The Annals of Applied Statistics
2019, Vol. 13, No. 4, 2586–2610
https://doi.org/10.1214/19-AOAS1286

ON BAYESIAN NEW EDGE PREDICTION AND ANOMALY DETECTION IN COMPUTER NETWORKS[1]

BY SILVIA METELLI* AND NICHOLAS HEARD*,†

*Imperial College London** and Heilbronn Institute for Mathematical Research†*

Monitoring computer network traffic for anomalous behaviour presents an important security challenge. Arrivals of new edges in a network graph represent connections between a client and server pair not previously observed, and in rare cases these might suggest the presence of intruders or malicious implants. We propose a Bayesian model and anomaly detection method for simultaneously characterising existing network structure and modelling likely new edge formation. The method is demonstrated on real computer network authentication data and successfully identifies some machines which are known to be compromised.

1. Introduction. Statistical anomaly detection can complement existing, typically "signature-based" enterprise network defence systems which monitor for known security violations. In contrast, anomaly-based methods use probability models for the normal evolution of a network and look for any significant deviations (Neil et al. (2013), Turcotte, Heard and Neil (2014)). Signature-based methods rely on databases of known compromises (Cahill et al. (2002)), and are therefore less well suited to detecting rare or new infections; anomaly detection methods face a more difficult task, but allow possible identification of new, unknown attack vectors (Patcha and Park (2007)).

Computer network traffic data can be collected from network routers as an online stream of connection events, which will be summarised here as a continuous-time stochastic process of time-varying directed graphs $\{G_t\}$: Assuming potentially very large but fixed candidate sets of client nodes X and server nodes Y, $\{G_t\}$ will be a time-increasing set of directed edges in $X \times Y$, such that an edge (x, y) exists from $x \in X$ to $y \in Y$ in G_t if and only if client x has connected to server y by time t.

An intruder gaining a foothold in a computer network at time t may initially have very limited or no information about the previous communication patterns summarised by G_t. Therefore the intruder, despite not wishing to stand out in the network traffic, may be more likely to initiate new connections between hosts which have never communicated before. Such activity can provide a valuable signal for detecting the presence of the intruder. However, performing meaningful

Received May 2019; revised July 2019.

[1] Supported in part by The Alan Turing Institute and the Lloyds Register Foundation Programme on Data-Centric Engineering.

Key words and phrases. Computer networks, anomaly detection, new edges, Bayesian inference.

anomaly detection on the arrivals of new edges is nontrivial since new connections occur at a relatively high frequency for legitimate reasons, and with considerable heterogeneity between different network hosts.

A robust statistical model of edge formation for anomaly detection requires two distinct components: First, we require an understanding of the *rate* at which individual client hosts form new edges. Second and more challenging, we must predict the *identity* of new edges formed by each client; what could be an unsurprising connection for some hosts could be a very unusual connection for others. In the absence of further information about the structure of the network, the second consideration requires development of a notion of similarity of network hosts, such that *similar* clients may connect to *similar* servers; here, similarity will be considered under hard-thresholding with a clustering model, or soft-thresholding in a latent feature space.

We propose a point process modelling framework for directed (client, server) edges in the computer network. Specifically, we propose a Bayesian semi-parametric Cox model for the conditional intensity function of each potential (client, server) edge being made across the entire network, which addresses both the rates at which the client forms new edges and any underlying latent structural relationship between the clients and servers in the network. For the latter, the covariates in the Cox model will include unobserved structural features inferred from historic connections, encoding first cluster memberships and subsequently more flexible dot-products of respective latent feature positions. Both models allow us to examine whether shared connectivity is predictive of similar future interactions.

Expressing the arrivals of connections as a point process is a natural choice for modelling network interactions which has been used before in the analysis of evolving networks. In particular, Perry and Wolfe (2013) employ a Cox hazard model to predict interactions in communication networks, based on both the history of interactions and the node attributes. Similarly, we use a Cox proportional hazards model, but we focus on the stochastic intensity of each *new* (client, server) connection in the network, and we propose a different formulation of the baseline intensity, which will be common to all network hosts. In addition, we construct tailored covariates, which are better-suited for capturing both the heavy-tailed component and the underlying structure of complex computer networks. Such a model will allow to detect specific clients responsible for causing anomalous behaviour, potentially offering a stronger detection power. Following a similar perspective, autoregressive and Cox processes have been employed as potential point process models (Taddy (2010), Zammit-Mangion et al. (2012)) of dynamic network. In particular, multivariate self-exiting Hawkes processes have been recently used for modelling arrivals of events in the fields of social networks (Linderman and Adams (2014), Hall and Willett (2016), Li et al. (2017)) and crime data (Zhou, Zha and Song (2013)). In the present work, the self-exiting nature of computer network data, where specific events can be seen to occur in bursts and at particular times of day, will be captured by the time-dependent covariates included in the model.

Actor-based models have also been frequently used for modelling changes in network behaviour over time (Snijders, van de Bunt and Steglich (2010)).

The remainder of this article is organised as follows: Section 2 describes a motivating computer network data stream that will be used to demonstrate our new edge anomaly detection method. Section 3 introduces a mathematical formulation for the aspects of computer network data which we will need, utilising counting processes and marked point processes. Section 4 proposes a framework for modelling the intensity of arrival of new edges, with the two specific latent feature formulations outlined in Sections 5 and 6. Section 7 validates the methodology proposed using synthetic data and then compares the performance of the two proposed formulations on real computer authentication data, and Section 8 uses the superior model to perform anomaly detection. Finally, Section 9 offers some conclusions.

2. LANL computer network authentication.

The network traffic data used for analysis consist of authentication events (Kent (2015a, 2015b)) collected over 58 days from the enterprise computer network at Los Alamos National Laboratory (LANL). Those 58 days yielded 336,806,387 time-stamped client-to-server authentication events between 16,230 clients and 15,417 servers. Figure 1 shows the respective outdegree and indegree distributions of clients and servers, where the outdegree of a client is defined to be the number of unique server computers receiving authentication requests from that client, while the indegree for a server is the number of unique clients making authentication connections. Both degree distributions follow an approximate power law, although the modal outdegree is higher than the modal indegree, since the majority of servers have a very small population of connecting clients.

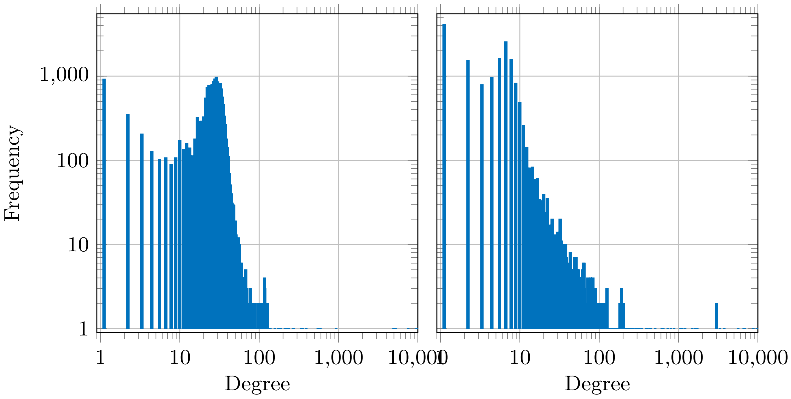

FIG. 1. *Log-log plot of the degree distributions for clients (left) and servers (right) in the LANL computer network authentication data.*

TABLE 1

Numbers of events and unique server computers connected to by four client computers in the LANL authentication data identified as compromised (*number of red team labelled events/total number of events*)

	Event frequency		Unique authenticated servers	
Compromised client	Red team	Total	Red team	Total
C17693	701	1717	296	534
C18025	3	101	1	29
C19932	19	10,008	8	30
C22409	26	36,253	3	31

These authentication data provide an ideal benchmark for testing anomaly-detection methods as they contain a "red team" penetration testing operation during the period of data collection; a subset of 749 authentication connections have been labelled as known compromise events. Table 1 shows how those events were distributed across four compromised client machines.

As we will be restricting our interest to the formation of new edges over time, Figure 2 shows a bar plot for the daily rate of occurrence of new edges, both for the full "bulk" data and just for red team events. There are 134,688 new edges created during the first day, corresponding to almost 35% of the total new edges in the data. This is to be expected, as formation of new edges is a nonstationary process; initially each client will establish a high number of new connections that will later correspond to its regular traffic, while in the longer term the rate of new edge formation will necessarily reduce, but will typically still be far from zero. The compromised clients in the red team data do not conform to this model of

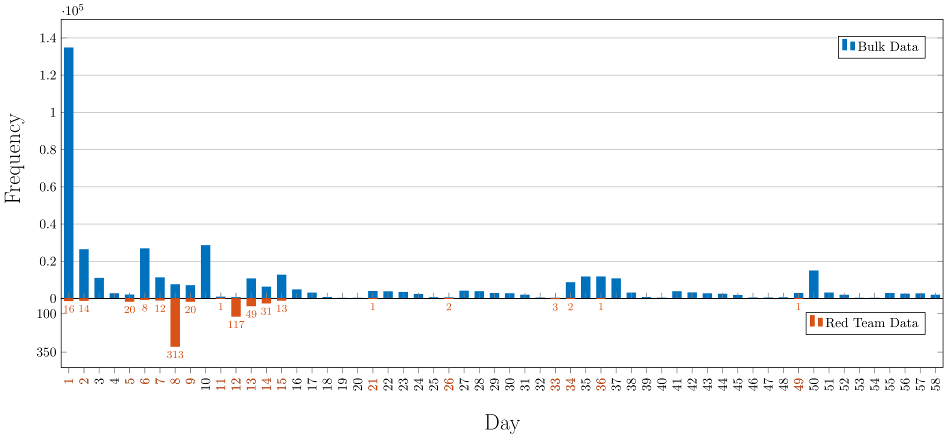

FIG. 2. *Bar plots of the number of new edges formed during each day of traffic in the bulk data* (*blue bars*) *and just in the red team data* (*red bars*).

normal new edge formation: from the 29 days of traffic, the largest number of compromised new edges is formed during days 8 and 12.

