}_i = [t_{i1}, \dots, t_{iy_i}]$, can be modeled by an *NHPP* with covariates. Let λ_i be the rate parameter of a particular node's Poisson distribution which is modeled as an exponential function of covariates and time dependent on the coefficient vector \mathbf{x}_i . Suppose we have c dynamic variables d_{ijt} which may take different observable values at each time t . For example, at any given time we can observe c values, $\mathbf{d}_{it} = [d_{i1t}, \dots, d_{ict}]$, at each site i . Therefore, up to $c + 2$ coefficients may be fit at each site such that: $\mathbf{x}_i = [x_{i0}, x_{i1}, \dots, x_{ic}, x_{it}] \in R^{c+2}$.

The model can be written as:

$$\lambda_i(t, \mathbf{d}_{it} | \mathbf{x}_i) = \exp \left(x_{i0} + \sum_{j=1}^c x_{ij} d_{ijt} + x_{it} t \right) \quad (0)$$

for all $t \geq 0$. To fit such a model, the negative log-likelihood function for a given rate λ_i is used (assuming access to historical dynamic covariate and failure time filtrations):

$$-\ell_i(\mathbf{x}_i) = - \sum_{k=1}^{y_i} \log(\lambda_i(t_k)) + \int_0^T \lambda_i(t) dt \quad (1)$$

3.2 Network Lasso

However, fitting separate Poisson processes at each vertex without data sharing would ignore relevant information from their neighbors that could improve parameter estimates. The above formulation is thus extended to a *network lasso* (Hallac, Leskovec and Boyd, 2015) context in which the problem is viewed holistically.

$$\text{minimise} \sum_{i \in \mathcal{V}} -\ell_i(\mathbf{x}_i) + \mu \sum_{(j,k) \in \mathcal{E}} w_{jk} \|\mathbf{x}_j - \mathbf{x}_k\|_2 \quad (2)$$

This system objective function balances two objectives: The node objective (given by ℓ above) and the edge objective. The edge objectives penalize differences between the variables at adjacent nodes, where the edge between nodes i and j has weight μw_{ij} . Here we can think of $w_{ij} > 0$ as setting relative weights among the edges of the network (for example euclidean distance). Thus, static covariates (such as longitude and latitude) should be used to determine the similarity between nodes and thus the extent to which they share data. $\mu > 0$ is an overall parameter that scales the edge objectives relative to the node objectives. A balance is thus struck between vertex data sharing and specificity in model fitting.

Some relevant features of *network lasso* (Hallac, Leskovec and Boyd, 2015):

- At $\mu = 0$ no data sharing occurs between nearby nodes, parameters are fit independently using (1).
- The ℓ_2 -norm penalty over the edge difference, $\|\mathbf{x}_j - \mathbf{x}_k\|_2$ incentivizes the differences between connected nodes to be exactly zero, rather than just close to zero. An edge difference of zero means that $\mathbf{x}_j = \mathbf{x}_k$. When many edges are in consensus like this, we have grouped the nodes into clusters with equal values of \mathbf{x}_i ¹. This fact (proved in Appendix) may surprise some readers given that, in ridge and lasso regression, ℓ_1 -norm penalties are used to shrink vectors of parameter estimates to 0 rather than ℓ_2 . 'LASSO' stands for 'Least Absolute Shrinkage and Selection Operator' meaning the word is somewhat abused in *network lasso*. However, *network lasso* and LASSO Regression both shrink estimates right to 0 which somewhat justifies its use.
- Cluster size tends to get larger as μ increases, until at μ_{critical} the consensus solution can be thought of as a single cluster for the entire network. The fitted $\hat{\mu}$ value can thus be used to infer practical details about the similarity of vertices. It's for this reason that models will be fit for multiple μ values (giving a 'regularization path'²).
- Given that $-\ell$ is a convex function (as shown in the Appendix), the parameters in (2) can be fit using the ADMM optimization procedure. These estimates are guaranteed to converge to global optima under iteration (Hallac, Leskovec and Boyd, 2015).

¹See Figure 1 for an example in which *network lasso* is used to infer geographical strata through housing prices. Note that λ is used as the regularisation parameter by Hallac, Leskovec and Boyd (2015) instead of μ .

²Figure 2

- ADMM allows computations to be split among different machine cores. It is also known for converging to, at least, moderately accurate solutions quickly. These features are of great practical importance in problems where \mathcal{V} , \mathcal{E} , and c (i.e. the number of parameters in (2)) are large.

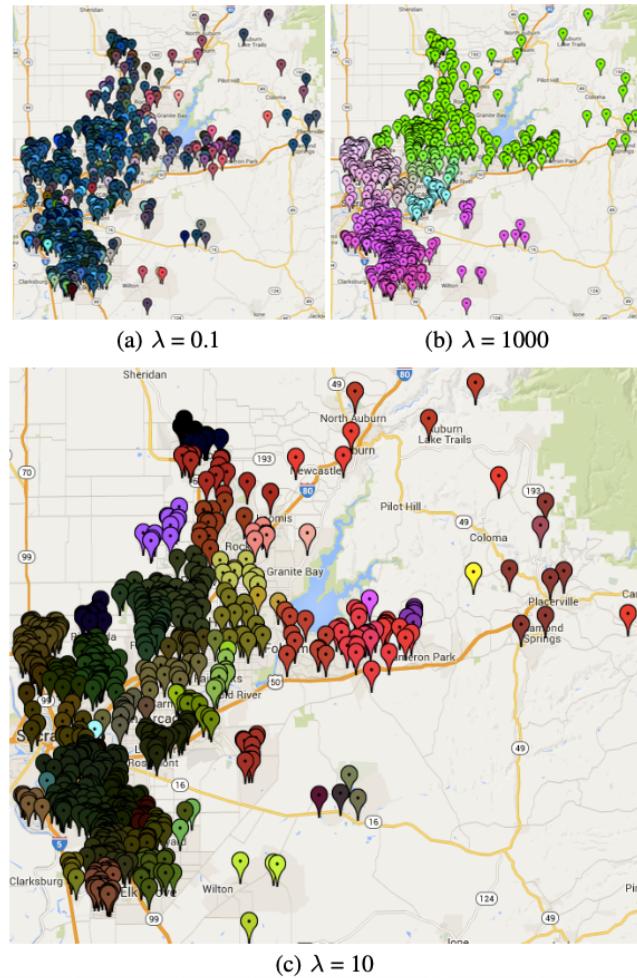


Figure 1: Exemplar Parameter Estimate Clustering From David Hallac, Jure Leskovec, and Stephen Boyd (2015), *Network Lasso*.

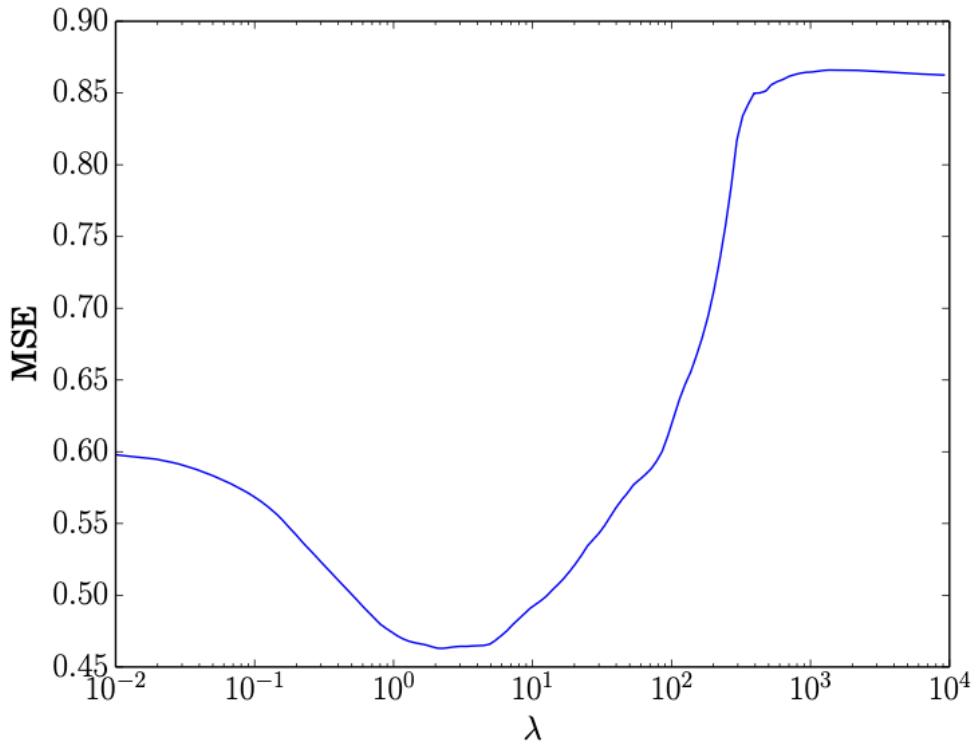


Figure 2: Exemplar Regularisation Path From David Hallac, Jure Leskovec, and Stephen Boyd (2015), *Network Lasso*.

