JOURNAL OF APPLIED STATISTICS
2022, VOL. 49, NO. 4, 858–883
https://doi.org/10.1080/02664763.2020.1845621

Taylor & Francis
Taylor & Francis Group

Check for updates

Statistical modeling of computer malware propagation dynamics in cyberspace

Zijian Fang[a], Peng Zhao[b], Maochao Xu[c], Shouhuai Xu[d], Taizhong Hu[a] and Xing Fang[e]

[a]Department of Statistics and Finance, University of Science and Technology of China, Hefei, Peoples Republic of China; [b]School of Mathematics and Statistics and Research Institute of Mathematical Sciences (RIMS), Jiangsu Provincial Key Laboratory of Educational Big Data Science and Engineering, Jiangsu Normal University, Xuzhou, Peoples Republic of China; [c]Department of Mathematics, Illinois State University, Normal, IL, USA; [d]Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA; [e]School of Information Technology, Illinois State University, Normal, IL, USA

ABSTRACT

Modeling cyber threats, such as the computer malicious software (malware) propagation dynamics in cyberspace, is an important research problem because models can deepen our understanding of dynamical cyber threats. In this paper, we study the statistical modeling of the macro-level evolution of dynamical cyber attacks. Specifically, we propose a Bayesian structural time series approach for modeling the computer malware propagation dynamics in cyberspace. Our model not only possesses the parsimony property (i.e. using few model parameters) but also can provide the predictive distribution of the dynamics by accommodating uncertainty. Our simulation study shows that the proposed model can fit and predict the computer malware propagation dynamics accurately, without requiring to know the information about the underlying attack-defense interaction mechanism and the underlying network topology. We use the model to study the propagation of two particular kinds of computer malware, namely the Conficker and Code Red worms, and show that our model has very satisfactory fitting and prediction accuracies.

ARTICLE HISTORY
Received 24 April 2019
Accepted 25 October 2020

KEYWORDS
Bayesian time series; cyber threats; MCMC; SIS; SIR

1. Introduction

Computer malicious software (malware), such as computer worms, are malicious computer programs that can replicate themselves to propagate in computer networks. For example, the Conficker worm, which first appeared in November 2008, rapidly spread in the Internet and infected millions of computers in the Internet within a short period of time. This malware exploited a vulnerability in Windows operating systems and used many advanced techniques, such as domain generation algorithms, self-defense mechanisms, updating via Web and Peer-to-Peer (P2P) networks, and efficient local propagation. The other well-known malware is the Code Red worm, which was first observed in July 2001. This

CONTACT Peng Zhao zhaop@jsnu.edu.cn

© 2020 Informa UK Limited, trading as Taylor & Francis Group

malware exploited a buffer overflow vulnerability and quickly infected millions of computers in the Internet. These incidents are just two examples of the many computer malware that spread in cyberspace, motivating the importance of understanding their propagation dynamics.

The importance of understanding computer malware propagation dynamics has motivated many studies, which can be categorized into two families. The first family of studies aim to model the *micro-level* attack-defense interactions that take place on top of computer networks, leading to the general concept of cybersecurity dynamics (cf. [34] and the references therein). These models accommodate the widely-studied cyber epidemic models, such as Susceptible-Infectious-Susceptible (SIS) and Susceptible-Infectious-Recovered (SIR) and their extensions [1,8,24,26,31,43], as special cases. Recently, Zhao et al. [44] investigated a model that contains a central node and a multiplex network with a patch dissemination network layer and a computer malware propagation network layer. This study considers constraints on the capacity of the central node and the bandwidth of the network links. This simulation-based study investigates the interplay between the computer malware propagation and the patch dissemination, which compete against each other. It is worth mentioning that this type of competing dynamics taking place on top of computer networks were studied earlier in [22,37,38,46]. Feng et al. [14] studied a Susceptible-Infectious-Recovered-Susceptible (SIRS) model describing the spatial and temporal dynamics of worm spreading in wireless sensor networks. Srivastava et al. [30] investigated a Susceptible-Exposed-Infected-Quarantined-Recovered (SEIQR) model for describing the dynamics of worm propagation in wireless sensor networks. Xia et al. [33] studied botnet propagation in social Internet of Things, and employed the mean-field equation theory to analyze the dynamics of botnet propagation. Zhen et al. [45] studied a particular kind of cybersecurity dynamics caused by the interactions between two classes of attacks and two classes of defenses. This study tackled a research problem that has been open for 1 years, by proving that the particular kind of dynamics is globally convergent in the entire parameter universe of the model. This result was further extended in [21] to show that an broader class of cybersecurity dynamics model is still globally convergent in the entire parameter universe of the model. Han et al. [16] very recently proved that an even more general class of cybersecurity dynamics is globally attractive, but may not be globally convergent, hinting that there is an inherent boundary between the cybersecurity dynamics models that converge to an equilibrium and the cybersecurity dynamics models that converge to a trajectory. Studies in this family of models can be characterized as follows: (i) they often use high-dimensional and highly-nonlinear differential equations to model the underlying micro-level attack-defense interactions; (ii) they often make some assumptions, such as the independence between certain events, although weakening such independence assumption has attracted the due amount of attention [10,35]; (iii) these models are yet to be evaluated by real-world data because it is hard to collect micro-level attack-defense interaction activities, which is worsen by the need to protect privacy of such interactions; (iv) these models are competent for analyzing the *asymptotic* behavior of the dynamics in the long run (i.e. the time $t \to \infty$), despite that such dynamics may converge exponentially; and (v) these models require full information about the underlying attack-defense interaction mechanisms, the underlying network topology, and the network security policy (for deriving what is known as the attack-defense structure [34]).