3. Computer network traffic as a marked point process. We encode the traffic observed on a computer network as a directed graph of connections from a set of *clients* X to a set of *servers* Y, which can be naturally represented as a time-varying bipartite graph; note that X and Y may refer to the same collection of computers or IP addresses, but for these purposes they can be considered as separate entities.

For the reminder of the article, the central quantity of interest for modelling the formation of new edges in the bipartite graph will be the intensity at time t for observing a new connection between a client computer $x \in X$ and a server $y \in Y$, which will be denoted $\lambda_{xy}(t)$. Its formal specification will be discussed in more details in the next section, while here we provide a mathematical description of some aspects of the computer network that will be necessary for our specification of $\lambda_{xy}(t)$.

Specifically, the arrivals of connections between computers can be expressed as a marked point process $(\mathcal{T}, \mathcal{E}) = ((T_n)_{n\geq 1}, (E_n)_{n\geq 1})$ where each random variable T_n is an $\mathbb{R}^+$-valued event time and E_n is a corresponding $(X \times Y)$-valued (client,server) mark. Let $0 \leq t_1 \leq t_2 \leq \cdots$ be the realised sequence of event times and let $e_n = (x_n, y_n) \in X \times Y$ be the corresponding mark for the nth event.

From $(\mathcal{T}, \mathcal{E})$ we define a continuous-time, left-continuous stochastic process of random graphs $\{G_t | t \geq 0\}$ from the set of network graph edges observed. Formally,

$$G_t = \{(x, y) | (x, y) \in X \times Y, N_{x,y}(t) > 0\},$$

where

$$N_{x,y}(t) = \sum_{n\geq 1} \mathbb{1}_{[0,t)}(t_n)\mathbb{1}_{(x,y)}(e_n) \tag{3.1}$$

is the left-continuous counting process of connections from client x to server y prior to time t. Note that edges first observed at time t are not included in G_t in this formulation. Summing (3.1) over X or Y yields corresponding counting processes of connections prior to time t from client x or to server y respectively,

$$N_{x,\cdot}(t) = \sum_{y\in Y} N_{x,y}(t), \qquad N_{\cdot,y}(t) = \sum_{x\in X} N_{x,y}(t).$$

For client x, let $(\mathcal{T}^x, \mathcal{Y}^x) = ((T_n^x)_{n\geq 1}, (Y_n^x)_{n\geq 1})$ be the subprocess for which the client mark is x, corresponding to those indices n for which $\mathbb{1}_x(x_n) = 1$. We will be interested in bursts of formation of new edges, so first define the binary variables

$$u_n^x = \mathbb{1}_{(X\times Y)\setminus G_{t_n^x}}\{(x, y_n^x)\}$$

such that $u_n^x = 1$ if and only if the nth connection from client x is a new edge. Then define

$$I_{x,1}(t) = \begin{cases} 1, & N_{x,\cdot}(t) = 0, \\ u^x_{N_{x,\cdot}(t)}, & N_{x,\cdot}(t) \geq 1; \end{cases} \tag{3.2}$$

and then recursively, for $m \geq 2$,

$$I_{x,m}(t) = \begin{cases} I_{x,m-1}(t), & N_{x,\cdot}(t) < m, \\ u^x_{N_{x,\cdot}(t)-1} I_{x,m}(t), & N_{x,\cdot}(t) \geq m. \end{cases} \tag{3.3}$$

The binary variable $I_{x,m}(t)$ will take value 1 if the last m connections made by client x were each new, and therefore represent a burst of new edge formation of length m.

Finally, from $(\mathcal{T}, \mathcal{E})$ we define the subprocess $(\mathcal{T}', \mathcal{E}') = ((T'_n)_{n\geq 1}, (E'_n)_{n \geq 1})$ of unique new edges observed in $(\mathcal{T}, \mathcal{E})$, corresponding to those indices n for which $\mathbb{1}_{G_{t_n}}\{(x_n, y_n)\} = 0$. From the realised sequence of unique edges $(e'_n)_{n\geq 1} = ((x'_n, y'_n))_{n\geq 1}$, it is simple to define the time-varying outdegrees of clients and in-degrees of servers: respectively,

$$N_x^+(t) = \sum_{n\geq 1} \mathbb{1}_{[0,t)}(t'_n)\mathbb{1}_x(x'_n), \qquad N_y^-(t) = \sum_{n\geq 1} \mathbb{1}_{[0,t)}(t'_n)\mathbb{1}_y(y'_n). \tag{3.4}$$

4. New edge intensity model.

We model the conditional intensity of a new directed connection being formed for every possible (client, server) pair through a Cox proportional hazards model (Cox (1972)) incorporating both static and time-dependent covariates and a baseline hazard rate, whose shape will be determined by a function $r(t) \geq 0$. More specifically, the intensity at time t for observing a new connection between a client computer $x \in X$ and a server $y \in Y$ is modelled as the product of the baseline hazard and the exponential of a linear combination of covariates,

$$\lambda_{xy}(t) = r(t) \exp\{\alpha \cdot D_{xy}(t) + \beta_{xy} \cdot Z_{xy}(t)\} \mathbb{1}_{(X\times Y)\setminus G_t}\{(x, y)\}, \tag{4.1}$$

where $D_{xy}(t) = (N_x^+(t), N_y^-(t), I_{x,1}(t), I_{x,2}(t))$ and the corresponding coefficients $\alpha = (\alpha_1, \ldots, \alpha_4) \in \mathbb{R}^4$; for each $(x, y) \in X \times Y$, $Z_{xy}(t) \in \mathbb{R}^k$ is a vector of length $k > 0$ quantifying the relative attraction of client x to server y at time t, and $\beta_{xy} \in \mathbb{R}^k$. Note that the intensity (4.1) becomes zero once the pair (x, y) have been observed. Furthermore, note that the model implies that the baseline intensity $r(t)$ is considered to be common to all network hosts and therefore a nuisance parameter whose functional form does not affect the model inference. Despite this assumption provides a vital inferential simplification, it will not always fully hold in practice. Here, the assumption can be consider valid if no client-specific seasonal, periodic patterns can be detected. The validity of this assumption will be assessed in Section 7 for the real computer network data introduced in Section 2.

The formulation of the conditional intensity (4.1) will be used throughout the remainder of the article, and the quantity $Z_{xy}(t)$ will be of central interest. Its specification, together with the other chosen covariates, will be discussed in the next section.

The conditional intensity function for the counting process $\mathcal{T}'$ of new connections being made across the entire network is the double sum of (4.1) over X and Y,

$$\lambda(t) = r(t) \sum_{x \in X} \sum_{y \in Y} \exp\{\alpha \cdot D_{xy}(t) + \beta_{xy} \cdot Z_{xy}(t)\} \mathbb{1}_{(X \times Y) \setminus G_t}\{(x, y)\}. \tag{4.2}$$

4.1. *Discussion of chosen regression covariates*, $D_{xy}(t)$. The population of servers in an enterprise computer network typically has a heavily right-skewed degree distribution, with a small number of servers having a very high indegree, meaning they are connected to by most clients. Similarly, the outdegree distribution of clients can also follow a power law for large degree values, although very small outdegrees are less common. These observations are illustrated for real data in Figure 1. To incorporate this highly variable *popularity* of different client and server machines into the model, we use the time-varying outdegree of each client computer x and indegree of each server computer y defined in (3.4).

Furthermore, while compromised clients tend to form a large number of new edges within a small period of time, study of computer traffic data suggests new edges are also commonly formed in bursts by benign nodes. In Heard and Metelli (2014) the indicator variables $I_{x,1}$ (3.2) and $I_{x,2}$ (3.3) were found to be strongly significant predictors of the rate of occurrence of new edges in a computer network. These two variables, which indicate whether the last connection was new, or whether the last two connections were new, are therefore included in the model.

Finally, we propose a family of covariates $\{Z_{xy}(t)|(x, y) \in X \times Y, t \geq 0\}$ representing a general notion of *attraction* between clients and servers. Two alternative formulations will be considered in the next two sections: a hard-threshold clustering model and a soft-threshold latent feature model.