4 Parameter Estimation: ADMM

To solve via ADMM (Hallac, Leskovec and Boyd 2015), a brief method sketch is given below. See the Appendix for a full proof. First, introduce a copy of \mathbf{x}_i , called \mathbf{z}_{ij} , at each edge ij . Note that the same edge also has a \mathbf{z}_{ji} , a copy of \mathbf{x}_j . We rewrite problem (2) as an equivalent problem:

$$\begin{aligned} \text{minimize}_{\mathbf{x}, \mathbf{z}} \quad & \sum_{i \in V} -\ell_i(\mathbf{x}_i) + \mu \sum_{(j,k) \in \mathcal{E}} w_{jk} \|\mathbf{z}_{jk} - \mathbf{z}_{kj}\|_2 \\ \text{subject to} \quad & \mathbf{x}_i = \mathbf{z}_{ij}, \quad i = 1, \dots, m \end{aligned}$$

We then derive its augmented Lagrangian:

$$\begin{aligned} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u}) = & \sum_{i \in V} -\ell_i(\mathbf{x}_i) + \mu \sum_{(j,k) \in \mathcal{E}} \mathbf{w}_{jk} \|\mathbf{z}_{jk} - \mathbf{z}_{kj}\|^2 \\ & - \frac{\rho}{2} \sum_{(j,k) \in \mathcal{E}} (\|\mathbf{u}_{jk}\|_2^2 + \|\mathbf{u}_{kj}\|_2^2) \\ & + \frac{\rho}{2} \sum_{(j,k) \in \mathcal{E}} (\|\mathbf{x}_j - \mathbf{z}_{jk} + \mathbf{u}_{jk}\|_2^2 + \|\mathbf{x}_k - \mathbf{z}_{kj} + \mathbf{u}_{kj}\|_2^2) \end{aligned}$$

where u is the scaled dual variable and $\rho > 0$ is the penalty parameter. To solve this problem while fixing a μ value, one may carry out (with $f_i = -\ell_i$):

$$\begin{aligned}
1 \quad & \mathbf{x}_i^{k+1} = \arg \min_{\mathbf{x}_i} \left(f_i(\mathbf{x}_i) + \sum_{j \in N(i)} \frac{\rho}{2} \|\mathbf{x}_i - \mathbf{z}_{ij}^k + \mathbf{u}_{ij}^k\|_2^2 \right) \\
2 \quad & \mathbf{z}_{ij}^{k+1} = \theta(\mathbf{x}_i^k + \mathbf{u}_{ij}^k) + (1 - \theta)(\mathbf{x}_j^k + \mathbf{u}_{ji}^k) \\
3 \quad & \mathbf{u}_{ij}^{k+1} = \mathbf{u}_{ij}^k + (\mathbf{x}_i^{k+1} - \mathbf{z}_{ij}^{k+1}) \\
& \theta = \max \left(1 - \frac{\mu \mathbf{w}_{ij}}{\rho \|\mathbf{x}_i^k + \mathbf{u}_{ij}^k - (\mathbf{x}_j^k + \mathbf{u}_{ji}^k)\|_2}, 0.5 \right)
\end{aligned}$$

This procedure is then repeated for different $\mu \in [0, \mu_{consensus}]$ to obtain a regularization path. The fitted values from this path are validated on test data to obtain final parameter estimates as well as performance heuristics. $\mathbf{u}_{ij} = 0$ and $\rho = 1$ are common starting values in ADMM. In some cases, ρ may require guess and check tuning to ensure convergence.

As previously mentioned, a number of this algorithm's features make it computationally efficient: The first step of the regularization path ($\mu = 0$) can be completed by solving (1) for each i independently. Although there are no closed form solutions to (1) or 1, these sub-problems are convex meaning that they can be solved quickly using standard programming libraries. They can also be carried out in parallel on different machine cores for each node i . Once these starting values are found, fitted values from previous steps of the regularization path may be passed on as good starting values in following steps. Furthermore, steps 2-3 can be carried out as matrix operations which are computationally cheap in standard programming libraries.

5 Network Homogenous Poisson Process + Static Covariates Case Study: US Oil And Gas Well Dataset

5.1 Dataset Description (Liu and Pan 2020)

According to data from the US Energy Information Administration (2017), there are currently more than a million active oil and gas wells in the United States. These wells are repairable, making it essential to accurately forecast future failures in order to direct maintenance efforts. A publicly accessible data set includes information on 8,232 wells installed between 2007 and 2017, detailing both failure events and a variety of system characteristics. The data set distinguishes between static attributes, such as design features, manufacturing data and installation parameters that remain unchanged, and dynamic sensor readings, which reflect real-time operational conditions and performance, offering a time-sensitive view of system behavior. As previously stated, this section will ignore dynamic covariates.

5.2 Summary Statistics

Figures 1-4 represent the cumulative number of failures that occurred in particular gas wells over time. There is a pronounced difference in intensity between the two pairs (Figure 1 and Figure 2) and (Figure 3 and Figure 4). This reflects the fact that each nodes' underlying failure process may be heterogeneous (to some extent). Additionally, these particular plots suggest (non-rigorously) that a homogeneous Poisson process may be a good data fit. Again, this section does not deal with *NHPP* so the time-independent nature of failure intensity is of interest here. Two more visualizations which support this assumption are given below.

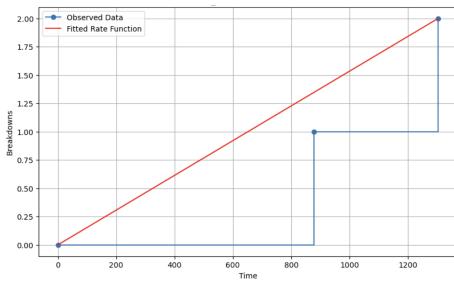


Figure 3: Node Cumulative Event Plot.

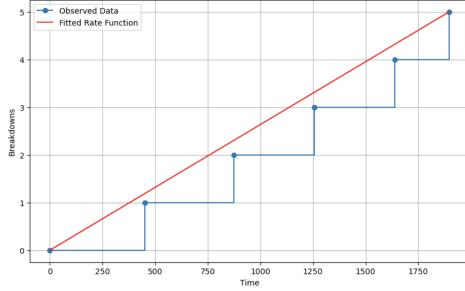


Figure 4: Node Cumulative Event Plot.

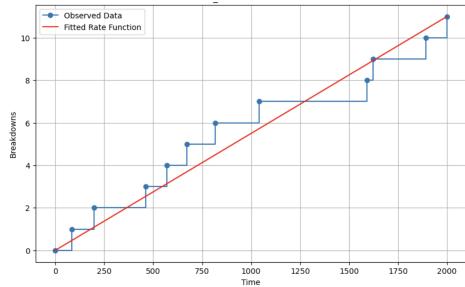


Figure 5: Node Cumulative Event Plot.

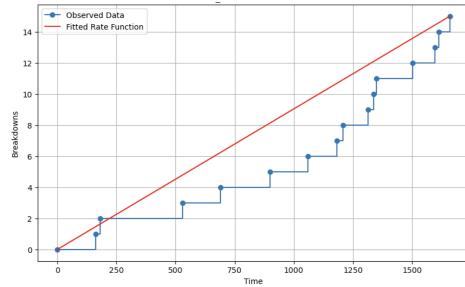


Figure 6: Node Cumulative Event Plot.

Figures 5-6 differ from 1-4 in that they treat the network as one point process, ignoring system heterogeneity. They therefore, once again, act as heuristics rather than empirical supporters of employing homogeneous Poisson processes for this data rather than *NHPP*.

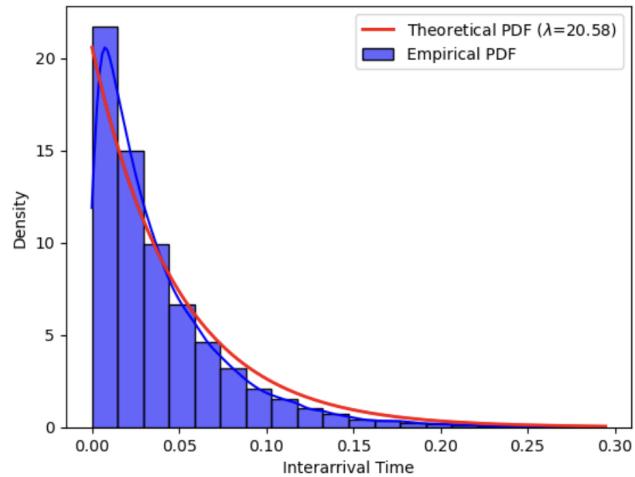


Figure 7: Empirical Vs Theoretical ($\lambda = 20.58$) PDF Arrival Times.

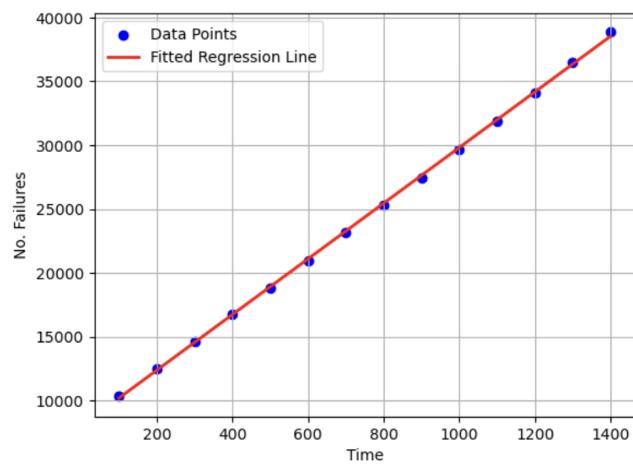


Figure 8: Network Cumulative Event Plot.

Areas of the oil and gas well network with similar longitude and latitude seemed to exhibit similar failure patterns. Specifically, those in the bottom right hand corner of Figure 7 seemed more prone to failure. Cartesian distance (calculated using the static covariates, longitude and latitude) thus seems to be a suitable determinant for the datasharing term, w_{ij} , in NPP .