The second family of studies aim to model the computer malware propagation dynamics at the *macro-level* without considering the micro-level attack-defense interactions. These studies are data-driven, such as those modeling dynamical cyber attack rates, namely univariate or multivariate time series of the numbers of attempted attacks waged by attackers [3,9,27,28,36,39,41,42]. The present study falls into this family of studies but on a different problem than dynamical cyber attack rates because computer malware propagation dynamics corresponds to the time series of the evolution of the number(s) of infected computers (i.e. successful attacks, rather than attempted attacks). More specifically, our study is motivated by the following question that have not been investigated until now: *Given only the data describing the macro-level computer malware propagation dynamics (i.e. no information about the underlying attack-defense interactions, no information about the underlying network topology, and no information about the network security policy), how can we model the computer malware propagation dynamics using as few parameters as possible and predict (or forecast) the transient behavior of the dynamics while accommodating the potential uncertainty in the data. Answering this question would provide a deep understanding of the computer malware propagation dynamics. The importance of this question can also be justified by the statistical studies on modeling epidemic spreading outside the domain of cybersecurity [11,17,20,25].*

In this paper, we answer the preceding question by proposing to use a Bayesian Structural Time Series (BSTS) model to study data-driven computer malware propagation dynamics in cyberspace. We propose using a Bayesian Local Linear Trend (BLLT) model to investigate the dynamics of the number of compromised (or infected) computers in cyberspace, owing to the propagation of a computer malware. We show that our parsimonious BLLT model can effectively describe the dynamics and achieve a satisfactory prediction accuracy, by using both synthetic data and real malware propagation data. The proposed mode has a clear cybersecurity interpretation. We also discuss how the proposed model can take uncertainty into account in prediction. More specifically, our model can characterized as follows: (i) it falls into a statistical modeling approach, which is in contrast to the differential equation approach mentioned above; (ii) it is data-driven and therefore can be evaluated using real-world data; (iii) it describes and predicts transient behaviors of the computer malware propagation dynamics, which is in contrast to the asymptotic behaviors mentioned above; and (iv) it is a partial-information model, meaning that it does not require full information about the underlying attack-defense interaction mechanisms, the underlying network topology, and the network security policy. In summary, our model makes a particular contribution to the literature of computer malware propagation model dynamics in cyberspace, owing to its parsimony property and its Bayesian nature at the macro level.

The rest of the paper is organized as follows. Section 2 describes the datasets of the propagation dynamics of two particular computer malware in the real work, known as the Conficker and Code Red worms, and analyzes their basic statistics properties. Section 3 describes the proposed BSTS model and elaborates its cybersecurity relevance. Section 4 generates synthetic data of computer malware propagation dynamics on top of random and contact networks, and uses the synthetic data to evaluate the effectiveness of the proposed model. Section 5 uses the proposed model to study the Conficker and Code Red worms propagation dynamics datasets. Section 6 concludes the paper with future research directions.

2. Conficker and code red worms

In this section, we first describe two real-world datasets of computer malware propagation dynamics and then perform exploratory data analyses on them.

2.1. Data description and preprocessing

Conficker worm. This worm is a particular kind of computer malware. The dataset was collected between 20:00pm on November 20, 2008 and 6:00am on November 21, 2008, which is the initial period of the Conficker worm propagation dynamics, by the *network telescope* of the Center for Applied Internet Data Analysis (CAIDA) [6]. The CAIDA telescope passively monitors a /8 network (i.e. 2^{24} Internet IP addresses), which are associated to no Internet services but purely set up to receive (without responding to) incoming connections [2,40]. The telescope can recognize Conficker's probing packets because they target the Transmission Control Protocol (TCP) with destination port number 445, which is the vulnerable service the Conficker worm can exploit. In order to filter out the background radiation, the last hour data (i.e. 11:00pm-12:00am) monitored by the telescope on 19 November 2008 was used as a filter such that the packets received by the telescope during this hour are discarded.

Each Conficker worm probing packet includes a timestamp and a source IP address. There are a total number of 1,410,742 unique IP addresses in the dataset. In order to analyze the evolution of the propagation dynamics, namely the evolution of the total number of infected computers, we aggregate the data into 20-second time windows, resulting in 1,800 time windows. A computer is considered infected by the worm when the telescope observes the first Conficker probing packet originating from the computer, and an infected computer is considered recovered from the infection when the telescope observes the last probing packet originating from the computer before the end of data collecting period. The last 30 minutes is used as the observation window for determining whether an infected computer is recovered or not. That is, if the telescope does not observe a previously-observed infected computer for originating probing packets in this last 30 minutes, this computer is regarded as having recovered from the infection because the worm is designed to spread itself. The total number of infected computers at time window t, denoted by C_t, is computed by removing the number of recovered computers from the cumulative number of unique IPs by the end of time step t. This leads to a total of 1710 observations, namely $\{C_t, t = 1, \ldots, 1710\}$.

Code Red worm. The Code Red worm attacked computers running Microsoft's IIS web server and was first observed on July 15, 2001. The data analyzed in this paper comes from three sources: packet headers collected from CAIDA's /8 network telescope, timestamp/IP address pairs in TCP SYN packets received by two /16 networks at Lawrence Berkeley Laboratory, and sampled netflows from a router upstream traffic at CAIDA's /8 network telescope. The data was collected between 19:00 UTC on July 18, 2001 and 2:10 UTC on July 20, 2001. This preprocessed data is provided by CAIDA, which contains the timestamps of an IP address for transmitting the worm [6]. For our analysis purpose, we aggregate the data into 1-minute windows, and record the number of infected computers during the time windows, leading to a total number of 1,812 observations, denoted by $\{R_t, t = 1, \ldots, 1812\}$.

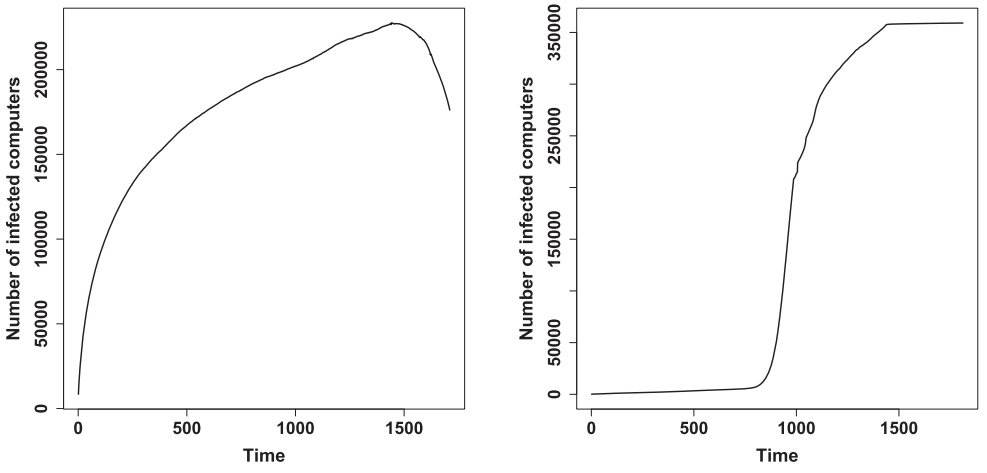

Figure 1. Propagation dynamics of the Conficker and Code Red worms. (a) C_t: the Conficker worm. (b) R_t: the Code Red worm.