The coefficients α in (4.1) could be viewed as nuisance parameters; the corresponding covariates $N_x^+(t)$, $N_y^-(t)$, $I_{x,1}(t)$, $I_{x,2}(t)$ (which respectively measure the relative connectivity of the client and server, and whether the client is currently making a burst of new edges) have already shown in previous research (Heard and Metelli (2016)) to be informative about the hazard of new edges. The question here is whether, having accounted for these simple, intuitive covariates, we can find any underlying latent structure in the network which can provide further information about which (client, server) pairs might be particularly suited and therefore more likely to connect; this will be measured by the inferred magnitude of the coefficients $\beta = \{\beta_{xy}\}$ in (4.1).

4.2. *Conditional likelihood-based Bayesian inference.* Following Cox (1972), we condition on the event times of $\mathcal{T}'$ of $(\mathcal{T}', \mathcal{E}')$ and work with the conditional likelihood of the event marks $\mathcal{E}'|\mathcal{T}'$. The conditional likelihood can be calculated sequentially, as a product of predictive probabilities for the identities of each new edge, given the corresponding event time and the previous edges formed so far.

Given time-ordered edges $(t'_1, e'_1), \dots, (t'_{n-1}, e'_{n-1})$, the predictive distribution for the nth new edge is simply

$$\begin{aligned}\mathbb{P}_{E'_n}\{(x, y)|(t'_1, e'_1), \dots, (t'_{n-1}, e'_{n-1}), t'_n\} &= \frac{\lambda_{xy}(t'_n)}{\lambda(t'_n)} \\ &= \frac{\exp\{\alpha \cdot (N^+_x(t'_n), N^-_y(t'_n), I_{x,1}(t'_n), I_{x,2}(t'_n)) + \beta_{xy} \cdot Z_{xy}(t'_n)\}}{\sum_{(x',y')\notin G_{t'_n}} \exp\{\alpha \cdot (N^+_{x'}(t'_n), N^-_{y'}(t'_n), I_{x',1}(t'_n), I_{x',2}(t'_n)) + \beta_{x'y'} \cdot Z_{x'y'}(t'_n)\}}.\end{aligned} \tag{4.3}$$

After observing n time-ordered edges $(t'_1, e'_1), \dots, (t'_n, e'_n)$, the conditional likelihood is simply the product of these predictive probabilities,

$$\begin{aligned}\mathbb{P}(\mathcal{E}'|\mathcal{T}', \alpha, \beta, \{Z_{xy}\}) &= \prod_{i=1}^{n} \mathbb{P}_{E'_i}\{(x'_i, y'_i)|(t'_1, e'_1), \dots, (t'_{i-1}, e'_{i-1}), t'_i\} \\ &= \prod_{i=1}^{n} \frac{\exp\{\alpha \cdot (N^+_{x'_i}(t'_i), N^-_{y'_i}(t'_i), I_{x'_i,1}(t'_i), I_{x'_i,2}(t'_i)) + \beta_{x'_i y'_i} \cdot Z_{x'_i y'_i}(t'_i)\}}{\sum_{(x,y)\notin G_{t'_i}} \exp\{\alpha \cdot (N^+_x(t'_i), N^-_y(t'_i), I_{x,1}(t'_i), I_{x,2}(t'_i)) + \beta_{xy} \cdot Z_{xy}(t'_i)\}}.\end{aligned} \tag{4.4}$$

For a Bayesian approach, we choose standard normal prior distributions for components of α and the free parameters of $\beta = \{\beta_{xy}|(x, y) \in X \times Y\}$. Different choices of prior distributions are discussed and analysed in the Supplementary Material (Metelli and Heard (2019)), while the two proposed constructions for $\{Z_{xy}|(x, y) \in X \times Y\}$ and their respective prior distributions are now described in Sections 5 and 6.

5. Clustering formulation.

The first proposal for constructing $Z_{xy}(t)$ is to build "peer group" clusters within both X and Y, based on similarity in connectivity patterns. The idea is that if many of the clients with similar connection patterns to x also connect to server y, then perhaps it is more likely that x will form a connection with y.

We first define a generalisation of the outdegrees and indegrees, respectively, of clients and servers (3.4) by considering the degree of a node in the bipartite graph restricted to subsets of clients or servers: For $C \subseteq X$ and $S \subseteq Y$, let

$$N^+_{x|S}(t) = \sum_{n\geq 1} \mathbb{1}_{[0,t)}(t'_n)\mathbb{1}_x(x'_n)\mathbb{1}_S(y'_n), \qquad N^-_{y|C}(t) = \sum_{n\geq 1} \mathbb{1}_{[0,t)}(t'_n)\mathbb{1}_y(y'_n)\mathbb{1}_C(x'_n).$$

In words, $N^+_{x|S}(t)$, for example, is the number of servers in S connected to by client x, prior to time t.

Let $\mathbb{C} = \{C_1, \dots, C_L\}$ be a partition of X, and $\mathbb{S} = \{S_1, \dots, S_M\}$ a partition of Y. Then, in a slight abuse of notation, let $\mathbb{C}(x) \in \mathbb{C}$ be the unique cluster containing client x, and $\mathbb{S}(y) \in \mathbb{S}$ be the cluster containing server y. Then based on these clustering configurations, define the attraction covariate for the pair (x, y) as

$$Z_{xy}(t) = \left(N^{+}_{x|\mathbb{S}(y)}(t), N^{-}_{y|\mathbb{C}(x)}(t)\right). \tag{5.1}$$

For $L \geq 1$ client clusters and $M \geq 1$ server clusters, the specification (5.1) for $Z_{xy}(t)$ implies LM free parameters for $\beta = \{\beta_{xy}\}$, which are assigned independent standard normal priors. To complete a Bayesian model specification, the prior distributions for the clustering configurations $\mathbb{C}$ and $\mathbb{S}$ are assumed uniform over the space of all possible configurations. As most clustering models, the model presented here takes a class-oriented representation similar to stochastic blockmodels (Holland, Laskey and Leinhardt (1983), Rohe, Chatterjee and Yu (2011), Sussman et al. (2012)) where each entity can only be assigned to a single row or column cluster. This assumption will be relaxed in the next section, where each client and each server will be associated with a vector of latent features, and the two performance will be then compared in Section 7.

Under the conditional likelihood (4.4), posterior inference is required for the joint distribution of all unknown parameters,

$$\mathbb{P}(\alpha, \beta, \mathbb{C}, \mathbb{S} | \mathcal{T}', \mathcal{E}') \propto \mathbb{P}(\mathcal{E}' | \mathcal{T}', \alpha, \beta, \{Z_{xy}\}) \mathbb{P}(\alpha) \mathbb{P}(\beta | \mathbb{C}, \mathbb{S}) \mathbb{P}(\mathbb{C}) \mathbb{P}(\mathbb{S}).$$

As for most Bayesian models, exact inference is not analytically tractable and so Markov chain Monte Carlo (MCMC) simulation is required to perform posterior inference. Details of the MCMC simulation employed can be found in the Supplementary Material, while below we describe a spectral clustering approach used to provide initial cluster configurations to seed the MCMC sampling algorithm.

5.1. *Spectral biclustering.* For notational convenience, suppose here that the clients and servers have been numbered such that $X = \{1, \dots, |X|\}$ and $Y = \{1, \dots, |Y|\}$. After observing n edges $(x'_1, y'_1), \dots, (x'_n, y'_n)$, let $A \in \{0, 1\}^{|X| \times |Y|}$ be the $|X| \times |Y|$ adjacency matrix with entries $A_{x,y} = \sum_{i=1}^{n} \mathbb{1}_{(x,y)}\{(x'_i, y'_i)\}$ indicating which of the possible edges have been observed. Let D_X and D_Y be diagonal matrices of the row and column sums of A, respectively, equal to the outdegrees of clients and indegrees of servers.

The commonly used spectral biclustering algorithm of Dhillon (2001) and Cho et al. (2004) calculates a truncated-singular value decomposition of

$$D_X^{-1/2} A D_Y^{-1/2}. \tag{5.2}$$

In this context, the left singular-vectors of (5.2) correspond to the clients projected into a K-dimensional latent space, and similarly the right singular vectors correspond to latent-space positions for the servers. Performing k-means clustering on these latent representations yields cluster configurations of the clients and the servers.

6. Latent feature formulation. In Section 5.1, latent-space representations were used as data locations within a clustering algorithm, for grouping together similar clients or servers. Latent features increase the flexibility of the generative process of class-oriented models by letting each entity be associated with a vector of latent features. In this representation, the latent-space vectors for each client and each server of the network graph have a common but potentially unbounded dimension $K < \infty$. The latent positions are then combined by a simple dot-product (Rubin-Delanchy et al. (2017)) to provide a score of attraction between clients and servers.

Let $U = (u_1, \dots, u_{|X|}) \in \mathbb{R}^{|X| \times K}$, $V = (v_1, \dots, v_{|Y|}) \in \mathbb{R}^{|Y| \times K}$ be matrices containing the K-dimensional, real-valued latent feature vectors for the client and server computers respectively. Then the latent feature model score of attraction between client x and server y is simply given by

$$Z_{xy} = u_x \cdot {v_y}^T. \tag{6.1}$$

Note that (6.1) is fixed and not time-varying. The time-varying nature of the sociability among clients and servers is here captured by the baseline intensity, which is assumed to incorporate the seasonal diurnal behaviour of the network, with traffic network more dense at specific times of the day.