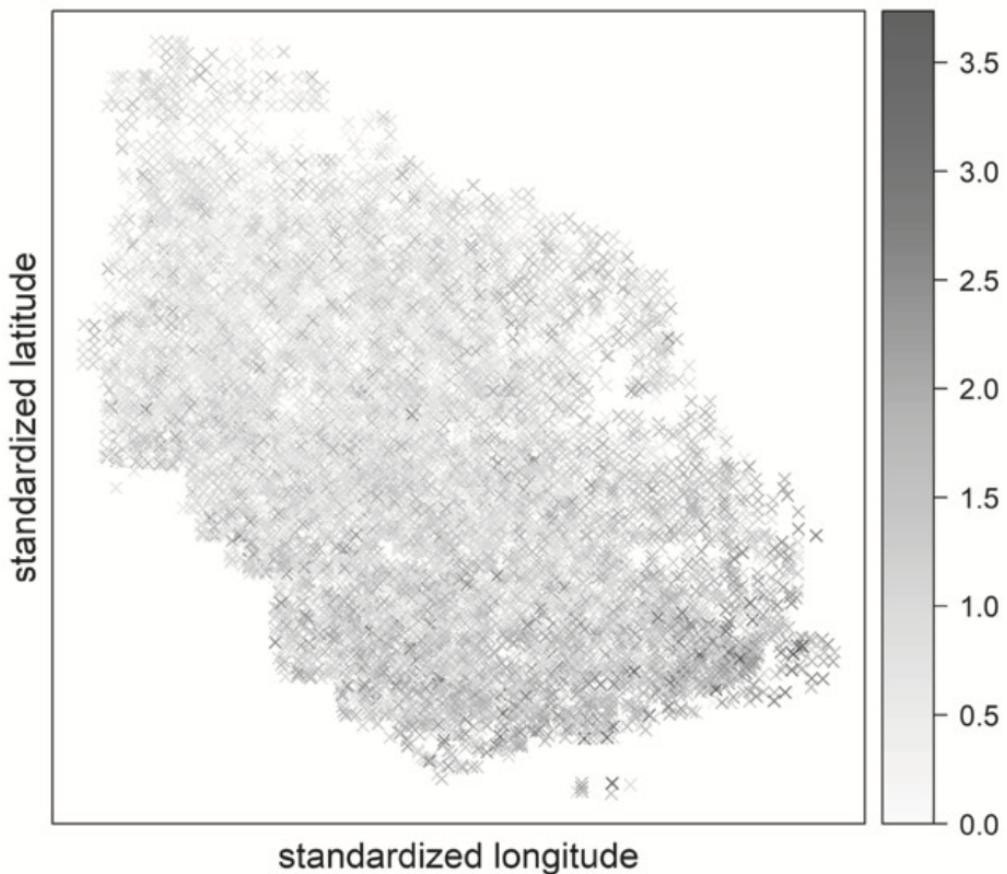


Figure 9: Node Coordinates, Av. Failures/Year As Colour (Liu and Pan 2020).

5.3 Modeling

Two *NPP* models (formulated below as (I) and (II)) were employed in order to leverage summary statistic trends. Only their w_{ij} terms differ. Consider the y_i arrival process times at site $i \in \mathcal{V}$ observed until time T , $\mathbf{t}_i = [t_{i1}, \dots, t_{iy_i}]$. Poisson rate parameters $\lambda_i \in R$ are fit using their negative log likelihood functions $-\ell_{1i}$ and *network lasso* with edges \mathcal{E} :

$$-\ell_{1i}(\lambda_i | \mathbf{t}_i) = -\sum_{k=1}^{y_i} \log(\lambda_i) + \lambda_i T$$

$$\text{minimize}_{i \in \mathcal{V}} -\ell_{1i}(\lambda_i) + \mu \sum_{(j,k) \in \mathcal{E}} w_{jk} \|\lambda_j - \lambda_k\|_2$$

The main complexity in modeling here comes from the different weight terms $w_{ij}^{(I)}$ and $w_{ij}^{(II)}$. To define them, let $\mathbf{l}_i = [x_i, y_i]$ be a 2-dimensional location vector for node i , where $x_i, y_i \in [0, 1]$ represent normalized coordinates scaled between 0 and 1. $w_{ij}^{(I)} = \sqrt{2} - \|\mathbf{l}_i - \mathbf{l}_j\|_2$ and $w_{ij}^{(II)} = 1$. The specific form of $w_{ij}^{(I)}$ ensures that nodes which are closer together physically have a larger edge objective. $\sqrt{2}$ is the furthest apart any 2 nodes can be given their scaling. (I) and (II) thus represent different points on the data-sharing trade off. Estimates in (I) can be thought of as 'receiving more relevant data' given our assumption that different geographical regions of the network are governed by different underlying processes. Estimates in (II) 'receive more data overall', however, because the information from far away nodes is not scaled down in the edge objective. Other static covariates could hypothetically be used to determine w_{ij} .

5.4 Forecasting

To obtain *NPP* performance metrics, the US oil and gas well dataset was split into 3 equally sized parts by time. The first two data packets were used for parameter estimation and regularisation paths respectively. The final third was left as unseen test data. Once parameter estimates $\hat{\lambda}_i$, $i \in \mathcal{V}$ are calculated at optimal step $\hat{\mu}$ of the regularisation path, the aforementioned model can be used to predict the number of failures at node i , $y_{\text{test},i}$, over time period $[a, b]$ in the test set:

$$y_{\text{test},i}^\lambda = (a - b)\hat{\lambda}_i$$

Two naive models, α and β , were also used to predict $y_{\text{test},i}$. They simply use $y_{\text{train},i}$, the number of failures in the data's first 2/3 at node i . Treating the entire network as one homogenous Poisson point process gives the α estimate which is the same across nodes. β estimates can be made by fitting a homogeneous Poisson process at each node individually. α and β thus represent opposite ends of the data sharing spectrum.

$$y_{\text{test},i}^\alpha = \frac{1}{2|\mathcal{V}|} \sum_{i \in \mathcal{V}} y_{\text{train},i}$$

$$y_{\text{test},i}^\beta = y_{\text{train},i}/2$$

5.5 Forecasting Performance Metrics

Two metrics were used to evaluate the performance of models (I), (II), α and β . The first is the simple TSE measure given by $TSE = \sum_{i \in \mathcal{V}} (y_{test,i} - \hat{y}_{test,i})^2$. The second is the $C\text{-Index}$ (Liu and Pan 2020) which is calculated using the entire test dataset a.k.a 'out-of-bag sample' (OOB):

1. **Pair Formation:** Construct all possible pairs of systems from the OOB sample. The total number of unique system pairs is $\binom{|\mathcal{V}|}{2}$.
2. **Observed Ranking:** For each system pair j , determine their observed 'ranking' based on which experienced more failures in the OOB sample.
3. **Predicted Ranking:** For each system pair j , determine their predicted 'ranking' based on which was expected to experience more failures in the OOB sample.
4. **Concordance Check:** For each pair j , compare the predicted and observed rankings. If the rankings agree, set $C_j = 1$; otherwise, set $C_j = 0$.
5. **Calculation:** The $C\text{-Index}$ for the OOB sample is then computed as:

$$C\text{-Index} = \frac{1}{\binom{|\mathcal{V}|}{2}} \sum_j C_j \in [0.5, 1]$$

The $C\text{-Index}$ is a sum over $\binom{|\mathcal{V}|}{2}$ binary values whereas TSE is a sum over $|\mathcal{V}|$ non-binary values. The $C\text{-Index}$ is thus typically a less variable performance metric than TSE . However, it only measures the performance of node estimates relative to each other rather than relative to the observed values $y_{train,i}$ themselves. This is converse to TSE making them a good pair for evaluation. Given that these models are time independent, 3-fold cross validation can be carried out using each portion of the data to calculate performance metrics and estimate their errors.

5.6 Forecasting Results

The validity of *NPP* with homogenous Poisson process and static covariates is supported by the fact that model (II) produced the highest *C-Index* score of 0.56. Estimated errors demonstrate the statistical significance of this result. Furthermore, Liu and Pan analysed this dataset (2020) using a tree based method and obtained a *C-Index* score of 0.54. However, they used a more computationally expensive cross-validation scheme where 50% of data (on average) is used in model fitting rather than this paper's 66%. Exact comparison between our results is thus difficult. Model (II) also achieves the lowest *TSE* score (closely followed by (I)), however these results are less clear given their estimated errors.

Model	Formula	<i>C-Index</i>	$\overbrace{SD(C\text{-}Index)}^{\text{Estimate}}$	<i>TSE</i>	$\overbrace{SD(TSE)}^{\text{Estimate}}$
(I)	$w_{ij}^{(I)} = \sqrt{2} - \ \mathbf{l}_i - \mathbf{l}_j\ _2$	0.5251	0.0079	9214.73	33.13
(II)	$w_{ij}^{(II)} = 1$	0.56395	0.00015	9138.29	111.67
α	$y_{\text{test},i}^\alpha = \frac{1}{2 \mathcal{V} } \sum_{i \in \mathcal{V}} y_{\text{train},i}$	0.5	0	9223.78	45.43
β	$y_{\text{test},i}^\beta = y_{\text{train},i}/2$	0.5588	0.00095	12624.00	114.55

Table 1: Oil and Gas Well Dataset Forecasting Performance Metrics with Highlighted Strongest Model (II).

5.7 Inference On Existing Nodes

A number of statistics suggest that these US oil and gas wells' underlying failure processes are fairly homogenous:

- The *NPP* models' regularisation paths (Figures 10-11) don't exhibit clear vertices. Equivalently, $\hat{\mu} \approx \mu_{consensus}$.
- Model α has a significantly lower *TSE* than β .
- Model (II) has a higher *C-Index* and lower *TSE* than (I).
- The fitted $\hat{\lambda}$ values, arranged geographically on the next page, are similar (in colour).

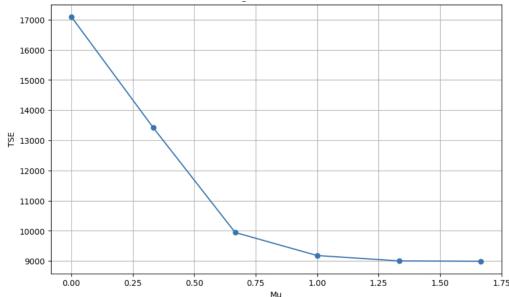


Figure 10: Model (I) Regularisation Path Example.

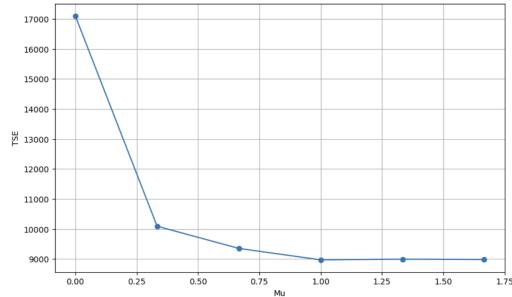


Figure 11: Model (II) Regularisation Path Example.