2.2. Exploratory data analysis

The time series plot of the total number of infected computers by the Conficker worm is displayed in Figure 1(a). It is observed that the total number of infected computers increases rapidly during the initial stage, and then shows a steady increasing trend. After reaching the peak, the number of infected computers shows a decreasing trend. The time series plot of the total number of infected computers by the Code Red worm is displayed in Figure 1(b). It is observed that the total number of infected computers increases slowly during the initial stage and then increases rapidly. After reaching the peak, the number of infected computers shows a steady trend. It is interesting to see that the dynamics of the Code Red worm infection is different from that of the Conficker worm.

The boxplots of the numbers of infected computers by the Conficker and Code Red worms are displayed in Figure 2. For the Conficker worm, the mean number of infected computers is 179,481 and the median is 192,607. This suggests that the number of infected computers is very skewed as seen from Figure 2. The sample standard deviation is 44,295.61, which indicates that there exists a large variability among the numbers of infected computers. For the number of computers infected by the Code Red worm, the mean is 158,157 and the median is 56,539. It can be seen that the mean is much larger than the median, which suggests that the data is also very skewed. The sample standard deviation is 161,143.73, which indicates that there exists a large variability among the numbers of infected computers. As seen from Figure 2, the numbers of computers that are infected by the Code Red worm have many more observations on the left tail than the numbers of computers that are infected by the Conficker worm, which coincides with the Code Red worm's slowly-increasing pattern in the initial stage.

For modeling computer malware propagation dynamics in the real world, it is important to accommodate the uncertainty that can be incurred by network noise and/or misidentified network traffic. This is possible because, for example, a normal traffic may be

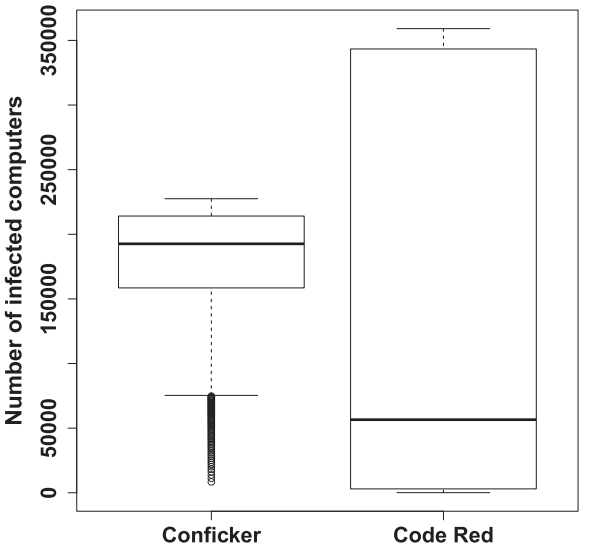

Figure 2. Boxplots of the numbers of infected computers by the Conficker and Code Red worms.

mistakenly identified as a malicious one (i.e. false positive) and/or a malicious traffic may be mistakenly identified as a normal one (i.e. false negative). Since Bayesian models are well suited for accommodating uncertainty and can provide intuitive and meaningful inferences [15], it motivates us to use the Bayesian approach to model the worm propagation dynamics.

3. Model and justification

In this section, we present the BLLT model and discuss its cybersecurity relevance.

3.1. Bayesian local linear trend (BLLT) model

The BLLT model is a state-space model with a Bayesian component. Let y_t be the observation (i.e. the observed number of infected computers when casted in the context of computer malware propagation dynamics) at time t, where $t = 1, \ldots, n$. The model is described as follows:

$$y_t = \mu_t + \epsilon_t \tag{1}$$

$$\mu_{t+1} = \mu_t + \delta_t + \epsilon_{\mu,t} \tag{2}$$

$$\delta_{t+1} = \delta_t + \epsilon_{\delta,t} \tag{3}$$

where μ_t is the level value of the trend at time t, δ_t is the expected increase in μ between t and $t+1$ and can be treated as the slope at time t, and $\epsilon_t \sim N(0, \sigma_t^2)$, $\epsilon_{\mu,t} \sim N(0, \sigma_\mu^2)$ and $\epsilon_{\delta,t} \sim N(0, \sigma_\delta^2)$ are noises that reflect the inherent uncertainty and/or measurement error. Intuitively, Equation (1), dubbed *observation* equation, relates the observed data y_t to the unobserved latent state μ_t, and Equations (2) and (3), dubbed *state transition* equations, describe the evolution in the latent space.

The BLLT model has several appealing properties. First, it allows one to infer the unobserved latent state μ_t from the observed data up to time $t \leq n$, denoted by $y_{1:t} = \{y_1, \ldots, y_t\}$ for short. Let $\alpha_t = (\mu_t, \delta_t)^\top$ be the inferred vector at time t in the latent state. The inference can be achieved by using the Kalman filter and Kalman smoother. At a high level, the Kalman filter recursively computes the predicted distribution $p(\alpha_{t+1}|y_{1:t})$ by combining $p(\alpha_t|y_{1:t-1})$ and y_t in a certain fashion; the Kalman smoother then updates the output from the Kalman filter to compute $p(\alpha_t|y_{1:n})$ at each value of t (see [4,13]). Second, it is parsimonious because it has only three parameters; this is an important property because parsimonious models are always preferred in statistics. Third, it is flexible because it can quickly adapt to *local variations*, which makes it capable of short-period predictions [29]. Fourth, it can accommodate complicated phenomena such as non-stationarity and structure breaks (e.g. different means or variances [4]), and can be extended to accommodate covariates and seasonal trends [29]. Since it is Bayesian in nature, it can provide distribution prediction rather than point prediction, leading to richer information in the prediction.

A special case of the BLLT model is called the Bayesian Local Linear (BLL) model [4], which can be obtained by setting $\delta_t = 0$ in Equation (2), meaning that there is no slope (i.e. no increase or decrease in the latent space). That is, the BLL model is:

$$y_t = \mu_t + \epsilon_t; \tag{4}$$

$$\mu_{t+1} = \mu_t + \epsilon_{\mu,t}. \tag{5}$$

In order to see if the BLLT model can be replaced by a simpler model, the BLL model is studied as well.