Furthermore, since the magnitude of the vectors u_x and v_y provide a *sociability* effect for each client x or server y respectively, we restrict the $\{\beta_{xy}\}$ regression coefficients to a single constant

$$\beta_{xy} = \beta,$$

implying just one free parameter β which is assigned a standard normal prior distribution. Again, different choices of normal prior distributions are evaluated in the Supplementary Material.

To specify a prior distribution on the triple (K, U, V), we look to introduce some cluster structure in the features by encouraging sparsity in the matrices U and V, so that each client/server only possesses a subset of the possible latent feature measurements. The cluster feature matrix U is therefore decomposed into two components: a binary matrix $\Delta_U \in \{0, 1\}^{|X| \times K}$ with entry $\Delta_{xk} = 1$ if and only if client x possesses feature k, and a second matrix $\tilde{U} \in \mathbb{R}^{|X| \times K}$ comprising the continuous feature values of each feature for each client. The feature matrix U can then be expressed as the elementwise Hadamard product of these two matrices,

$$U = \Delta_U \odot \tilde{U}.$$

Similarly, the server feature matrix $V = (v_1, \dots, v_{|Y|}) \in \mathbb{R}^{|Y| \times K}$ is expressed as the Hadamard product of matrices $\Delta_V \in \{0, 1\}^{|Y| \times K}$ and $\tilde{V} \in \mathbb{R}^{|Y| \times K}$,

$$V = \Delta_V \odot \tilde{V}.$$

For determining K and the subset of features selected for each client and server, independent Indian buffet process (IBP) priors (Ghahramani, Griffiths and Sollich

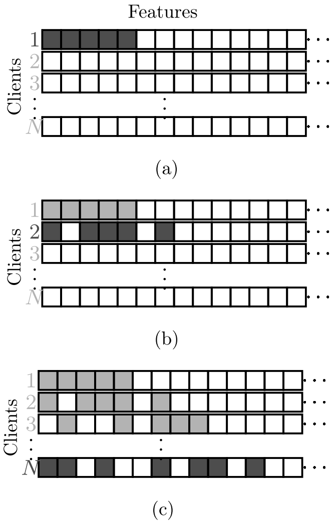

FIG. 3. *An illustration of the Indian Buffet Process for the client binary matrix* Δ_U. (a) *The first client samples* Poisson(θ) *features, which is recorded by changing the corresponding entries of* Δ_U *to one.* (b) *and* (c) *For the xth client, the first step is activating the previously sampled features with probability proportional to the number of clients who already have these features active, while the next step is to activate* Poisson(θ/x) *new features.*

(2007)) with Poisson parameter $\theta > 0$ are assigned to Δ_U and Δ_V. The IBP defines a distribution over the rows of an infinite binary matrix, which is most easily described sequentially with an illustration; the construction is described for the example of the client matrix Δ_U in Figure 3. The IBP assumes exchangeability of the rows of Δ_U and Δ_V, so each client and server has an expected number of features equal to the Poisson parameter θ. If K_U (K_V) is the total number of features activated within Δ_U (Δ_V), then the resulting dimension for the model is taken to be the maximum, $K = \max\{K_U, K_V\}$. Conditional on K, the continuous-valued entries of $\tilde{U}$ and $\tilde{V}$ are assigned independent standard normal priors.

Note that there is a slight identifiability issue between the free matrix entries of U, V and the single coefficient β. However, the relative dimensionality of the large matrices against a single scalar means that if the event data support a strong latent feature effect, under the chosen priors this will be reflected through the parameter β which incurs just a single penalty.

Given the conditional likelihood (4.4), the joint posterior distribution is given up to proportionality by

$$\mathbb{P}(\alpha, \beta, U, V|\mathcal{T}', \mathcal{E}') \propto \mathbb{P}(\mathcal{E}'|\mathcal{T}', \alpha, \beta, U, V)\mathbb{P}(\alpha)\mathbb{P}(\beta)\mathbb{P}(U)\mathbb{P}(V).$$

Full details on posterior inference for α, β, U and V can be found in the Supplementary Material. Following Section 5, we use a truncated-singular value decomposition with imposed sparsity to provide initial low-rank latent positions for clients and servers, which is then used to seed MCMC sampling.

6.1. *Sparse truncated SVD.* In Section 5, truncated-singular value decomposition was applied to a normalised transformation (5.2) of the binary adjacency matrix A. Under the latent feature formulation, we have obtained best results from using a weighted adjacency matrix whose (x, y) entry is an empirical estimate of the intensity of connections between client x and server y. From observing the arrivals of new connections over a time interval $[0, T]$, we construct a $|X| \times |Y|$ matrix, denoted $\hat{\Lambda}$, where the (x, y) component $\hat{\Lambda}_{xy}$ is an estimate of the intensity in (4.1) obtained from a simplistic conjugate Bayesian model. For the stochastic process of directed graphs $\{G_t | t \geq 0\}$, define the random variable

$$T_{xy} = \inf_t \{t | (x, y) \in G_t\}, \tag{6.2}$$

as the waiting time until the edge (x, y) is first observed. For a simple Bayesian model, we suppose $T_{xy} \sim \operatorname{Exp}(\Lambda_{xy})$, with conjugate prior for the unknown intensity $\Lambda_{xy} \sim \Gamma(\gamma, \upsilon)$. Correspondingly, let

$$\tilde{t}_{xy} = \begin{cases} T_{xy}, & T_{xy} \leq T, \\ T, & T_{xy} > T, \end{cases} \tag{6.3}$$

be the observed value, or else a right-censoring time, for (6.2) after observing the evolution of G_t on $[0, T]$. Then conditioning on this observed value (6.3), the posterior distribution for the intensity is straightforward,

$$\Lambda_{xy} | \tilde{t}_{xy} \sim \Gamma\big(\gamma + \mathbb{1}_{[0,T]}(t_{xy}), \upsilon + t_{xy}\big).$$

This yields the posterior mean estimate for the (x, y) entry of Λ,

$$\hat{\Lambda}_{xy} = \frac{\gamma + \mathbb{1}_{[0,T]}(\tilde{t}_{xy})}{\upsilon + \tilde{t}_{xy}}.$$

Note that $\hat{\Lambda}_{xy}$ is decreasing as a function of $\tilde{t}_{xy}$, meaning that a first observation of (x,y) early on in $[0, T]$ gives a higher adjacency value. From taking the truncated-singular value decomposition of $\hat{\Lambda}$, the left singular vectors correspond to the client computers projected into a K-dimensional latent space, and the right singular vectors corresponding to the servers.

Inducing sparsity penalties. Under the IBP prior, the implied assumption for Δ_U and Δ_V is to retain only a *sparse* subset of an unbounded number of features. Thus, to obtain initial matrices resembling draws from an IBP we need to enforce sparsity on both the left and the right singular vectors obtained above from the truncated-SVD of $\hat{\Lambda}$. This can be achieved by interpreting these singular vectors as the regression coefficients of a linear regression and then imposing adaptive lasso penalties to the least square regression (Lee et al. (2010)). For a triplet (s, u, v) of u, v singular vectors and scalar singular value s, we minimise the following penalised sum-of-squares criterion with respect to (s, u, v),

$$\|\hat{\Lambda} - suv^T\|_F^2 + \rho_u P_1(s, u) + \rho_v P_2(s, v),$$

where $P_1(s,u)$ and $P_2(s,v)$ are sparsity-inducing penalty terms and ρ_u and ρ_v tuning parameters that determine the amount of regularisation. Furthermore, the sparseness of U and V can strongly depend on the choice of the penalisation parameters; hence, following Sill et al. (2011), we use stability selection (Meinshausen and Bühlmann (2010)) to obtain stable penalisation parameters and to control the degree of sparsity.

7. Results. Before proceeding with anomaly detection, in this section we evaluate the plausibility of the two proposed model formulations on simulated data and then we apply the methodology on real computer network data, assessing the strength of the effects of each included covariate as well as the goodness-of-fit, for both the cluster and latent feature model. In both cases, we use 5000 MCMC iterations with a burn-in period of size 1000.