Keen inspection of Figures 12-15 shows the clear stratification of Model (I)-(II) $y_{\text{test},i}$ parameter estimates (in one fold of cross validation) characteristic of *network lasso*. Although they occur at very different frequencies, the only unique estimates among the 8232 nodes from model (II) are 0.04267273, 0.06144288, 0.55420295, 1.01634044, 1.02232695, 1.0266068, 1.05939645, 1.14507862 and 1.19725866. Even though (II) had a constant $w_{ij}^{(II)} = 1$ and (I) had the geographical $w_{ij}^{(I)} = \sqrt{2} - \|\mathbf{l}_i - \mathbf{l}_j\|_2$, (II) was still able to capture the dataset's spatial trend. This is shown through the faint yellow dots prevalent in the bottom-right hand corner of Figure 13.

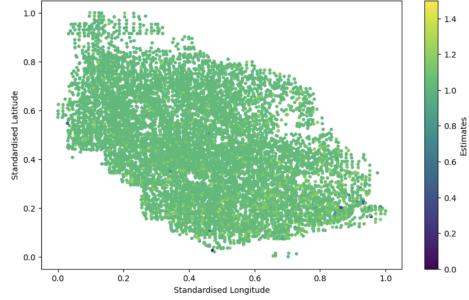


Figure 12: Colour Coded Model (I) Estimates Arranged Geographically.

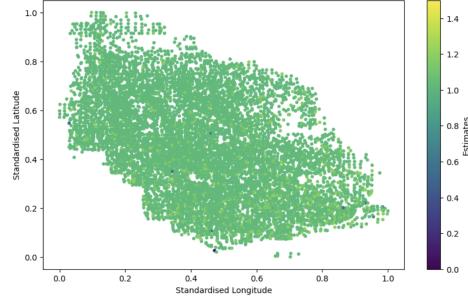


Figure 13: Colour Coded Model (II) Estimates Arranged Geographically.

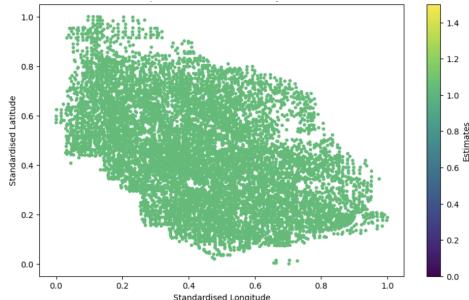


Figure 14: Colour Coded Model α Estimates Arranged Geographically.

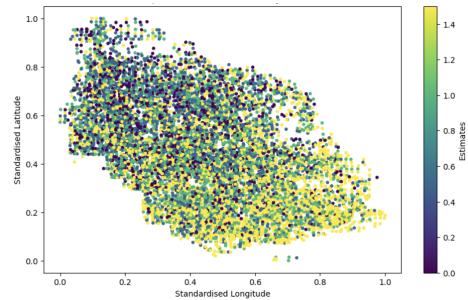


Figure 15: Colour Coded Model β Estimates Arranged Geographically.

5.8 Inference On New Nodes

Once NPP models are fit at each node, it's possible to infer the recurrent event processes of new nodes based on their pre-existing neighbors. Let $\hat{\lambda}_i \in \mathcal{V}$ be a fitted rate parameter from the regularization path's optimal step. Inference on new nodes λ_j can be carried out by solving the Weber problem:

$$\text{minimise} \sum_{i \in \mathcal{V}} w_{ij} \left\| \lambda_j - \hat{\lambda}_i \right\|_2$$

Here, estimates of λ_j can be thought of as the weighted median of their neighbors. The Weber problem can be readily computed, even for large problems (Hallac, Leskovec and Boyd, 2015). To test the accuracy of Weber problem solutions, the aforementioned testing procedure was carried out on 3 random data splits made at the node level: 2/3 were assumed to be pre-existing in the training dataset and the models of the remaining 1/3 were fit by solving the Weber problem before being tested on an OOB sample.

5.9 Inference On New Nodes: Results

The fact that Weber problem solutions from model (I) give the highest *C-Index* score seen thus far (0.575) demonstrates their usefulness. New nodes' inferred models are able to capture the dataset's spatial trend as shown in Figure 16. However, this is somewhat offset by (I)'s *TSE* values which are indistinguishable from naive model α 's (fit by considering the mean number of existing node failures in training). β is not applicable to this problem and (II) is omitted because its inferred solutions for all new nodes would be the same - $w_{ij}^{(II)} = 1$.

Model	Formula	<i>C-Index</i>	$\overbrace{SD(C\text{-}Index)}$ Estimate	TSE	$\overbrace{SD(TSE)}$ Estimate
(I)	$w_{ij}^{(I)} = \sqrt{2} - \ \mathbf{l}_i - \mathbf{l}_j\ _2$	0.5752	0.0036	2979.40	48.62
α	$y_{\text{test},i}^\alpha = \frac{1}{2 \mathcal{V} } \sum_{i \in \mathcal{V}} y_{\text{train},i}$	0.5	0	2978.25	47.95

Table 2: Oil and Gas Well Dataset Inference Performance Metrics with Highlighted Strongest Model (I).

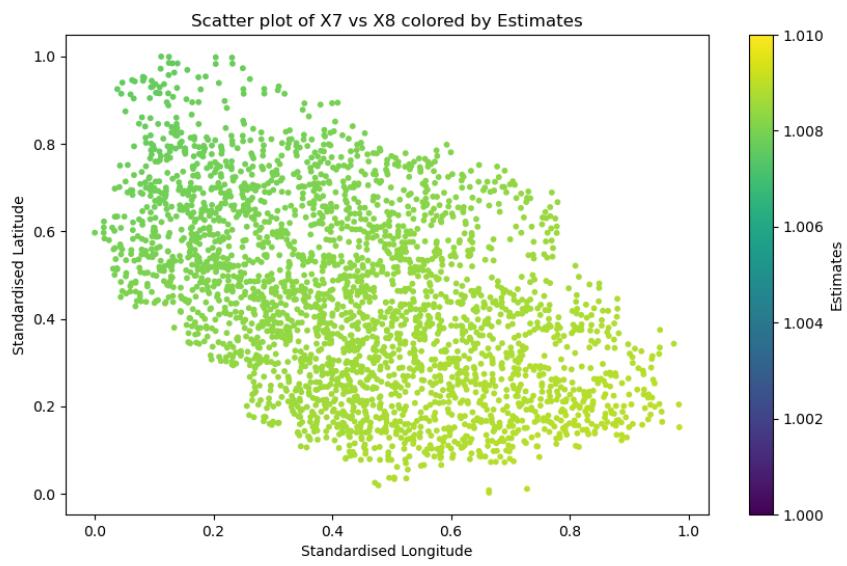


Figure 16: Weber Solutions' Estimates On OOB Sample .

6 Network Homogenous Poisson Process + Static Covariates Case Study: Scotland Pipe Dataset

6.1 Dataset Description (Xiao, Ye and Revie, 2024)

The dataset contains over 300,000 pipe segments and 230,000 failure records from a major Scottish water utility, spanning 16 years. Each pipe is annotated with attributes such as material, diameter, length, installation year, and geographic location. Failures are time-stamped and linked to specific pipe IDs, enabling the reconstruction of temporal and spatial failure cascades. The network is modeled as a graph with nodes representing junctions or endpoints and edges as pipes, supporting both topological and temporal analysis of cascading events. This dataset will be used to further demonstrate the validity of 'Network Homogeneous Poisson Process + Static Covariates' using different w_{ij} terms. 3 iterations of OOB sampling (with equal third splits between fitting, regularisation path and testing data) will once again be used to calculate performance metrics. However, due to this dataset's size, different (random) selections of 10,000 nodes will be used in each iteration rather than the aforementioned cross validation approach. Trustworthy results (based on estimated standard errors) are obtained.