We propose the following cybersecurity interpretation of the BLLT model when casting it to the context of computer malware propagation dynamics. First, y_t can be interpreted as the observed number of infected computers at time t, which is different from the ground-truth number μ_t of infected computers because the infected computers are only partially observed in practice, owing to a range of factors (e.g. the size of a network telescope, observation errors incurred by false positive and/or false negatives). Second, ϵ_t is the noise incurred by inherent uncertainties or measurement errors. Third, δ_t is the increase (or decrease) in μ_t over time interval $[t, t+1]$, and is related to the previous changing amount δ_{t-1} with random noise $\epsilon_{\delta,t}$. Fourth, $\epsilon_{\mu,t}$ is the noise in μ_t, which can be incurred (for example) by the drop of attack packets when the network is congested [7].

3.2. Coping with the prior distribution of the BLLT model

When using a Bayesian model, one needs to cope with the matter of prior distribution. For the BLLT model, we have the model variance parameter vector

$$\theta_t = (\sigma_t, \sigma_\mu, \sigma_\delta)$$

and the inferred vector $\alpha_t = (\mu_t, \delta_t)^\top$, where $t = 1, \ldots, n$. That it,

$$\theta = (\theta_1, \ldots, \theta_n) \quad \text{and} \quad \alpha = (\alpha_1, \ldots, \alpha_n)$$

are respectively the model parameter sequence and the inferred state sequence. Therefore, we need to specify a prior distribution $p(\theta)$ on the model parameter θ_1 and a distribution $p(\alpha_1|\theta_1)$ on the initial vector α_1 under model parameter θ_1. For parameters $\theta_t =$

$(\sigma_t, \sigma_\mu, \sigma_\delta)$, it is assumed that the prior distribution of the inverse variance is the following Gamma distribution,

$$\frac{1}{\sigma_t^2}\left(\frac{1}{\sigma_\mu^2}, \frac{1}{\sigma_\delta^2}\right) \sim G(a/2, b/2),$$

where $G(a, b)$ is the Gamma distribution with expectation a/b. This implies that b/a is a prior estimate of $\sigma_t^2(\sigma_\mu^2, \sigma_\delta^2)$ and the posterior distribution of the variance is the inverse gamma distribution [13,29]. The sample variance of the time series that has been observed so far is used as the prior estimate of the variance.

The initial values of parameters α_1 and θ_1 are assumed to follow the normal distribution. The initial value of the mean of μ_t is set to be y_1 because the initial value of the mean is directly related to the first observation y_1, which can be used as an initial estimate. We further set the initial value of the mean of δ_t to be $(y_n - y_1)/n$ because the difference between the last observation y_n and the first observation y_1 reflects the changing trend in the data and therefore the average difference $(y_n - y_1)/n$ roughly conveys the slope information.

3.3. Computing the posterior distribution of the BLLT model

For computing the posterior distribution, a Gibbs sampler is used to simulate a sequence $(\theta^{(1)}, \alpha^{(1)}), (\theta^{(2)}, \alpha^{(2)}), \cdots$, from a Markov Chain with stationary distribution $p(\theta, \alpha|y_{1:n})$. The sampling algorithm has two parts: the first part samples $p(\alpha|y_{1:n}, \theta)$ to derive α, and the second part samples $p(\theta|y_{1:n}, \alpha)$ to derive θ.

For using a Gibbs sampler to simulate the sequence $(\theta^{(1)}, \alpha^{(1)}), (\theta^{(2)}, \alpha^{(2)}), \cdots$, we use the algorithm described in Durbin and Koopman [12] because it is simple and computationally efficient. Specifically, given the initial parameters a_1 and P_1 as well as the initial prior density $p(\theta_1)$, where a_1 is the mean vector of α_1 and P_1 is the covariance matrix of α_1, Algorithm 1 is used to sample $\theta^{(i)}$ and $\alpha^{(i)}$.

Forecasting (i.e. prediction) is conducted using the posterior predictive distribution as follows. (i) The posterior distribution $p((\theta, \alpha)|y_{1:n})$ is readily available after simulating model parameters $(\theta^{(i)}, \alpha^{(i)})$. (ii) The distribution of $p(y_{n+1}|(\theta, \alpha))$ can be estimated based on the proposed BLLT model. (iii) The predicted distribution of y_{n+1} is

$$p(y_{n+1}|y_{1:n}) = \int p(y_{n+1}|\tau)p(\tau|y_{1:n})\,\mathrm{d}\tau, \tag{6}$$

where $\tau = (\theta, \alpha)$.

4. Simulation study

In this section, we perform a simulation study to examine the fitting and prediction performances (i.e. accuracies) of the BLLT model in different scenarios. In particular, we compare the performances of BLLT model to that of the classic SIS and SIR models of the propagation dynamics taking place on top of random and contact networks. This allows us to study whether or not the BLLT model can describe the computer malware propagation dynamics. The fitting and prediction accuracies of the BLLT model and that of the BLL model are also compared. The performance of the BLLT model when applied to data containing observation errors (incurred by misclassifications) is examined as well.

Algorithm 1 Algorithm for simulating $(\theta^{(i)}, \alpha^{(i)})$ sequence

Input: Observations $\{y_1, y_2, \ldots, \}$; initial parameters a_1 and P_1 and initial prior density $p(\theta_1)$, where a_1 is the mean vector of α_1 and P_1 is the covariance matrix of α_1.

for $i = 1$ to M **do**
  Generate a random vector $\dot{\alpha}_1^{(i)}$ from the normal distribution $N(a_1, P_1)$
  for $t = 1$ to n **do**
    Generate a random vector $\dot{\theta}_1^{(i)}$ according to probability density $p(\theta_1)$ (when t=1, θ_1^i=θ_1)
    Compute $\dot{y}_t^{(i)}$ according to Eq. (1) by replacing $\dot{\theta}_t^{(i)}$ with $\theta_t^{(i)}$
    $\hat{\theta}_t^{(i)} \leftarrow E(\theta_t^{(i)}|y_t)$ and $\hat{\dot{\theta}}_t^{(i)} \leftarrow E(\dot{\theta}_t^{(i)}|\dot{y}_t^{(i)})$ via the standard Kalman filtering and smoothing equations
    $\tilde{\theta}_t^{(i)} \leftarrow \hat{\theta}_t^{(i)} - \hat{\dot{\theta}}_t^{(i)} + \dot{\theta}_t^{(i)}$
    Compute $\alpha_{t+1}^{(i)}$ according to Eq. (2) and Eq. (3) with $\tilde{\theta}_t^{(i)}$;
    Sample $\theta_{t+1}^{(i)}$ according to the Inverse Gamma density $p(\theta_{t+1}^{(i)}|\alpha_{t+1}^{(i)}, y_{1:t+1})$
  end for
end for

Output: Simulated sequence $(\theta^1, \alpha^1), \ldots, (\theta^M, \alpha^M)$.