7.1. *Synthetic data.* In this section we validate the proposed methodology on synthetic data generated from both model formulations, respectively introduced in Section 5 and Section 6. For each model, we simulate 100,000 events and we set the nuisance parameters to $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1.5$. For the sociability parameters, we set $\beta_{1,l} = 2$ for $l = 1, \dots, L$ and $\beta_{2,m} = 2$ for $m = 1, \dots, M$ for the clustering formulation, while for the latent formulation we set the single latent features coefficient to $\beta = 2$. We can confirm the accuracy of the methodology by plotting the estimated intensity function for a specific (x, y) pair against the true intensity for the simulated model in Figure 4. The intensity can be modulated up or down according to the pair's covariates: here the most significant peak in intensity occurs after observing a bursts of new edges in the previous times, where

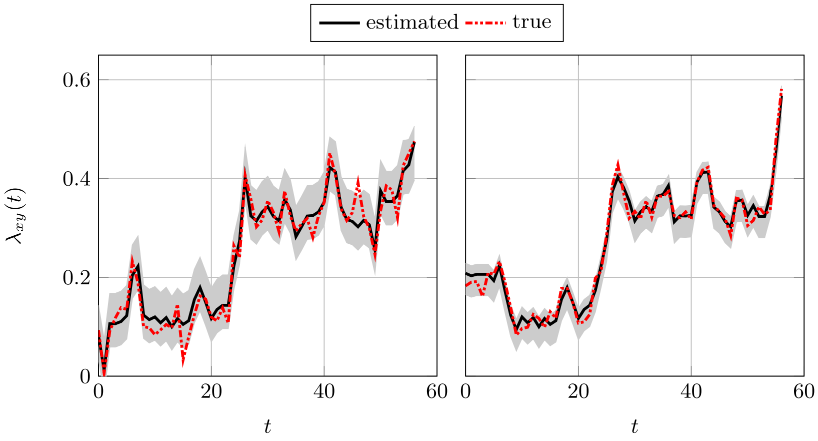

FIG. 4. *True model intensity along with estimated model intensity* (*with* 95% *HPD range in grey*) *for the clustering formulation* (*left*) *and latent formulation* (*right*) *for the first* 56 *time units before the new edge event occurred.*

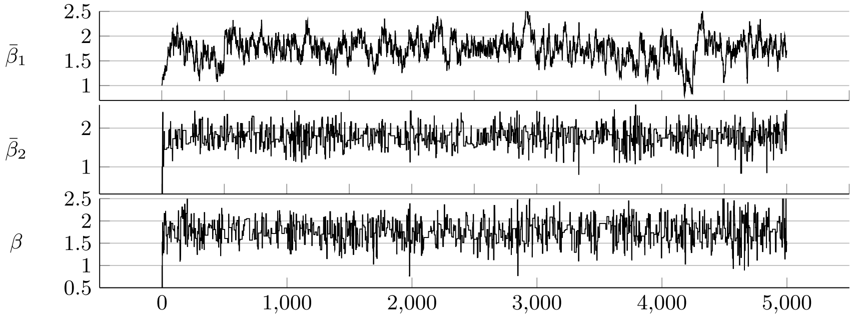

FIG. 5. *Trace plots of sociability parameter posterior distributions under both model formulations.*

most of the server computers observed in the bursts belong to the same cluster as server y, thus increasing the hazard of observing that particular server in the next following times. The shaded region indicates the 95% highest posterior density (HDP) credible intervals for the posterior intensity. We note that the estimates and credible intervals appear more precise when the flexible, latent feature model formulation is used. Then, Figure 5 shows the trace plots of the posterior distributions of the parameters of interest. For the cluster formulation, to allow coherent posterior model averaging across the variable dimensions L and M (cf. Section 5), we report the mean coefficients $\bar{\beta}_1 = \sum_{l=1}^{L} \beta_{1,l}/L$ and $\bar{\beta}_2 = \sum_{m=1}^{M} \beta_{2,m}/M$ respectively, summarising the coefficients over the fitted client and server clusters. For both models, convergence to the stationary distribution is reached in less than 5000 MCMC iterations. Finally, in Figure 6 we plot the posterior distribution of the model parameters of main interest along with the true parameters (indicated by a red vertical line) used in the simulation.

7.2. *Application to real authentication data.* In this section, we apply the methodology proposed in Section 5 and Section 6 to the real computer network

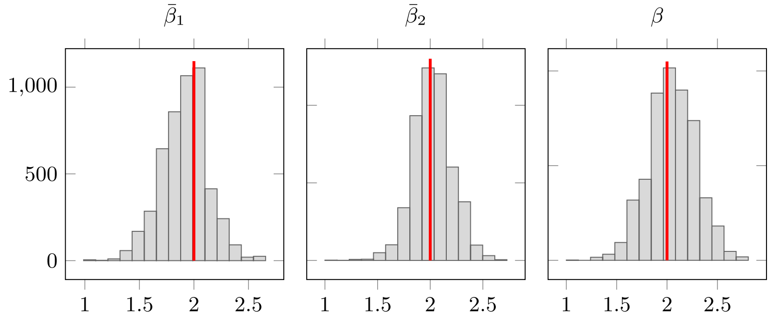

FIG. 6. *Histograms for the sampled posterior parameter distributions for the parameters of main interest, with true values (solid lines).*

data described in Section 2. For computational tractability, we have tested our method on a sample of event data from a random selection of 1000 clients. These events were combined with the red team event data for analysis; to demonstrate robustness 15 repetitions of this sampling procedure were performed, to cover the entire population of client computers. The Supplementary Material contains details on the MCMC iteration procedure used and posterior checking to inspect convergence.

7.2.1. *Proportional hazards assumption.* As stated in Section 4.2, a key feature of model (4.1) is to consider the baseline intensity $\lambda(t)$ as a nuisance parameter—common to all network hosts—whose functional form does not affect model inference. To ensure both model formulations proposed can be considered valid, we need to test this assumption in the context of computer network data before focusing on covariate effects and goodness-of-fit. In particular, we are interested in analysing whether the empirical distributions of the event times of each client in the network can be considered homogeneous: if no significant difference can be found, then it is plausible to assume that the baseline intensity function reflects the sinusoidal diurnal behaviour of the network, with traffic more dense at specific times of the day, thus being a common nuisance to all network hosts. To this end, we employ a k-sample Anderson–Darling test (Scholz and Stephens (1987)), which is a nonparametric test for the hypothesis that k samples belong to the same population. Here we have $k = |X|$. Let $\hat{F}_1, \dots, \hat{F}_{|X|}$ be the empirical distributions of event times t for each client x and $\hat{G}_N$ that of the pooled sample of all $N = n_1 + \cdots + n_{|X|}$ event times. The test statistic is then given by

$$A^2_{|X|,N} = \sum_{x=1}^{|X|} n_x \int_S \frac{\{\hat{F}_{x,n_x}(t) - \hat{G}_N(t)\}^2}{\hat{G}_N(t)\{1 - \hat{G}_N(t)\}} \, d\hat{G}_N(t),$$

where $S = \{t : \hat{G}_N(t) < 1\}$. Under the null hypothesis, it can be shown that the test statistic has asymptotically the same distribution as $\sum_{x=1}^{\infty} \frac{1}{x(x+1)} Z_x^2$, with Z_x^2 i.i.d. χ^2 random variables with $|X| - 1$ degrees of freedom. The percentiles of this limiting null distribution can be approximated by means of Pearson curves (Solomon and Stephens (1978)). Here, this test yields a nonsignificant estimated p-value of 0.432, validating the reasonability of the model in the proposed context.

7.2.2. *Covariate effects.* The posterior means of model coefficients and boxplots of their 95% HPD credible intervals, under both the cluster formulation and the latent-feature formulation, are shown in Figures 7 and 8 respectively. The posterior estimates of the parameters are all significantly positive, with an acceptably small level of variation across samples. For the nuisance parameters α, this confirms the *popularity* effect for both client and servers, where computers that have many connecting neighbours are more likely to make further connections, and also

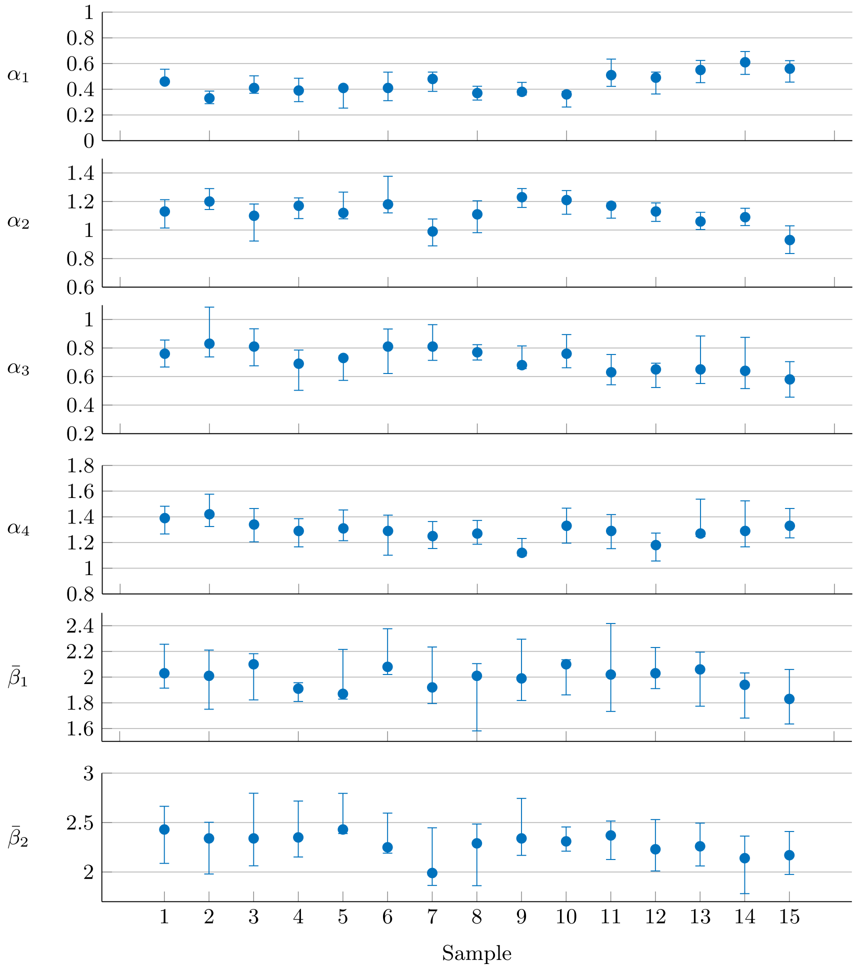

FIG. 7. *Posterior estimates coefficients under the cluster formulation, with credible intervals.*

the presence of bursts of new edge formation by clients. More interestingly, the significant β parameters confirm that strong additional information is provided by the identity of the links already formed and the latent communities they suggest, whether this is characterised by hard clustering of clients and servers, or softer partitioning through a dot-product model using latent positions.