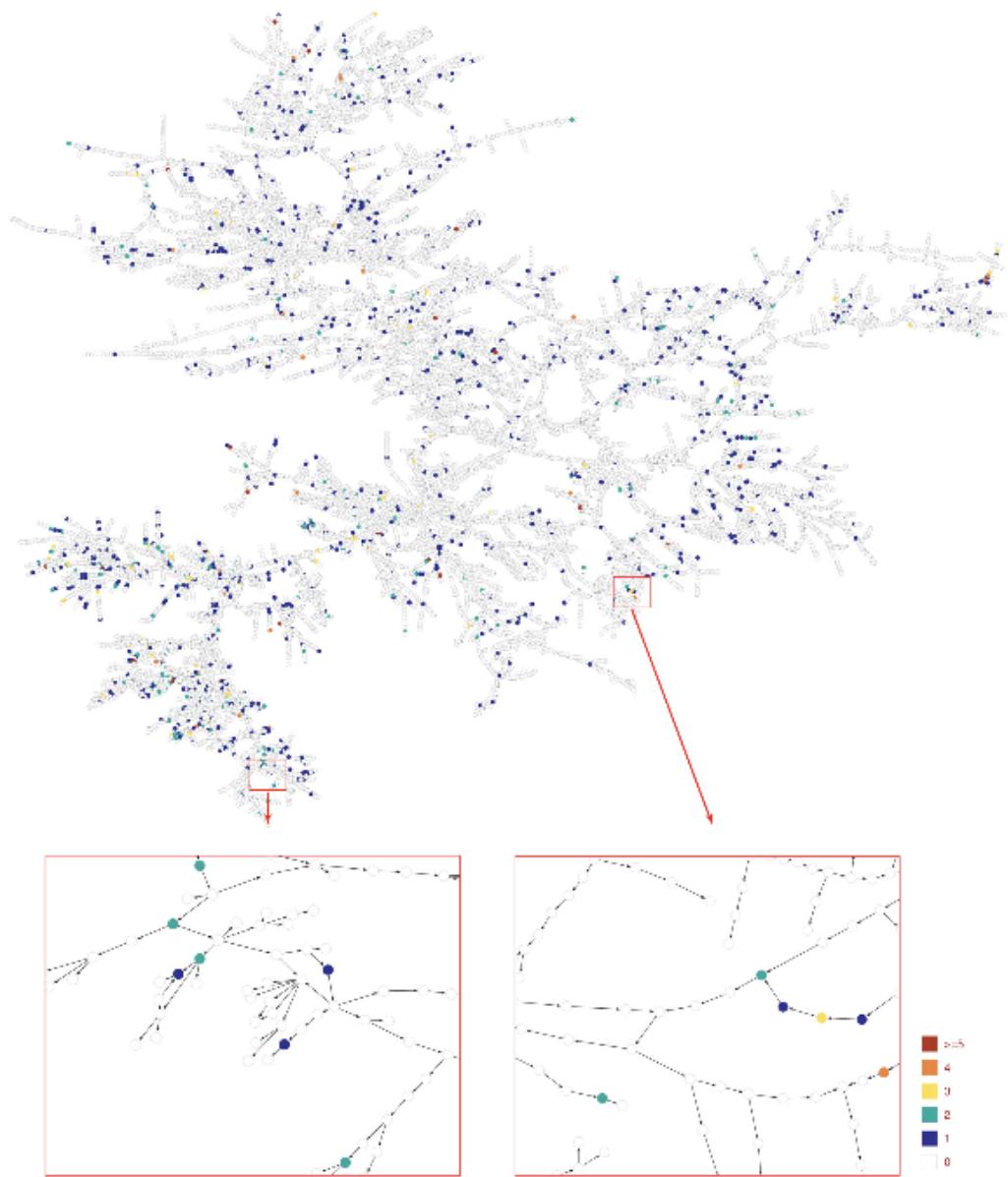


Figure 17: Scotland Pipe Dataset (Zhai, Ye, Li, Revie and Dunson, 2024).

6.2 Summary Statistics

Similar homogenous Poisson assumptions to that of the US oil and gas well dataset are made here - It is assumed that the frailty of pipes is time invariant. However, the Scotland pipe dataset has a different static covariate structure. Two are of particular interest - 'WOA-REF' (nominal area reference data) and 'ARM-CONNECTIONS' (discrete data denoting the number of pipes joining a particular node). As shown in Figures 18-19, they seem associated with inter-arrival times averaged within and across sites. Specifically, sites with less than 10 'ARM-CONNECTIONS' seem to follow a different process to those with more than 10. Therefore, model (III) is introduced below. The aforementioned model (II) will also be used as before.

$$\begin{aligned} w_{ij}^{(III)} &= \mathbf{1} \{ \text{WOA-REF}_i = \text{WOA-REF}_j \} \\ &\quad + \mathbf{1} \{ \text{ARM-CONNECTIONS}_i = \text{ARM-CONNECTIONS}_j \} \end{aligned}$$

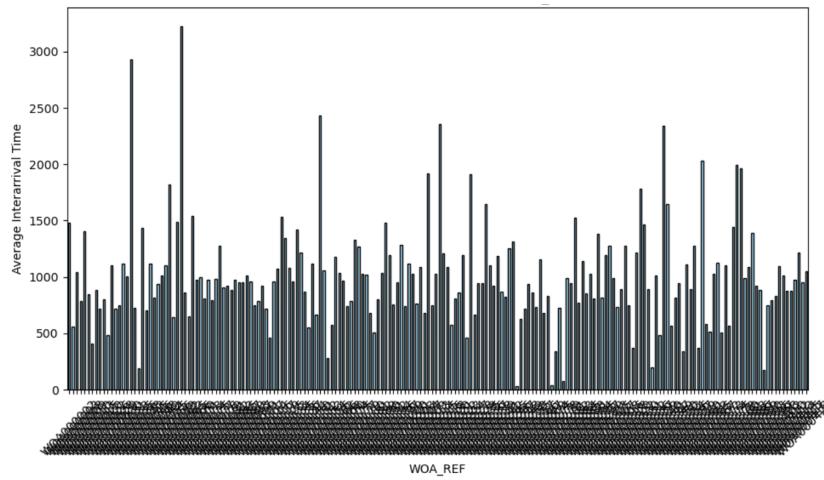


Figure 18: Average Interarrival Time vs WOA-RERERENCE.

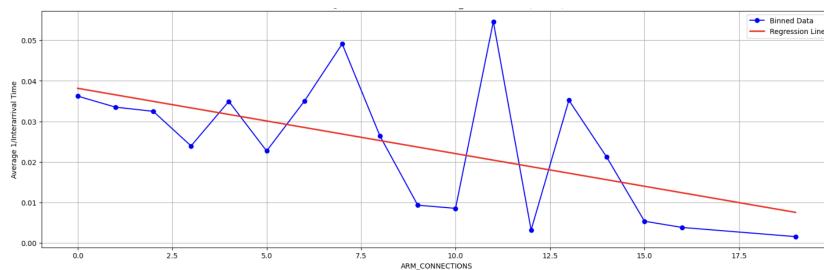


Figure 19: Average 1/Interarrival Time vs ARM-CONNECTIONS.

6.3 Forecasting Results

The validity of *NPP* with homogenous Poisson process and static covariates is supported by the fact that models (II) and (III) have significantly higher *C-Index* scores than the naive models. Their *TSE* scores are also significantly lower than that of α and β .

Model	Weighting Formula	<i>C-Index</i>	$\overbrace{SD(C-Index)}^{\text{Estimate}}$	<i>TSE</i>	$\overbrace{SD(TSE)}^{\text{Estimate}}$
(II)	$w_{ij}^{(II)} = 1$	0.6989	0.0017	5920.48	192.75
(III)	$w_{ij}^{(III)}\{\text{ARM-CONNECTIONS, WOA-REFERENCE}\}$	0.6342	0.0011	5902.81	210.00
α	$y_{\text{test},i}^{\alpha} = \frac{1}{2 \mathcal{V} } \sum_{i \in \mathcal{V}} y_{\text{train},i}$	0.5	0	10247.33	100.50
β	$y_{\text{test},i}^{\beta} = y_{\text{train},i}/2$	0.5387	0.0014	10732.92	234.37

Table 3: Scotland Pipe Dataset Forecasting Performance Metrics with Highlighted Strongest Model (II).

6.4 Inference

As with the first case study, model (II) outperforms the static covariate focussed *NPP* model (III) in terms of *C-Index*. This suggests that the value added by nodes receiving data from all members of the network outweighs any value added by focusing on static covariates. Interestingly, the *NPP* models ((II) and (III)) perform much better than the naive models (α and β) while this difference is less pronounced in the US oil well dataset. By way of explanation, consider the core goal of *NPP* - finding an optimal amount of data sharing between nodes as governed by the μ parameter which balances the edge objective and node objective from (2). Naive models β and α sit on opposite ends of this spectrum and could be mirrored by a *NPP* with $\mu = 0$ and $\mu = \mu_{consensus}$ respectively. The previous dataset saw α perform much better than β , suggesting that the underlying Poisson processes at each node should be fairly similar. In other words, the value added by *NPP* was more limited because there was less value added by giving weight to the individual eccentricities at each node in training. Conversely, In the Scotland pipe dataset, α and β perform similarly (in terms of *TSE*). *NPP* is thus able to balance the validity of both naive models by choosing an optimal $\hat{\mu}$ from the middle of 0 and $\mu_{consensus}$ which, in turn, lie at opposite sides of example regularization paths: Figures 20-21.

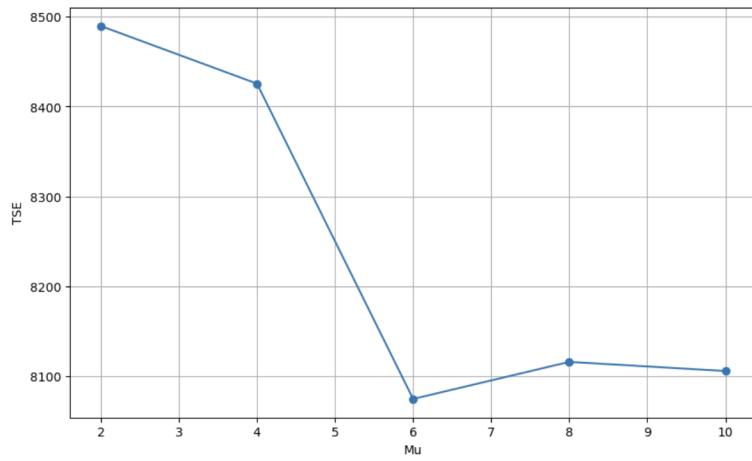


Figure 20: Model (II) Regularisation Path Example.

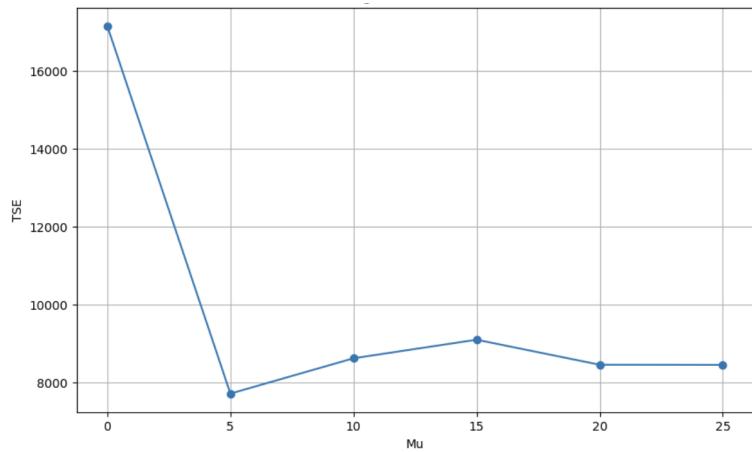


Figure 21: Model (III) Regularisation Path Example.