4.1. Benchmark models

Let $(S(t), I(t), R(t))$ represent the security state vector of a network of N nodes, where $S(t)$ is the number of susceptible nodes that are subject to infection, $I(t)$ is the number of infected nodes, and $R(t)$ is the number of recovered nodes that are no longer subject to infection. In the classic SIS model, the dynamics of worm propagation is described by two differential equations:

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{\beta S(t)I(t)}{N} + \gamma I(t),$$
$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\beta S(t)I(t)}{N} - \gamma I(t),$$

where β is called the infection rate and γ is called the recovery rate. Note that $N = S(t) + I(t)$ for any t. In the classic SIR model, the dynamics of worm propagation is described by three differential equations:

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\frac{\beta S(t)I(t)}{N},$$
$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\beta S(t)I(t)}{N} - \gamma I(t),$$
$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \gamma I(t).$$

Note that $S(t) + I(t) + R(t) = N$ for any t. These models have been widely used in the literature for modeling epidemic dynamics [34] and are therefore used as benchmark models in the present study.

4.2. Simulating the propagation dynamics over networks

In the following, we simulate the propagation dynamics over random networks and contact networks.

a) Generating synthetic dataset D_1 by simulating the SIS dynamics on a random network. For our experiment, a random network (i.e. graph) G with 1, 000 nodes and independent edge-probability .1 is generated (i.e. each pair of nodes is connected with an independent probability .1). After generating the network, we randomly select 50 nodes as infected ones, and set parameters $\beta = 0.5$ and $\gamma = 2$. Algorithm 2 is used to simulate the spreading process.

Algorithm 2 Algorithm for simulating SIS infection

Input: Random graph with 1,000 nodes; infection rate $\beta = 0.5$; recovery rate $\gamma = 2$; $T = 5$.

1: Randomly assign 50 nodes as the infected nodes, and label the state of each infected node as 1. For a susceptible node, the node state is labeled as 0.
2: **while** $t \leq T$ **do**
3: Generate the random exponential recovery times $r_1, \ldots, r_m$ with rate γ, where m is the number of infected nodes at time t. That is, $r_i = -\log(1 - u_i)/\gamma$, where the u_i's are randomly generated from the interval $(0, 1)$;
4: For each susceptible node $v \in \mathcal{S}$, randomly generate the exponential infection time l_{d_v} based on rate $d_v\gamma$, where d_v is the number of infected neighbors of node v, and $\mathcal{S}$ is set of susceptible nodes;
5: Determine what event occurs first, i.e. $t_1 = \min\{r_1, \ldots, r_m, l_{d_v}, v \in \mathcal{S}\}$;
6: **if** infection occurs **then**
7: Change the node's state from 0 to 1;
8: **else**
9: Change the node's state from 1 to 0;
10: **end if**
11: $t \leftarrow t + t_1$
12: **return** t and the nodes' states at time t.
13: **end while**

Output: The nodes' states over time T.

The simulated infection time series data, denoted by D_1, is aggregated over 500 time steps. Figure 3(a) plots the number of infected nodes over time. We observe that the number of infected nodes increases rapidly during the initial time period and then becomes relatively stable, which is consistent with the theoretic result in a more general setting [21,45]. The summary statistics show that the mean number of infected nodes over time is 783.5 and the median number of infected nodes is 950. This suggests that the infection data is skewed, which is indeed exhibited in Figure 3. The standard deviation of the numbers of infected nodes is 291.27, which indicates that there is a large variability among the

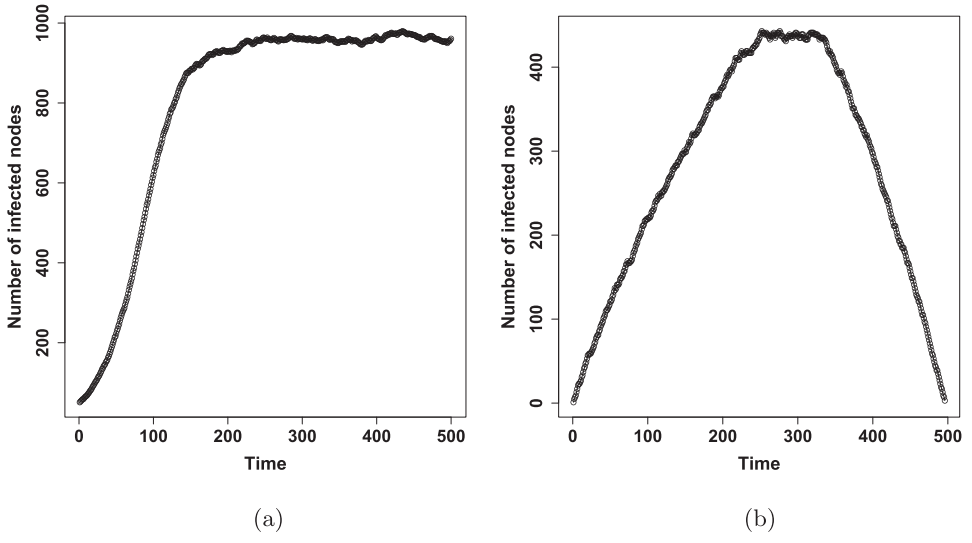

Figure 3. The simulated numbers of infected nodes over a random network. (a) SIS D_1. (b) SIR D_2.

numbers of infected nodes. This can be explained by the small number of infected nodes during the initial period but a large number of infected nodes afterwards.

b) Generating synthetic dataset D_2 by simulating the SIR dynamics on a random network. We use the same network in a) to generate a dataset for the SIR model. The initial infected node is set to be 1. The infection parameter is set to be $\beta = 0.5$ and the recovery parameter is set to be $\gamma = 10$. The R package *igraph* [23] is used to generate the infection data. Then, we aggregate the number of infected nodes over 500 time steps. Let D_2 denote the time series of the number of infected nodes over time. Figure 3(b) plots the number of infected nodes over time. We observe that the number of infected nodes increases at the beginning, then reaches the peak and lasts for a short period of time, and finally decreases to zero. This is different from the SIS simulation in a) because the recovered nodes cannot be infected anymore. The summary statistics show that the mean number of infected nodes over time is 278.7 and the median number of infected nodes is 300. The standard deviation of the numbers of infected nodes is 135.19. Again, a large variability is exhibited by the data.