Figure 8 shows four sets of estimates for the latent-feature model. The bullet points correspond to the full inference procedure proposed in Section 6 and the Supplementary Material, where sparse truncated SVD is used to seed an MCMC exploration of the variable dimension latent feature space, while the diamond points correspond to a simpler finite latent feature beta-Bernoulli (BeP) process with K known number of features, used for the purpose of comparison. As shown

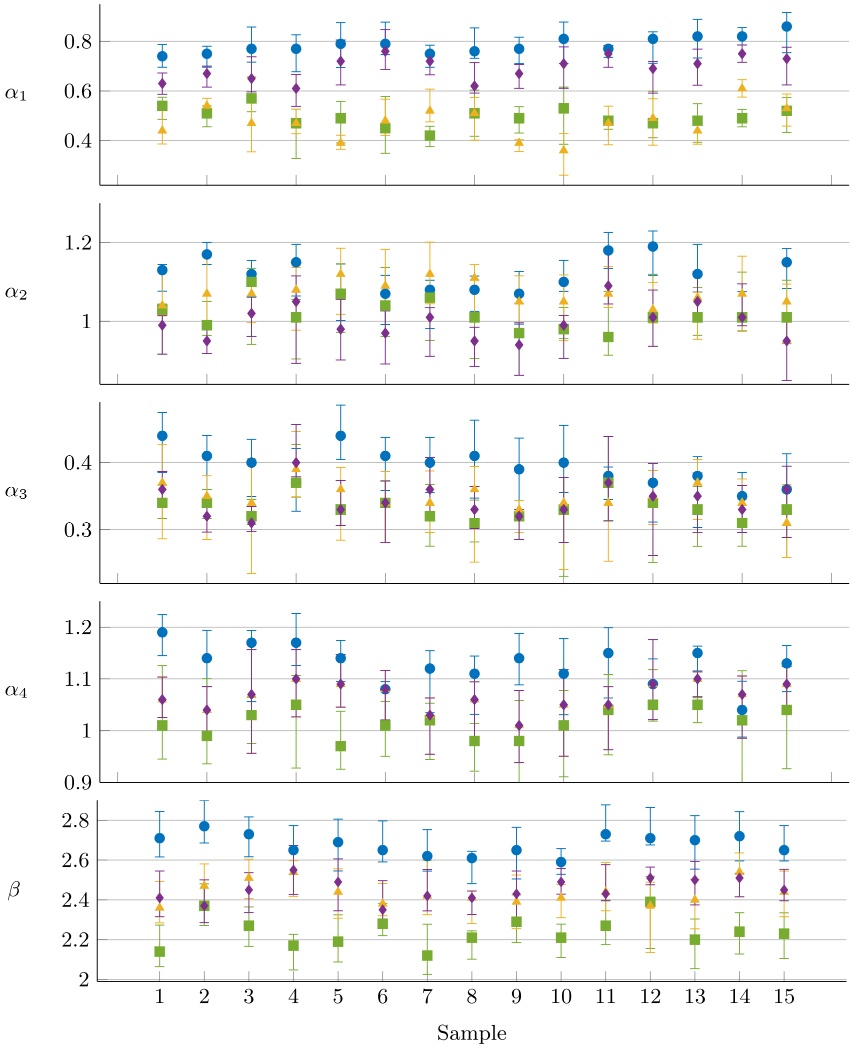

FIG. 8. *Four sets of posterior estimates coefficients under the latent feature formulation, with credible intervals, obtained from full MCMC under the IBP* (●), *finite BP MCMC* (♦), *sparse truncated SVD with stability selection* (■), *and standard truncated SVD* (▲).

in Ghahramani, Griffiths and Sollich (2007), the generative BeP process taking the limit as $K \to \infty$ corresponds to the IBP detailed in Section 6. The other two sets of points represent the resulting estimates from not performing MCMC and just using the truncated SVD to propose latent features, either with sparsity imposed (square points) or with no sparsity (triangle points). When using sparsity penalties, we have tested the matrix $\hat{\Lambda}$ against normality, as stability selection has been introduced for standard regression models with gaussian errors: Shapiro–Wilks

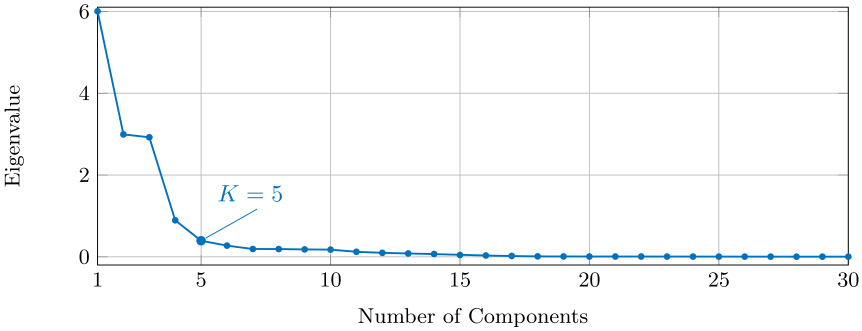

FIG. 9. *Scree plot of SVD eigenvalues for the first sample analysed, in the range* $[0, 30]$, *with the characteristic elbow occurring at* $K = 5$.

normality tests, which can be found in the Supplementary Material, confirm that $\hat{\Lambda}$ is approximately normally distributed.

To ensure a meaningful comparison of these coefficients, after MCMC the latent positions U and V from the IBP processes are rescaled so that the mean absolute value of the features $\{Z_{xy}\}$ is the same as that from the initial values from sparse truncated SVD. This is necessary as both U and V are allowed to change dimension during each MCMC step. In each case the coefficients are all significantly positive. In particular, all the coefficient estimates in Figure 8 are higher in magnitude when using the full inference procedure, suggesting that there is considerably more structure in the data and the MCMC exploration of the feature space under the IBP is worthwhile for identifying more significant latent feature covariates compared to both updating the latent positions under a simpler, finite model and to not performing any update move on the initial latent positions.

Figure 9 shows the scree plot of the SVD eigenvalues in the simplest case when no sparsity is imposed. Here $K = 5$ appears to be a suitable choice. When performing sparse truncated-SVD incorporating stability selection, a dimensionality of $K = 6$ is automatically chosen. Figure 10 shows the trace plot of the number

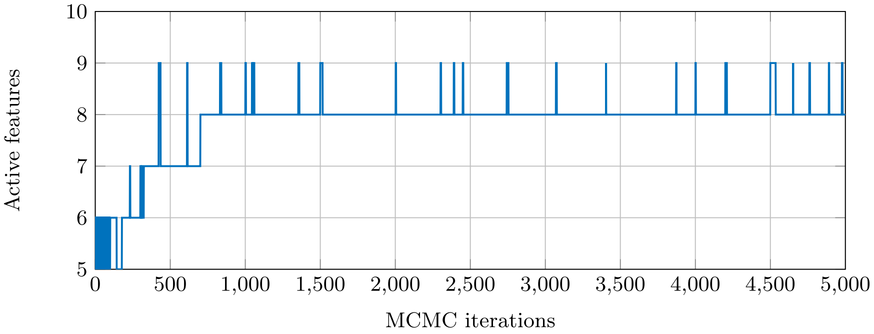

FIG. 10. *Number of active latent features during the MCMC run* (*IBP*).

of active features from MCMC in the full inference procedure; there is some mixing, and the resulting posterior distribution has a strong mode at $K = 8$, and some small probability associated with $K = 9$.