7 Network $NHPP$ + Dynamic Covariates

7.1 Further Model Formulation

Recall, from section 3.1, that at any given time t , we can observe c values $\mathbf{d}_{it} = [d_{i1t}, \dots, d_{ict}]$ at each site i . Therefore, up to $c+2$ coefficients may be fit at each node such that: $\mathbf{x}_i = [x_{i0}, x_{i1}, \dots, x_{ic}, x_{it}] \in R^{c+2}$. The model can be written as:

$$\lambda_i(t, \mathbf{d}_{it} | \mathbf{x}_i) = \exp \left(x_{i0} + \sum_{j=1}^c x_{ij} d_{ijt} + x_{it} t \right) \quad (0)$$

To fit such a model, the negative log-likelihood function for a given rate λ_i is used:

$$-\ell_i(\mathbf{x}_i) = - \sum_{k=1}^{y_i} \log(\lambda_i(t_k)) + \int_0^T \lambda_i(t) dt \quad (1)$$

Practically speaking, evaluating the integral in (1) may be difficult if d_{ijt} is stochastic. For example, the US oil and gas well dataset recorded dynamic covariates such as site pressure at fixed points (non-integrable) over time. We thus assume that dynamic covariate readings are observed at each site through the filtration $\{\mathcal{F}_t\}_{t=0}^T$. This can be viewed as the realization of an underlying, 'true' model: For each variable j observed at site i , let $d_{ijt} = g_{ijt} + \epsilon_{ijt}$ such that $\{\mathcal{F}_t\}_{t=0}^T$ is the sum of a deterministic function and noise. After determining a suitable g function, the negative log-likelihood function for a given rate λ_i can be calculated as usual by ignoring ϵ terms. Static covariates can be included in this framework as before.

8 Network NHPP + Dynamic Covariates Case Study: US Oil And Gas Well Dataset

8.1 Summary Statistics

This dataset includes real-time sensor readings at fixed points for each node scaled between $[0, 1]$. They are examples of dynamic covariates which would have to be preliminarily fit in the form $d_{ijt} = g_{ijt} + \epsilon_{ijt}$ before they can be incorporated into *NPP*. On the other hand, 'Time since last failure' at each site is a deterministic, dynamic covariate meaning that it can be included in (1) more easily. Its inclusion allows for the possibility that nodes' recurrent events are self exciting process (somewhat contradicting the base assumptions of a Poisson process).

Cumulative event plots for recurrent event data usually include time on the x-axis (for example to determine whether or not a Poisson process is homogeneous, as shown previously). However, similar plots can be made with dynamic covariates on the x-axis. Figures 22-23 thus provide heuristics for the effect of 2 dynamic covariates on site frailty by treating the network as a point process (and ignoring system heterogeneity). Although the plots are far from empirical, they were used to guide practical efforts made under time constraints: 'Time since last failure' seemed somewhat related to cumulative no. events meaning it was carried forwards rather than any of the stochastic dynamic covariates which had similar plots to 'Dynamic Torque' below.

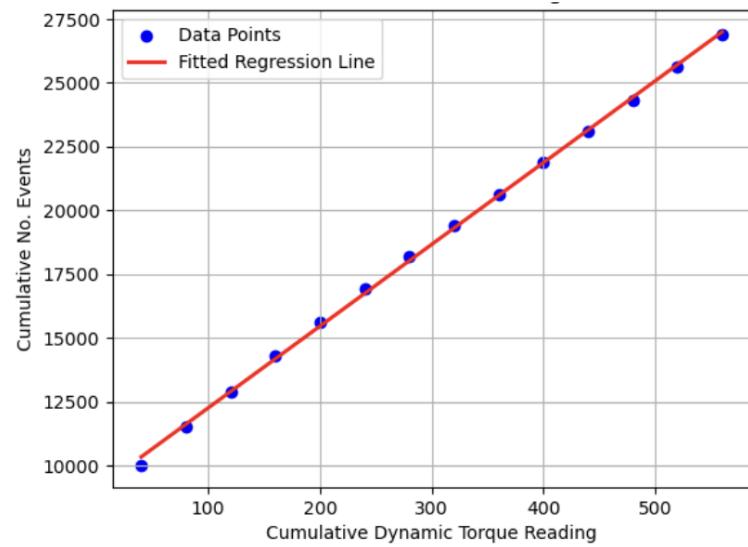


Figure 22: Cumulative No. Events Vs Cumulative Dynamic Torque Reading.

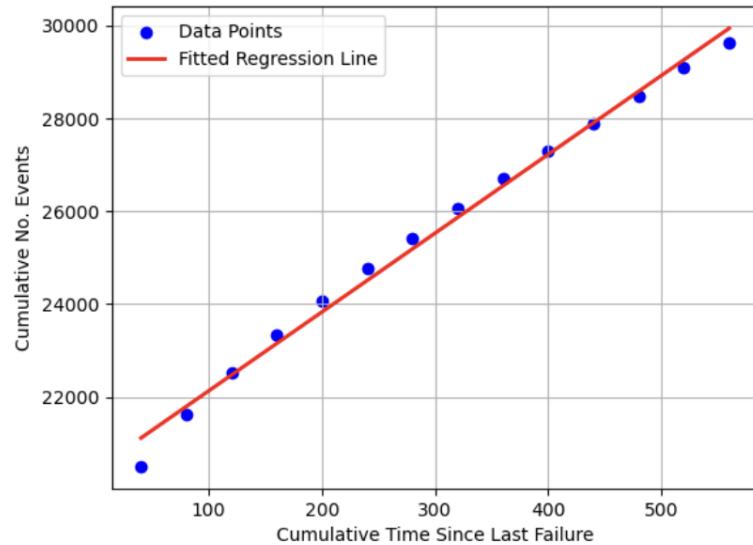


Figure 23: Cumulative No. Events Vs Cumulative Time Since Last Failure.

8.2 Modeling

Two 'Network NHPP + dynamic covariates' models were employed to try and capture the effect of 'time since last failure' and NHPP (time varying frailty). Their w_{ij} terms were simply set to 1 given the strong performance of model (II) in previous sections. Both give rise to the below problem with $\mathbf{x}_i \in R^3$ as model parameters. $t_{it, \text{last}}$ represents the time of site i 's most recent breakdown at time $t \geq 0$.

$$\lambda_i(t|\mathbf{x}_i) = \exp(x_{i0} + x_{i1}t_{i,\text{last}} + x_{it}t)$$

$$-\ell_{2i}(\mathbf{x}_i) = -\sum_{k=1}^{y_i} \log(\lambda_i(t_{ik}|\mathbf{x}_i)) + \int_0^T \lambda_i(t|\mathbf{x}_i) dt$$

$$\text{minimise} \sum_{i \in \mathcal{V}} -\ell_{2i}(\mathbf{x}_i) + \mu \sum_{(j,k) \in \mathcal{E}} w_{jk} \|\mathbf{x}_j - \mathbf{x}_k\|_2$$

Model (IV) has the exact form of above. Model (V) fixes all x_{i1} and x_{it} , only estimating x_{i0} . The fixed values x_{i1} and x_{it} in (V) are determined by simply treating the network as a point process, fitting a poisson regression and then adjusting for the number of nodes in the network. For example, Figure 6 shows why x_{it} was set to 0 in (V), treating the Poisson process as time invariant. To forecast the number of events in a test period using (IV) or (V), one must integrate their rate function over future time values. However, this calculation is difficult when using $t_{it, \text{last}}$ because its future value is stochastic. See the appendix for an exact forecasting method to calculate expected number of events observed in an OOB sample using (IV) or (V).

8.3 Modeling Complexities

8.3.1 Complexity 1: Unstable $\mu = 0$ Parameter Estimates

Fitting *NPP* begins with $\mu = 0$ in (2) at the first regularisation path step. This ignores the node objective and fits based on (1) at each node. Fitting a Poisson process with 3 free parameters (such as (IV)) based on only a few recurrent events over the training period (average no. failures ≈ 1 in training for both case study datasets) leads to unstable estimates. This problem also occurred with (V). For example, the Scotland pipe dataset's $\mu = 0$ parameter estimates have a *C-Index* ≈ 0.5 . This suggests that the node-wise parameter estimates contained almost no useful information. This problem was helped by adding a ridge term, lasso term or enforcing restrictions on the parameter estimates made by solvers (Hallac, Leskovec and Boyd, 2015). The US oil and gas well data 'Network NHPP + dynamic covariates' models were thus salvageable, however the Scotland pipe dataset's were not (even with these fixes).

8.3.2 Complexity 2: Exponential Function Asymmetry

$$\lambda_i(t|\mathbf{x}_i) = \exp(x_{i0} + x_{i1}t_{i,last} + x_{it}t)$$

Consider the $\mu = 0$ parameter estimates for nodes with 0 and 3 observed events in training data. Ignoring the dynamic covariates, for arguments sake, the fitted models would be $\hat{\lambda}_0 = \exp(-\infty)$ and $\hat{\lambda}_3 = \exp(\log(3))$ respectively

- Relatively similar observed frailties have given way to infinitely different x_0 estimates of $-\infty$ and $\log(3)$. At future steps of the regularization path, when the edge objective comes into effect, parameter estimates pull towards each other based on the ℓ_2 -norm penalty over their difference. All parameter estimates of the network will get (unreasonably) pulled towards $-\infty$ by the norm given its weight. This is an extreme example used to demonstrate a more general asymmetry issue. One may thus question the premise of using an exponential rate function in *NPP*, however this is required to ensure non negativity in recurrent event counts. Below are 2 heuristic solutions that were used to obtain results for models (IV) and (V):

1. Changing the scale of the data's time dimension (e.g. from days to weeks) to ensure that the exponential function's domain is in a neighbourhood of 0 (where it behaves like a linear function).
2. Changing the *network lasso* objective to account for this asymmetry using an alternative measure $\tilde{\mathbf{x}}_i = \exp(\mathbf{x}_i)$:

$$\text{minimize}_{\mathbf{x}} \sum_{i \in \mathcal{V}} -\ell_{2i}(\mathbf{x}_i) + \mu \sum_{(j,k) \in \mathcal{E}} w_{jk} \|\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_k\|_2$$

8.4 Results

The poor *C-Index* performance of models (IV) and (V) reflect poorly on 'Network NHPP + dynamic covariates'. It seems as if 'complexity 1' from above, in particular, leads to a high degree of instability in parameter estimates. To offset this, and justify the inclusion of dynamic covariates in *NPP*, one would need them to be highly associated with frailty. This trend is not observed in Figures 22 and 23. Nonetheless, the respectable *TSE* scores of (IV) and (V) suggest that they may be somewhat valid.