c) Generating synthetic dataset D_3 by simulating the SIS dynamics on a contact network. We use the R package *EpiModel* [19] to simulate the data from the SIS model with a contact network. There are three parameters for this model: the average number of transmissible acts per node per unit time α; the probability of infection per transmissible act between a susceptible and an infected node β; the average rate of recovery with immunity γ. We set these parameters as $\alpha = 0.25$, $\beta = 0.2$, and $\gamma = 0.02$. The contact network is assumed to have 1, 000 nodes. The initial number of infected nodes is set to be 50 and therefore the initial number of susceptible nodes 950, and the time steps are set to be 500. Let D_3 denote the time series of the number of infected nodes. Figure 4(a) plots the number of infected nodes over time. We observe that there is a similarity between this curve and the curve of the simulation from the SIS model over the random network mentioned above: it

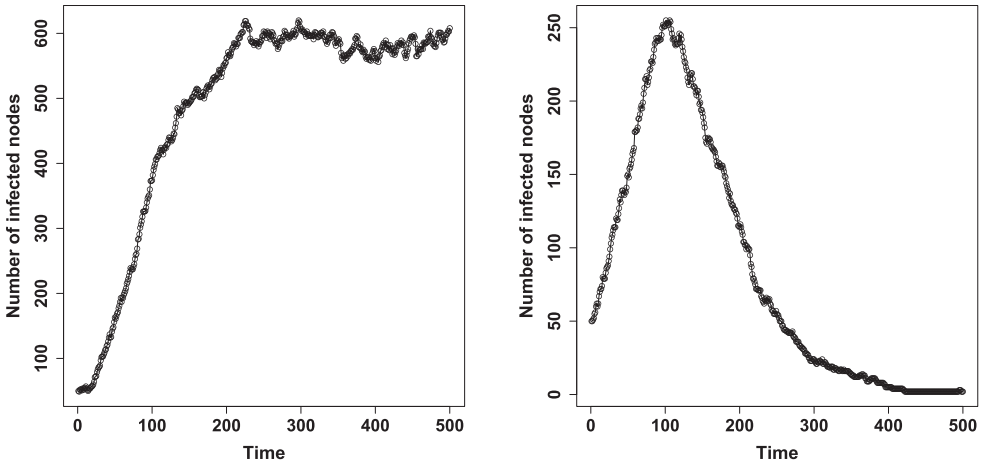

Figure 4. The simulated numbers of infected nodes over a contact network. (a) SIS D_3. (b) SIR D_4.

increases rapidly at the beginning and then comes to a steady state, which is consistent with the theoretic result in a more general setting [21,45]. The difference is that the number of infected nodes is more fluctuant in Figure 4(a) when compared to Figure 3(a). The mean and median numbers of infected nodes are 483.5 and 572.0, respectively, and the standard deviation of infected nodes is 168.27. Similarly, the data D_3 shows a large variability.

d) Generating synthetic dataset D_4 by simulating the SIR dynamics on a contact network. We use the same parameters as in c) to generate the infection data from the SIR model with a contact network by using the R package *EpiModel*. The difference is that the recovered nodes are not subject to infection for the SIR model. Let D_4 denote the time series of the number of infected nodes. Figure 4(b) plots the number of infected nodes over time. We observe that there is a similarity between this curve and the curve of the simulation for the SIR model with the random network mentioned above: the infected nodes increases at the beginning, then reaches the peak, and finally decreases to zero. The difference is that the number of infected nodes decreases directly after reaching the peak, and the decreasing trend is slower than that of Figure 3(b). The mean and median numbers of infected nodes are 83.44 and 51.50, respectively, and the standard deviation of the numbers of infected nodes is 83.46. Again, we observe a large variability from the data.

4.3. Model evaluation

In this section, we discuss the fitting and prediction performances of the proposed BLLT model. For evaluation purposes, D_i, where $i = 1, 2, 3, 4$, is divided into two evaluation periods: $D_{i,1}$ and $D_{i,2}$, where the size of $D_{i,1}$ is 80 for model fitting and the rest data of $D_{i,2}$ is used for the prediction evaluation. The prediction performance is assessed based on rolling prediction, namely that the model is updated when a new observation becomes available. Algorithm 3 shows the rolling prediction procedure. For Bayesian estimation purposes, the

Algorithm 3 Algorithm for rolling prediction of malware propagation dynamics.

Input: The number of infected nodes $\{y_i\}_{i=1,\ldots,m+n}$, where an in-sample $\{y_i\}_{i=1,\ldots,m}$ is used for fitting and an out-of-sample $\{y_i\}_{i=m+1,\ldots,n}$ is used for evaluation prediction accuracy.

for $i = m$ to $m + n - 1$ **do**
  for $t = 1$ to $10,000$ **do**
    Compute $\dot{y}_t^{(i)}$ according to Eq. (1) by replacing $\dot{\theta}_t^{(i)}$ with $\theta_t^{(i)}$
    $\hat{\theta}_t^{(i)} \leftarrow E(\theta_t^{(i)}|y_t)$ and $\hat{\dot{\theta}}_t^{(i)} \leftarrow E(\dot{\theta}_t^{(i)}|\dot{y}_t^{(i)})$ via the standard Kalman filtering and smoothing equations
    $\tilde{\theta}_t^{(i)} \leftarrow \hat{\theta}_t^{(i)} - \hat{\dot{\theta}}_t^{(i)} + \dot{\theta}_t^{(i)}$
    Compute $\alpha_{t+1}^{(i)}$ according to Eq. (2) and Eq. (3) and $\tilde{\theta}_t^{(i)}$;
    Sample $\theta_{t+1}^{(i)}$ according to the Inverse Gamma density $p(\theta_{t+1}^{(i)}|\alpha_{t+1}^{(i)}, y_{1:t+1})$
    Record $(\alpha_{t+1}^{(i)}, \theta_{t+1}^{(i)})$
  end for
  Use Eq. (6) with $(\theta, \alpha)_{t=8,001,\ldots,10,000}^{i}$ to predict the distribution of y_{i+1}.
  Use the predicted distribution to compute the mean $\hat{y}_{i+1}$.
end for

Output: Predicted number of infected nodes $\{\hat{y}_i\}_{i=m+1,\ldots,m+n}$.

MCMC steps are set to be 10,000 and the burn period is set to be 8,000. We use the standard metrics to assess the fitting and prediction accuracies: MSE (Mean Squared Error), MAD (Mean Absolute Deviation), MAPD (Mean Absolute Percentage Deviation), and SMAPE (Symmetric Mean Absolute Percentage Error) [18].