7.2.3. *Goodness-of-fit.* For assessing fit, we focus on the posterior predictive distribution (4.3) of each new edge observed in $(\mathcal{T}', \mathcal{E}')$, calculating a corresponding p-value. After observing time-ordered edges $(t'_1, e'_1), \dots, (t'_{n-1}, e'_{n-1})$, let $e'_n = (x'_n, y'_n)$ be the nth edge in $(\mathcal{T}', \mathcal{E}')$, observed at time t'_n. The model partial likelihood for observing a new edge at time t'_n is given by

$$\ell_n(x, y) = \frac{\lambda_{xy}(t'_n)}{\sum_{(x,y)\notin G_{t'_n}} \lambda_{xy}(t'_n)}, \tag{7.1}$$

where $0 \le \ell_n(x, y) \le 1$ and $\sum_{(x,y)\notin G_{t'_n}} \ell_n(x, y) = 1$. Then, a discrete p-value for e'_n, denoted p_n, can be most simply obtained by summing (7.1) over all edges no more probable than the observed edge e'_n:

$$p_n = \sum_{(x,y)\notin G_{t'_n}} \ell_n(x, y)\mathbb{1}_{(0,\ell_n(x'_n,y'_n)]}\{\ell_n(x, y)\}. \tag{7.2}$$

To test for model fit, we perform a Kolmogorov–Smirnov (KS) test for the sequence of observed p-values from (7.2). As these p-values are generated from discrete random variables and so are stochastically larger than uniform, randomised p-values (Pearson (1933))—which will be marginally uniform on the unit interval if the model is correct—were used to perform the test. As shown in Figure 11, the distributions of the observed p-values under both model formulations are approximately uniform, and the KS tests yielded p-values of 0.364 and 0.678 for the cluster and latent feature models respectively.

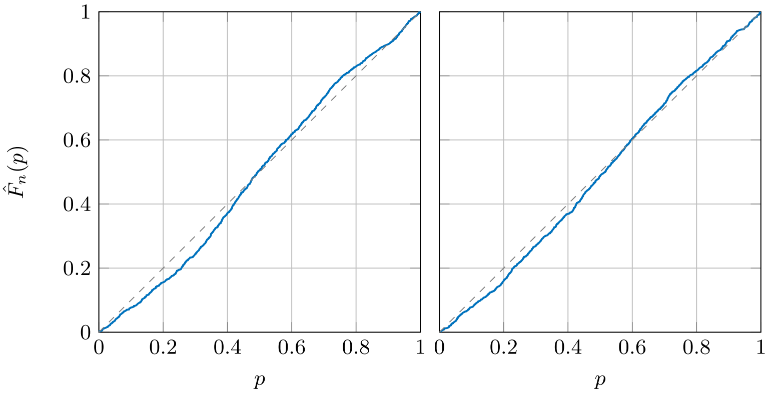

FIG. 11. *Empirical cumulative distribution of the observed p-values under the cluster model (left) and the latent feature model (right), against the* Uniform(0, 1) *cumulative distribution function.*

TABLE 2
Likelihood ratios for the three different model settings

Comparison	Likelihood ratio
MCMC (IBP) *vs* MCMC (BeP)	2.93
MCMC (IBP) *vs* Sparse SVD with stability selection	3.07
MCMC (IBP) *vs* SVD	2.32

The predictive performance of the latent feature model using the different inference methods used in Figure 8 was assessed on 100,000 out-of-training sample authentication events. Table 2 reports the averaged likelihood ratios, showing that performing full MCMC under the IBP leads to a distinct improvement in model fit. In particular, the additional layer of flexibility introduced by the IBP yields better predictive performance than the simpler, finite BeP model. In addition, when no MCMC moves are performed on the latent matrices, the plain SVD leads to better results than the sparse decomposition, which in this case may just remove useful information; clearly the extra effort of MCMC is required to find a useful sparse solution.

Finally, the predictive performance and computational efficiency of the clustering and latent feature models are compared in Table 3. We also report the AUC score, the area under the ROC curve, for the held-out data. These results confirm that the latent feature model, under the IBP prior, outperforms the cluster model in terms of likelihood. Unfortunately, this comes at a cost of a substantially longer running time for the MCMC sampler. Most of the MCMC computing time is spent recalculating the likelihood function: even if only one element of the latent position vectors is changed, the likelihood can still only be updated in $\mathcal{O}(K^2)$ time.

8. Anomaly detection. For anomaly detection in cyber-security, we are concerned with determining if the new authentication connections observed over some time period can be regarded as relatively normal with respect to the learned intensity model (4.2) or whether they should be flagged as anomalous. For this objective, we can utilise the sequence of predictive p-values $(p_n)_{n\geq 1}$ (7.2) derived from

TABLE 3
Log-likelihoods, AUC score and average MCMC computation time for 10,000 *out-of-training authentication events under the clustering and later feature models*

Model formulation	Log-likelihood	AUC	Iteration time
Cluster	−18802.34	0.9352	81.4 s
Latent-feature (IBP)	−18389.93	0.9745	131.7 s

the sequence of new edges $(\mathcal{T}', \mathcal{E}') = ((T_n')_{n\geq 1}, (E_n')_{n\geq 1})$ arriving in the dynamic graph G_t.

8.1. *Combining p-values.* To identify local deviations in the network, we are interested in finding anomalous behaviour over time for specific client computers (alternatively, interest could equally be focused on servers). For client x, let $(\mathcal{T}'^x, \mathcal{Y}'^x) = ((T_n'^x)_{n\geq 1}, (Y_n'^x)_{n\geq 1})$ be the subprocess of the new edge process $(\mathcal{T}', \mathcal{E}')$ for which the client mark is x, corresponding to those indices n for which $\mathbb{1}_x(x_n) = 1$. Let $(p_n^x)_{n\geq 1}$ be the corresponding subsequence of p-values from (7.2) relating to client x. Under the null hypothesis of no attack and the model of normality holding, the p-values are approximately uniformly distributed on the unit interval. To find anomalous clients in the network we need to combine the p-values for each client x to provide a single, time-varying score of surprise. This is a canonical p-value combination problem often performed with Fisher's method (Fisher (1925)), which we here used to define a control chart

$$s_x(t) = \bar{\chi}^2_{2\{1+N_x^+(t)\}}\left(-2\sum_{n\geq 1} \mathbb{1}_{[0,t)}(t_n'^x)\log p_n^x\right), \tag{8.1}$$

where $N_x^+(t)$ is the outdegree of client x approaching time t and $\bar{\chi}^2_\nu(\cdot)$ is the survivor function of the chi-squared distribution with ν degrees of freedom. Note that by extension from $(p_n^x)_{n\geq 1}$, the quantity $s_x(t)$ for any $t \geq 0$, is also approximately uniform on the unit interval. Extreme, small values of (8.1) correspond to anomalous behaviour with surprising new edge formation, and so we are typically interested in anomaly scores such as

$$\inf_{t\geq 0} s_x(t). \tag{8.2}$$

Note that the distribution of (8.2) will vary for different clients x, since some clients make many more new connections than others (see Figure 1). For this reason, tailored rejection regions for each client, based on their number of connections, must be calculated by simple Monte Carlo estimation.

8.2. *Results.* We restrict attention to the latent-feature model, which was shown in Section 7 to have the highest predictive accuracy. Figure 12 shows the p-values (7.2) and control chart scores (8.1) on the log-scale for two of the known compromised clients (C17693, C19932), and two randomly selected uninfected clients (C349, C586). In both infected cases we can see some extreme values in the p-values and the control charts, leading to clear detection at the indicated 1% and 0.1% significance thresholds; in contrast, the control charts of the uninfected clients are seen to be relatively well behaved, staying well above the thresholds for all t.

Figure 13 shows the receiver operating characteristic (ROC) curves, for each of the 15 random sample repetitions, for the sequence of p-values (7.2) and the

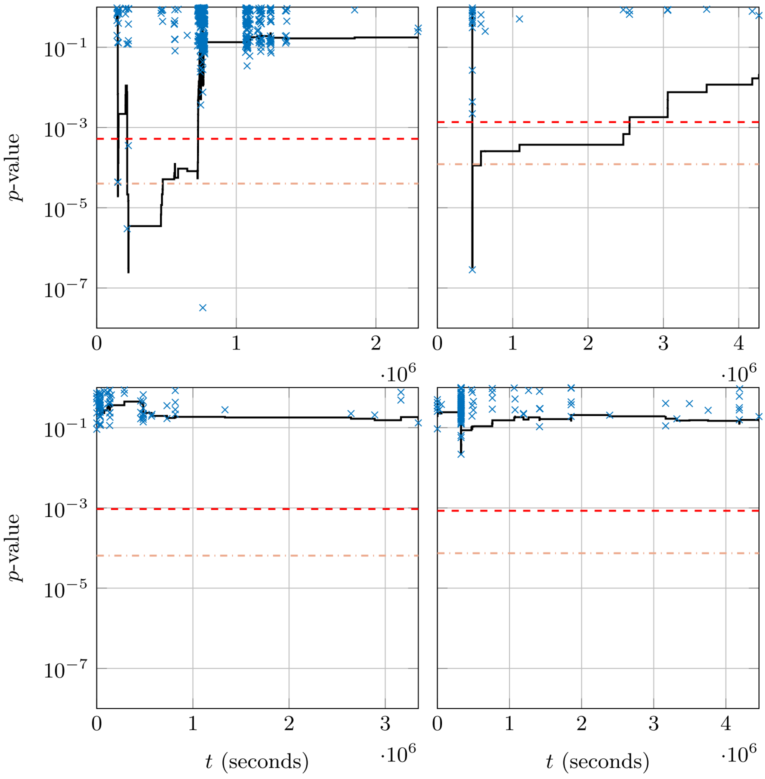

FIG. 12. *Observed p-values* (×) *over time and the corresponding control chart* (——) *for two compromised clients in the red team exercise* (*top left*: *C*17693; *top right*: *C*19932) *and two uninfected clients in the bulk data* (*bottom left*: *C*349; *bottom right*: *C*586). *Control chart thresholds at the* 1% (– – –) *and* 0.1% (–·–·–) *significance levels are shown for each client.*