	(I)	(II)	(IV)	(V)	α	β
<i>C-Index</i>	0.5251	0.56395	0.5036	0.5125	0.5	0.5588
$\overbrace{SD(C\text{-}Index)}^{\text{Estimate}}$	0.0079	0.00015	0.0098	0.0095	0	0.00095
<i>TSE</i>	9214.73	9138.29	9164.86	9227.35	9223.78	12624.00
$\overbrace{SD(TSE)}^{\text{Estimate}}$	33.13	111.67	120.45	119.88	45.43	114.55

Table 4: Oil and Gas Well Dataset Forecasting Performance Metrics with Newly Included (IV) and (V).

9 Conclusion

NPP is capable of modeling recurrent events on large network datasets. Such models can then be used to forecast future events and infer details about networks (such as geographical trends). It seems to be effective in capturing static covariate information and trading off node models' heterogeneity/homogeneity. However, further work is required to determine whether there are practical scenarios in which *NPP* can effectively capture dynamic covariate information.

10 Appendix

10.1 Convexity of the Negative Log-Likelihood (1)

Let the intensity function at node i be defined as:

$$\lambda_i(t, \mathbf{d}_{it} | \mathbf{x}_i) = \exp \left(x_{i0} + \sum_{j=1}^c x_{ij} d_{ijt} + x_{it} t \right)$$

where $\mathbf{d}_{it} = [d_{i1t}, \dots, d_{ict}]$ is the value of known dynamic covariate functions at time t . The negative log-likelihood over observation period $[0, T]$ with y_i arrival process times $\mathbf{t}_i = [t_{i1}, \dots, t_{iy_i}]$ is:

$$-\ell_i(\mathbf{x}_i) = -\sum_{k=1}^{y_i} \log(\lambda_i(t_{ik})) + \int_0^T \lambda_i(t) dt \quad (1)$$

Substituting λ_i , we obtain:

$$-\ell_i = -\sum_{k=1}^{y_i} \left(x_{i0} + \sum_{j=1}^c x_{ij} d_{ijk} + x_{it} t_{ik} \right) + \int_0^T \exp \left(x_{i0} + \sum_{j=1}^c x_{ij} d_{ijt} + x_{it} t \right) dt.$$

To establish the convexity of the integral term in the expression for $-\ell_i$, consider the function

$$f_i(\mathbf{x}_i) = \int_0^T \exp \left(x_{i0} + \sum_{j=1}^c x_{ij} d_{ijt} + x_{it} t \right) dt,$$

Let us define the integrand as:

$$g_i(t; \mathbf{x}_i) = \exp(\langle \mathbf{x}_i, \phi_i(t) \rangle),$$

where $\phi_i(t) = (1, d_{i1t}, \dots, d_{ict}, t)$ is a vector of known functions of t , and $\langle \cdot, \cdot \rangle$ denotes the standard inner product. The gradient of g_i with respect to \mathbf{x}_i is

$$\nabla_{\mathbf{x}_i} g_i(t; \mathbf{x}_i) = \phi_i(t) \exp(\langle \mathbf{x}_i, \phi_i(t) \rangle),$$

and the Hessian is

$$\nabla_{\mathbf{x}_i}^2 g_i(t; \mathbf{x}_i) = \phi_i(t) \phi_i(t)^\top \exp(\langle \mathbf{x}_i, \phi_i(t) \rangle).$$

This is a positive semidefinite matrix for all t , since it is the outer product of a vector with itself scaled by a surely positive scalar given by the exponential function. Therefore, $g_i(t; \mathbf{x}_i)$ is convex in \mathbf{x}_i for each fixed t . Since integration preserves convexity when the domain of integration is independent of the variables (Boyd Vandenberghe, *Convex Optimization*, Section 3.2.2), the function

$$f(\mathbf{x}_i) = \int_0^T g_i(t; \mathbf{x}_i) dt$$

is convex in \mathbf{x}_i . Consequently, the entire expression for $-\ell_i$ is convex in the variables \mathbf{x}_i because it is the sum of linear and convex functions.

10.2 Why the L2 Norm of Differences Leads to Stratification

In the *network lasso* framework (Hallac, Leskovec and Boyd, 2015), the objective function includes a regularization term that penalizes the pairwise differences between parameter vectors across a graph:

$$\min_{\{\theta_i\}} \sum_{i \in V} f_i(\theta_i) + \lambda \sum_{(i,j) \in E} w_{ij} \|\theta_i - \theta_j\|_2$$

While the L2 norm itself does not induce sparsity in individual parameters, the L2 norm of differences between parameters can lead to *stratification*, where multiple parameter vectors become exactly equal. This effect arises from the subdifferential structure of the L2 norm at zero.

Consider the regularization term $\|\theta_i - \theta_j\|_2$. This function is non-differentiable at $\theta_i = \theta_j$, and its subdifferential at that point is the unit ball:

$$\partial \|\theta_i - \theta_j\|_2 \Big|_{\theta_i = \theta_j} = \{u \in R^d : \|u\|_2 \leq 1\}$$

This means that when $\theta_i = \theta_j$, the gradient of the regularization term is not uniquely defined, and any vector within the unit ball is a valid subgradient. As a result, the optimizer can satisfy the first-order optimality condition without needing to separate θ_i and θ_j , provided the loss terms f_i and f_j are not too dissimilar.

The first-order optimality condition for θ_i becomes:

$$0 \in \nabla f_i(\theta_i) + \lambda \sum_{j \in \mathcal{N}(i)} w_{ij} \cdot \partial \|\theta_i - \theta_j\|_2$$

If $\theta_i = \theta_j$ for some j , then the subgradient term contributes a ball of radius w_{ij} centered at zero. This gives the optimizer flexibility to "stop" at equality, thereby reducing the regularization cost to zero. Consequently, the model encourages exact equality between parameters when the loss functions are compatible, leading to piecewise constant solutions across the graph.

This mechanism underlies the clustering behavior observed in *network lasso*: groups of nodes with similar local objectives are encouraged to share identical parameter vectors, forming stratified clusters.

10.3 ADMM Derivation for Network Lasso

We consider a graph $G = (V, E)$, where each node $i \in V$ has a variable $x_i \in R^d$, and a convex function $f_i(x_i)$. The optimization problem is:

$$\min_{\{x_i\}} \sum_{i \in V} f_i(x_i) + \lambda \sum_{(i,j) \in E} w_{ij} \|x_i - x_j\|_2$$

To apply ADMM, we introduce auxiliary variables $z_{ij}, z_{ji} \in R^d$ for each edge $(i, j) \in E$, and rewrite the problem as:

$$\text{minimise} \sum_{i \in V} f_i(x_i) + \lambda \sum_{(i,j) \in E} w_{ij} \|z_{ij} - z_{ji}\|_2$$

subject to:

$$x_i = z_{ij}, \quad x_j = z_{ji} \quad \forall (i, j) \in E$$

We define dual variables $u_{ij}, u_{ji} \in R^d$ for the constraints. The augmented Lagrangian is:

$$\mathcal{L}_\rho = \sum_{i \in V} f_i(x_i) + \lambda \sum_{(i,j) \in E} w_{ij} \|z_{ij} - z_{ji}\|_2 + \sum_{(i,j) \in E} \left[\frac{\rho}{2} \|x_i - z_{ij} + u_{ij}\|_2^2 + \frac{\rho}{2} \|x_j - z_{ji} + u_{ji}\|_2^2 \right]$$

The ADMM updates proceed as follows:

- 1. x -update:** For each node i , solve:

$$x_i^{k+1} := \arg \min_{x_i} \left(f_i(x_i) + \frac{\rho}{2} \sum_{j \in \mathcal{N}(i)} \|x_i - z_{ij}^k + u_{ij}^k\|_2^2 \right)$$

2. z -update: For each edge (i, j) , solve:

$$(z_{ij}^{k+1}, z_{ji}^{k+1}) := \arg \min_{z_{ij}, z_{ji}} \left(\lambda w_{ij} \|z_{ij} - z_{ji}\|_2 + \frac{\rho}{2} \|x_i^{k+1} - z_{ij} + u_{ij}^k\|_2^2 + \frac{\rho}{2} \|x_j^{k+1} - z_{ji} + u_{ji}^k\|_2^2 \right)$$

Let:

$$a := x_i^{k+1} + u_{ij}^k, \quad b := x_j^{k+1} + u_{ji}^k$$

$$\min_{z_{ij}, z_{ji}} \lambda w_{ij} \|z_{ij} - z_{ji}\|_2 + \frac{\rho}{2} \|z_{ij} - a\|_2^2 + \frac{\rho}{2} \|z_{ji} - b\|_2^2$$

where:

$$a = x_i^{k+1} + u_{ij}^k, \quad b = x_j^{k+1} + u_{ji}^k$$

We define:

$$\theta := \frac{a+b}{2}, \quad \delta := \frac{a-b}{2}$$

We now change variables to:

$$z_{ij} = \theta + \epsilon, \quad z_{ji} = \theta - \epsilon$$

Substituting into the objective:

$$\lambda w_{ij} \|2\epsilon\|_2 + \frac{\rho}{2} \|\epsilon - \delta\|_2^2 + \frac{\rho}{2} \|\epsilon + \delta\|_2^2$$

Expanding the squared terms:

$$\|\epsilon - \delta\|_2^2 + \|\epsilon + \delta\|_2^2 = 2\|\epsilon\|_2^2 + 2\|\delta\|_2^2$$

So the objective becomes:

$$2\lambda w_{ij}\|\epsilon\|_2 + \rho\|\epsilon\|_2^2 + \rho\|\delta\|_2^2$$

Dropping the constant term $\rho\|\delta\|_2^2$, we minimize:

$$\rho\|\epsilon\|_2^2 + 2\lambda w_{ij}\|\epsilon\|_2$$

This has a closed-form solution (Hallac, Leskovec and Boyd, 2015):

$$\epsilon^* = \left(1 - \frac{\lambda w_{ij}}{\rho\|\delta\|_2}\right)_+ \delta$$

Finally, we recover:

$$z_{ij}^{k+1} = \theta + \left(1 - \frac{\lambda w_{ij}}{\rho\|\delta\|_2}\right)_+ \delta, \quad z_{ji}^{k+1} = \theta - \left(1 - \frac{\lambda w_{ij}}{\rho\|\delta\|_2}\right)_+ \delta$$

where $(\cdot)_+ = \max(0, \cdot)$ is applied to the scalar shrinkage factor.