Data D_1. We fit $D_{1,1}$ by using the benchmark model, namely the SIS model, and the proposed Bayesian approach. For the Bayesian models, we examine both BLLT and BLL for comparison purposes. The fitting results are shown in Figure 5. It is observed that all the models have good fitting performances.

The prediction evaluation is performed on $D_{1,2}$, which is shown in Figure 5(b). It is seen that the benchmark model overpredicts the number of infected nodes. The BLLT and BLL models predict the infection very well. For a further comparison, we compute the prediction evaluation metrics in Table 1. It is found that the benchmark SIS model has a very large MSE value of 171.3291, while the BLL and BLLT models have much smaller MSE values. In particular, the BLLT model has the smallest MSE value of 5.7267. Similarly, the benchmark SIS model has the largest MAD value of 11.4579, while the BLLT model has the smallest MAD value of 1.9533. Furthermore, the BLLT model has the largest prediction accuracy of 99.78% based on MAPD and SMAPE, whereas the BLL model has the prediction accuracy of 99.70% based on the MAPD metric and 99.64% based on the SMAPE metric. The benchmark SIS model has lower prediction actuaries in terms of both MAPD (98.74%) and SMAPE (98.23%).

In conclusion, the proposed BLLT model significantly outperforms the other models based on all the metrics.

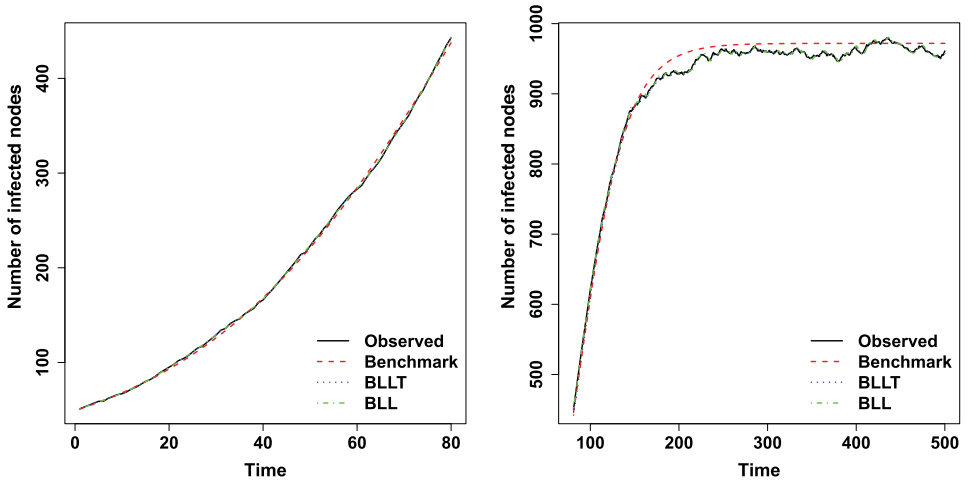

Figure 5. Fitting and prediction of different models for D_1. (a) Fitting $D_{1,1}$. (b) Predicting $D_{1,2}$.

Table 1. Prediction evaluation of different models for simulated propagation dynamics data.

Metrics	MSE	MAD	MAPD	SMAPE	MSE	MAD	MAPD	SMAPE
		$D_{1,2}$				$D_{2,2}$		
Benchmark	171.3291	11.4579	0.0126	0.0127	11406.29	98.7111	0.3146	0.3848
BLL	15.2713	2.7344	0.0030	0.0036	8.7482	2.4983	0.0080	0.0177
BLLT	5.7267	1.9533	0.0022	0.0022	3.7661	1.5910	0.0051	0.0072
		$D_{3,2}$				$D_{4,2}$		
Benchmark	14727.59	88.5749	0.1610	0.1373	15.9323	2.7154	0.0366	0.1226
BLL	25.9844	4.0308	0.0073	0.0077	3.4728	1.1506	0.0155	0.0274
BLLT	26.0779	4.0572	0.0074	0.0075	2.9264	1.0853	0.0146	0.0316

Data D_2. For D_2, we fit $D_{2,1}$ by using the benchmark SIR model, and compare it to the fitting performances of the BLL and BLLT models. The fitting results are shown in Figure 6(a). It is seen that the benchmark model has a very poor fitting performance, while the Bayesian models have much better fitting performances. The benchmark model cannot fit the data well because the real infection increases very fast even starting from one infected node, but the benchmark model fails to catch up with this rapid increase, which causes a significant underprediction of the number of infected nodes by benchmark model, as shown in Figure 6(b). The BLLT model can adapt quickly to accommodate the data and has a very good fitting and prediction performance.

For the prediction accuracy metrics on $D_{2,2}$, Table 1 shows that the benchmark model has very large MSE (11,406.29) and MAD (98.7111), whereas the BLLT model has the smallest values (MSE of 3.7661, MAD of 1.5910). In terms of the percentage accuracies, the benchmark model is only able to achieve 68.54% accuracy in MAPD and 61.52% accuracy in SMAPE. The BLLT model can significantly improve the prediction accuracies to 99.49% in MAPD and 99.28% in SMAPE.

In summary, the BLLT model has the best prediction performance for D_2.

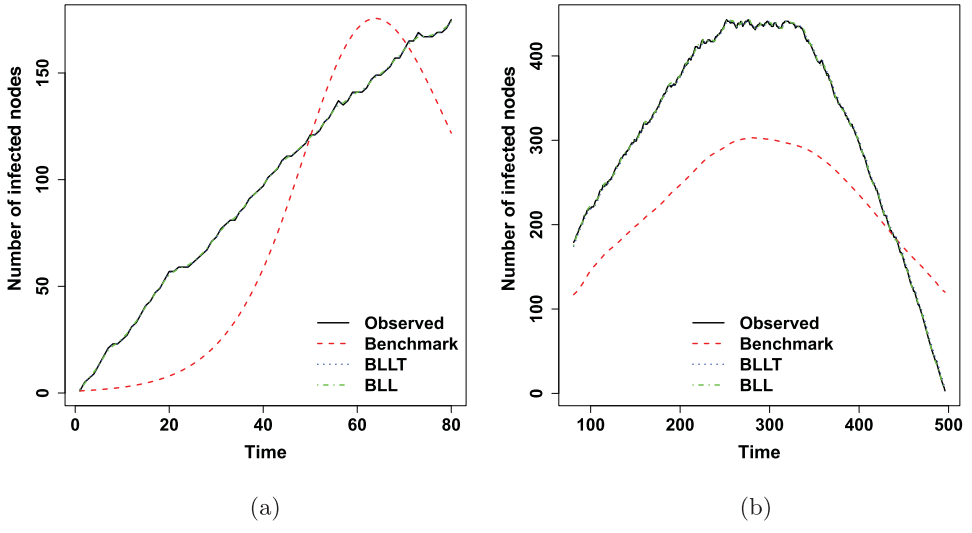

Figure 6. Fitting and prediction of different models for D_2. (a) Fitting $D_{2,1}$. (b) Predicting $D_{2,2}$.