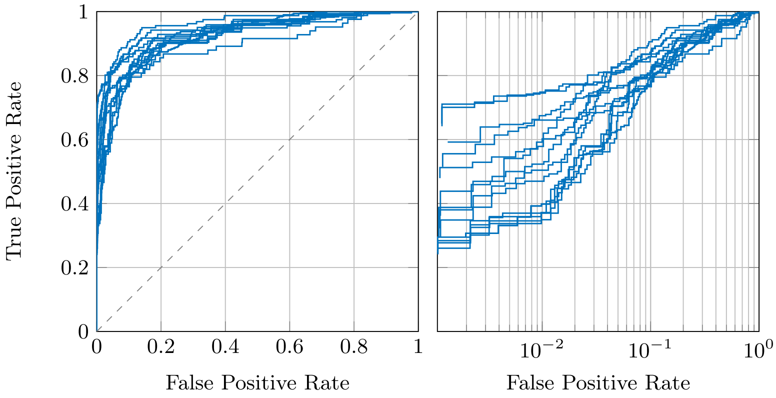

FIG. 13. *ROC curves for each client, for each sample repetition, shown on both linear* (*left*) *and log scales* (*right*).

scores obtained using (8.2). In the right panel of the figure, focusing on low false positive thresholds, we still see an encouraging number of true positives which we would like to detect.

9. Discussion. This paper has proposed a Bayesian approach for simultaneously characterising latent network structure and predicting likely new edge formation based on learning similarities between network hosts. Similarity has been considered under hard-thresholding with a clustering model, or soft-thresholding in a latent feature space.

The methodology has been shown to be well suited for modelling new edges in a large network: results from both formulations showed considerable significance attached to the time-varying covariates characterising evolving latent network structure, and strongly indicate the positive impact of introducing notions of client and server similarities into the model. In particular, the most flexible, nonparametric latent feature approach, utilising the Indian Buffet Process as a prior distribution, has led to the highest performance in terms of predictive accuracy.

The method has shown to drive encouraging, although not conclusive, anomaly detection performance in detecting compromised clients at low false positive rates. Efficiently combining p-values, where most will be from the null distribution, is an important problem often encountered in cyber-security applications, where only a tiny proportion of activity will correspond to cyber threat. Thus, the choice of the construction of the control chart could and should be adapted according to the precise target of the anomaly detection scheme.

SUPPLEMENTARY MATERIAL

Supplement: Simulation study and posterior inference (DOI: 10.1214/19-AOAS1286SUPP; .pdf). Details of the simulation study and Bayesian posterior inference considered in this paper.

REFERENCES

CAHILL, M. H., LAMBERT, D., PINHEIRO, J. C. and SUN, D. X. (2002). Detecting fraud in the real world. In *Handbook of Massive Data Sets* 911–929. Kluwer Academic, Dordrecht.

CHO, H., DHILLON, I. S., GUAN, Y. and SRA, S. (2004). Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the Fourth SIAM International Conference on Data Mining* 114–125. SIAM, Philadelphia, PA. MR2388433

COX, D. R. (1972). Regression models and life-tables. *J. Roy. Statist. Soc. Ser. B* **34** 187–220. MR0341758

DHILLON, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 269–274. ACM, New York.

FISHER, R. A. (1925). *Statistical Methods for Research Workers*. Oliver & Boyd, Edinburgh.

GHAHRAMANI, Z., GRIFFITHS, T. L. and SOLLICH, P. (2007). Bayesian nonparametric latent feature models. In *Bayesian Statistics* 8. *Oxford Sci. Publ.* 201–226. Oxford Univ. Press, Oxford. MR2433194

HALL, E. C. and WILLETT, R. M. (2016). Tracking dynamic point processes on networks. *IEEE Trans. Inform. Theory* **62** 4327–4346. MR3515754

HEARD, N. and METELLI, S. (2014). Modelling new edge formation in a computer network through Bayesian variable selection. In *Joint Intelligence and Security Informatics Conference* (*JISIC*), 2014 *European* 272–275. IEEE, New York.

HEARD, N. and METELLI, S. (2016). Model-based clustering and new edge modelling in a large computer network. In *IEEE International Conference on Intelligence and Security Informatics* (*ISI*), 2016 91–96. IEEE, New York.

HOLLAND, P. W., LASKEY, K. B. and LEINHARDT, S. (1983). Stochastic blockmodels: First steps. *Soc. Netw.* **5** 109–137. MR0718088

KENT, A. D. (2015a). Comprehensive, multi-source cyber-security events. Los Alamos National Laboratory, Washington, DC.

KENT, A. D. (2015b). Cybersecurity data sources for dynamic network research. In *Dynamic Networks in Cybersecurity* Imperial College Press, London.

LEE, M., SHEN, H., HUANG, J. Z. and MARRON, J. S. (2010). Biclustering via sparse singular value decomposition. *Biometrics* **66** 1087–1095. MR2758496

LI, S., XIE, Y., FARAJTABAR, M., VERMA, A. and SONG, L. (2017). Detecting changes in dynamic events over networks. *IEEE Trans. Signal Inform. Process. Netw.* **3** 346–359. MR3661705

LINDERMAN, S. W. and ADAMS, R. P. (2014). Discovering latent network structure in point process data. In *Proceedings of the* 31*st International Conference on Machine Learning* 1413–1421.

MEINSHAUSEN, N. and BÜHLMANN, P. (2010). Stability selection. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **72** 417–473. MR2758523

METELLI, S. and HEARD, N. (2019). Supplement to "On Bayesian new edge prediction and anomaly detection in computer networks." DOI:10.1214/19-AOAS1286SUPP.

NEIL, J., HASH, C., BRUGH, A., FISK, M. and STORLIE, C. B. (2013). Scan statistics for the online detection of locally anomalous subgraphs. *Technometrics* **55** 403–414. MR3176546

PATCHA, A. and PARK, J. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Comput. Netw.* **51** 3448–3470.

PEARSON, K. (1933). On a method of determining whether a sample of size n supposed to have been drawn from a parent population having a known probability integral has probably been drawn at random. *Biometrika* **25** 379–410.

PERRY, P. O. and WOLFE, P. J. (2013). Point process modelling for directed interaction networks. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **75** 821–849. MR3124793

ROHE, K., CHATTERJEE, S. and YU, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Statist.* **39** 1878–1915. MR2893856

RUBIN-DELANCHY, P., PRIEBE, C. E., TANG, M. and CAPE, J. (2017). A statistical interpretation of spectral embedding: The generalised random dot product graph. Preprint. Available at arXiv:1709.05506.

SCHOLZ, F.-W. and STEPHENS, M. A. (1987). k-sample Anderson–Darling tests. *J. Amer. Statist. Assoc.* **82** 918–924. MR0910001

SILL, M., KAISER, S., BENNER, A. and KOPP-SCHNEIDER, A. (2011). Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics* **27** 2089–2097.

SNIJDERS, T., VAN DE BUNT, G. and STEGLICH, C. (2010). Introduction to stochastic actor-based models for network dynamics. *Soc. Netw.* **32** 44–60.

SOLOMON, H. and STEPHENS, M. (1978). Approximations to density functions using Pearson curves. *J. Amer. Statist. Assoc.* **73** 153–160.

SUSSMAN, D. L., TANG, M., FISHKIND, D. E. and PRIEBE, C. E. (2012). A consistent adjacency spectral embedding for stochastic blockmodel graphs. *J. Amer. Statist. Assoc.* **107** 1119–1128. MR3010899

TADDY, M. A. (2010). Autoregressive mixture models for dynamic spatial Poisson processes: Application to tracking intensity of violent crime. *J. Amer. Statist. Assoc.* **105** 1403–1417. MR2796559

TURCOTTE, M. J., HEARD, N. A. and NEIL, J. (2014). Detecting localised anomalous behaviour in a computer network. In *Advances in Intelligent Data Analysis XIII—*13*th International Symposium* 321–332. Springer, Berlin.

ZAMMIT-MANGION, A., ALAN DEWAR, M., KADIRKAMANATHAN, V. and SANGUINETTI, G. (2012). Point process modelling of the Afghan War Diary. *Proc. Natl. Acad. Sci. USA* **109** 12414–12419.

ZHOU, K., ZHA, H. and SONG, L. (2013). Learning triggering kernels for multi-dimensional Hawkes processes. In *Proceedings of the* 30*th International Conference on Machine Learning* **28** 1301–1309.

DEPARTMENT OF MATHEMATICS
IMPERIAL COLLEGE LONDON
SOUTH KENSINGTON CAMPUS
LONDON SW7 2AZ
UNITED KINGDOM
E-MAIL: s.metelli13@imperial.ac.uk
n.heard@imperial.ac.uk