3. *u*-update: For each edge (i, j) :

$$u_{ij}^{k+1} := u_{ij}^k + x_i^{k+1} - z_{ij}^{k+1}$$

$$u_{ji}^{k+1} := u_{ji}^k + x_j^{k+1} - z_{ji}^{k+1}$$

These updates enforce the consensus constraints $x_i = z_{ij}$, $x_j = z_{ji}$, and indirectly $x_i \approx x_j$ via the regularization term.

Summary: Each ADMM iteration involves:

1. Local optimization at each node for x_i
2. Proximal update at each edge for z_{ij}, z_{ji}
3. Dual variable update for u_{ij}, u_{ji}

This structure is parallelizable and scalable for large graphs.

10.4 Rolling Expectation for a Non-Homogeneous Poisson Process with Dynamic Covariate

Let $\lambda_i(t)$ denote the intensity function of a non-homogeneous Poisson process for unit i , defined as:

$$\lambda_i(t) = \exp(x_{i0} + x_{i1}t_l + x_{it}t)$$

where:

- $x_{i0}, x_{i1}, x_{it} \in R$ are fixed covariates, $x_{it} \neq 0$ to avoid divergence in below expressions.
- t is the current time,
- t_l is the time of the most recent failure prior to t .

We aim to compute the expected number of events in the interval $[a, b]$, where $a \geq t_l$. Because the intensity depends on the time of the last event, the process must be evaluated recursively.

Step 1: Initialization

$$t_0 = a, \quad t_l^{(0)} = t_l, \quad n = 0$$

Step 2: Recursive Computation For each step n , define the intensity function over the interval $[t_n, t_{n+1}]$ as:

$$\lambda_i^{(n)}(t) = \exp\left(x_{i0} + x_{i1}t_l^{(n)} + x_{it}t\right)$$

The expected time until the next event, Δt_n , is defined implicitly by:

$$\int_{t_n}^{t_n + \Delta t_n} \lambda_i^{(n)}(s) ds = 1$$

Solving this yields:

$$\Delta t_n = \frac{1}{x_{it}} \log \left(1 + \frac{x_{it}}{\exp(x_{i0} + x_{i1}t_l^{(n)})} \cdot \exp(-x_{it}t_n) \right)$$

Then update:

$$t_{n+1} = t_n + \Delta t_n, \quad t_l^{(n+1)} = t_{n+1}$$

Repeat until $t_{n+1} > b$. Let N be the number of full events before b .

Step 3: Final Residual Expectation The residual expected number of events in the interval $[t_N, b]$ is:

$$E_{\text{residual}} = \frac{\exp(x_{i0} + x_{i1}t_l^{(N)})}{x_{it}} (\exp(x_{it}b) - \exp(x_{it}t_N))$$

Final Expected Count

$$E[N(a, b)] = N + E_{\text{residual}}$$

References

- [1] Andersen, P. K., & Gill, R. D. (1982). Cox's regression model for counting processes: A large sample study. *The Annals of Statistics*, 10(4), 1100–1120.
- [2] Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- [3] Clayton, D., & Cuzick, J. (1985). Multivariate generalizations of the proportional hazards model. *Journal of the Royal Statistical Society: Series A (General)*, 148(2), 82–117.
- [4] Cook, R. J., & Lawless, J. F. (1997). Marginal analysis of recurrent events and a terminating event. *Journal of the American Statistical Association*, 92(440), 1421–1430.
- [5] Diggle, P. J., Heagerty, P., Liang, K. Y., & Zeger, S. L. (2002). *Analysis of Longitudinal Data* (2nd ed.). Oxford University Press.
- [6] Ertekin, S., Rudin, C., & McCormick, T. H. (2015). Reactive point processes: A new approach to predicting power failures in underground electrical systems. *The Annals of Applied Statistics*, 9(1), 122–144.
- [7] Fleming, T. R., & Harrington, D. P. (1991). *Counting Processes and Survival Analysis*. Wiley.
- [8] Hanson, T., & Zhou, H. (2012). Bayesian semiparametric proportional odds models for survival data with a cure fraction. *Biometrics*, 68(1), 23–32.
- [9] Hennerfeind, A., Brezger, A., & Fahrmeir, L. (2006). Geoadditive survival models. *Journal of the American Statistical Association*, 101(475), 1065–1075.

- [10] Hong, Y., Meeker, W. Q., & Escobar, L. A. (2018). Big data and reliability applications: Challenges and opportunities. *Quality Engineering*, 30(1), 1–15.
- [11] Jin, M., et al. (2023). Large models for time series and spatio-temporal data: A survey and outlook. *arXiv preprint arXiv:2310.10196*. <https://arxiv.org/pdf/2310.10196>
- [12] Kalbfleisch, J. D., & Prentice, R. L. (2002). *The Statistical Analysis of Failure Time Data* (2nd ed.). Wiley.
- [13] Ke, Z. (2023). Random forest for dynamic risk prediction of recurrent events: A pseudo-observation approach. https://zhenkewu.com/assets/pdfs/papers/rfre_po_main.pdf
- [14] Lawless, J. F. (1995). The analysis of recurrent events for multiple subjects. *Statistics in Medicine*, 14(11), 1105–1112.
- [15] Lawson, A. B. (2001). *Statistical Methods in Spatial Epidemiology*. Wiley.
- [16] Li, Y., & Ryan, L. (2002). Modeling spatial survival data using semi-parametric frailty models. *Biometrics*, 58(2), 287–297.
- [17] Li, Y., & Lin, X. (2006). Semiparametric normal transformation models for spatially correlated survival data. *Journal of the American Statistical Association*, 101(474), 591–603.
- [18] Lin, D. Y., Wei, L. J., & Ying, Z. (2000). Semiparametric regression for the mean and rate functions of recurrent events. *Journal of the Royal Statistical Society: Series B*, 62(4), 711–730.
- [19] Liu, X., & Pan, R. (2020). Analysis of large heterogeneous repairable system reliability data with static system attributes and dynamic sensor measurement in big data environment. *Quality Technol-*

ogy & Quantitative Management, 17(2), 194–213. <https://doi.org/10.1080/16843703.2020.1712718>

- [20] Meeker, W. Q., & Hong, Y. (2014). Reliability meets big data: Opportunities and challenges. *Quality Engineering*, 26(1), 102–116.
- [21] Musa, K. I. (2000). Survival Analysis: Kaplan-Meier and Cox Proportional Hazard Regression. *Data Analysis in Medicine and Health using R*.
- [22] Nirmalkanna, K., & Cigsar, C. (2024). Analysis of recurrent event processes with dynamic models for event counts. *Statistics in Biosciences*. <https://link.springer.com/article/10.1007/s12561-024-09432-x>
- [23] Peña, E. A., Strawderman, R. L., & Hollander, M. (2001). Nonparametric estimation with recurrent event data. *Journal of the American Statistical Association*, 96(453), 129–142.
- [24] Prentice, R. L., Williams, B. J., & Peterson, A. V. (1981). On the regression analysis of multivariate failure time data. *Biometrika*, 68(2), 373–379.
- [25] Sahu, S. K., & Mardia, K. V. (2005). Recent trends in modeling spatio-temporal data. *University of Southampton Workshop Proceedings*. <https://www.southampton.ac.uk/~sks/workshop/revsis.pdf>
- [26] Smith, T. (2000). Kaplan Meier and Cox Proportional Hazards Modeling: Hands On Survival Analysis. *Academia.edu*.
- [27] Valdez, L. D., et al. (2020). Cascading failures in complex networks. *Journal of Complex Networks*, 8(2), cnaa013. <https://academic.oup.com/comnet/article/8/2/cnaa013/5849333>

- [28] Wang, N., & Taylor, J. M. G. (1999). Joint modeling of longitudinal and time-to-event data. *Biometrics*, 55(3), 587–598.
- [29] Wang, M. C., Qin, J., & Chiang, C. T. (2001). Analyzing recurrent event data with informative censoring. *Journal of the American Statistical Association*, 96(455), 1057–1065.
- [30] Wu, G., et al. (2024). The impact of cerebral vessels morphological alteration and white matter hyperintensities burden on the one-year risk of ischemic stroke recurrence. *BMC Medical Imaging*, 25, Article 150. <https://bmcmedimaging.biomedcentral.com/articles/10.1186/s12880-025-01687-0>
- [31] Xiao, X., Ye, Z., & Revie, M. (2024). Learning local cascading failure pattern from massive network failure data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 73(5), 1155–1184. <https://doi.org/10.1093/rssc/qlae030>
- [32] Zhai, Q., et al. (2024). Modeling recurrent failures on large directed networks. *Journal of the American Statistical Association*, 120(549), 251–265. <https://doi.org/10.1080/01621459.2024.2319897>
- [33] Zhang, R. Q., & Shi, X. P. (2024). LSTM-COX model: A concise and efficient deep learning approach for handling recurrent events. *arXiv preprint arXiv:2405.18518*. <https://arxiv.org/pdf/2405.18518v1.pdf>
- [34] Zhang, Y., Liu, X., & Pan, R. (2025). Advanced predictive models for recurrent event data in large networks. *Journal of Network and Systems Management*, 33(2), 123–145.
- [35] Zhao, L., & Hanson, T. (2011). Spatially varying coefficient models for survival data. *Biometrics*, 67(4), 1510–1518.

- [36] Zhou, H., & Hanson, T. (2018). A unified framework for fitting Bayesian semiparametric models to arbitrarily censored survival data, including spatially-referenced data. *Journal of the American Statistical Association*, 113(523), 1230–1241. <https://doi.org/10.1080/01621459.2017.1345742>