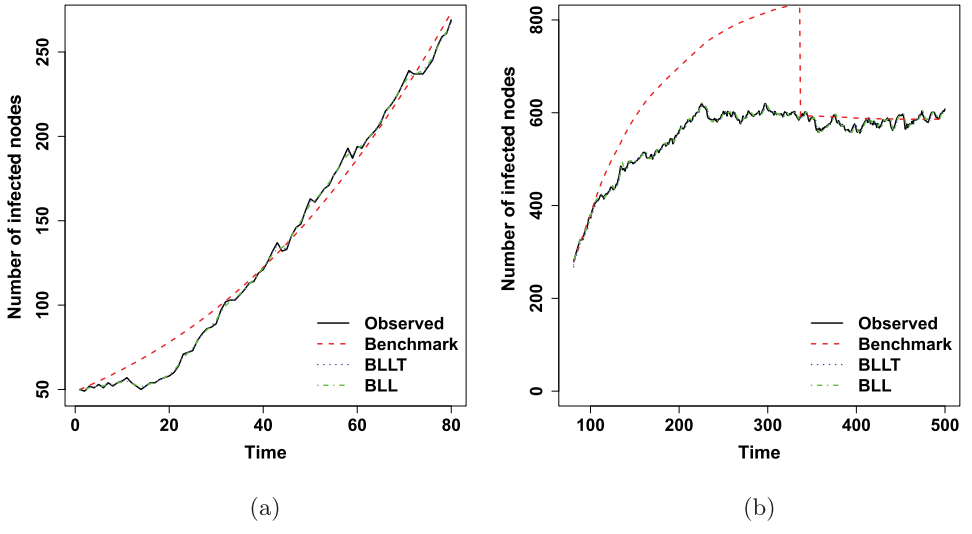

Figure 7. Fitting and prediction of different models for D_3. (a) Fitting $D_{3,1}$. (b) Predicting $D_{3,2}$.

Data D_3. The $D_{3,1}$ is fitted by the benchmark SIS model and the proposed Bayesian models. The fitting results are shown in Figure 7(a). Generally, all of the three models have good fitting performances, which are similar to the case of $D_{1,1}$. The prediction results are shown in Figure 7(b). It is observed that although the benchmark SIS model has a good fitting performance, it significantly overpredicts the number of infected nodes for $D_{3,2}$, which is mainly caused by its weak capability in quickly adapting to accommodate the changes. Both the BLL and BLLT models can predict the worm prorogation well.

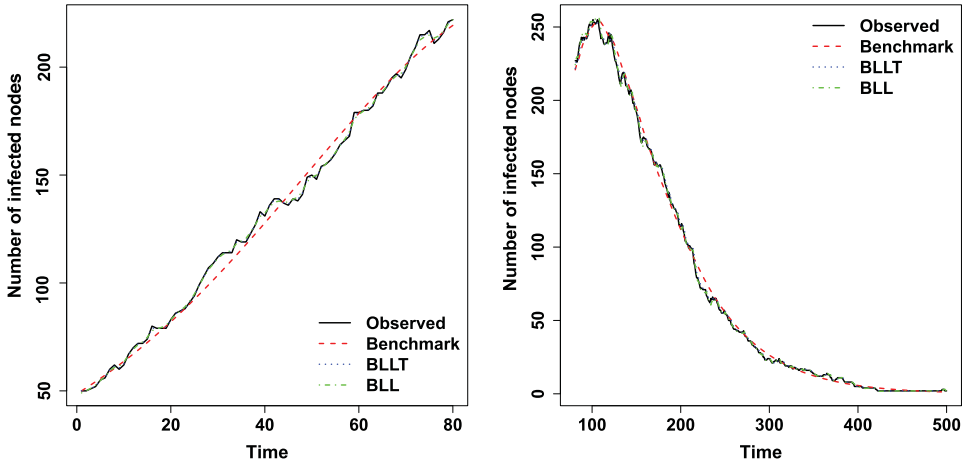

Figure 8. Fitting and prediction of different models for D_4. (a) Fitting $D_{4,1}$. (b) Predicting $D_{4,2}$.

Table 1 shows the prediction metrics for all of the models. It is seen that the prediction performance of the BLL model is comparable to that of the BLLT model. Both models have similar prediction performances, and are much better than that of the benchmark model.

Data D_4. We fit $D_{4,1}$ by using the benchmark SIR model and the proposed Bayesian models. The fitting results are shown in Figure 8(a). It is seen that the benchmark model can generally follow the increasing trend and that both Bayesian models are fitting very well. It is interesting to compare the fitting performance to that of $D_{2,1}$ by the benchmark model. The improved fitting performance of the benchmark model for $D_{4,1}$ can be attributed to the initial infection number. For $D_{4,1}$, the initial number of infected nodes is 50, which is compared to the one infected node of $D_{2,1}$. This is also the reason why the benchmark model can capture the increasing trend of $D_{4,1}$.

The prediction performances of $D_{4,2}$ are shown in Figure 8(b) and Table 1. It is seen that although three models can predict well, the BLLT model still outperforms the other models based on the MSE, MAD and MAPD metrics. The SMAPE of the BLL model is slightly smaller than that of the BLLT model. In particular, the BLLT model has the highest prediction accuracy of 98.54% in MAPD.

Data with misclassified traffics. Since the propagation dynamics data may include misclassified traffic, namely false positives and false negatives in practice, we study model performance based on the aforementioned data D_1, which is obtained by simulating the SIS model with a random network in the presence of observation errors. We assume that the number of misclassified traffic follows the Gaussian distribution with mean 0 and standard deviation 10, meaning that the misclassification includes both false-positives and false-negatives. Figure 9(a) plots the ground-truth number of infected nodes and the observed number of infected nodes, and shows that the observed number of infected nodes has a large variability owing to the false-positives and false-negatives.

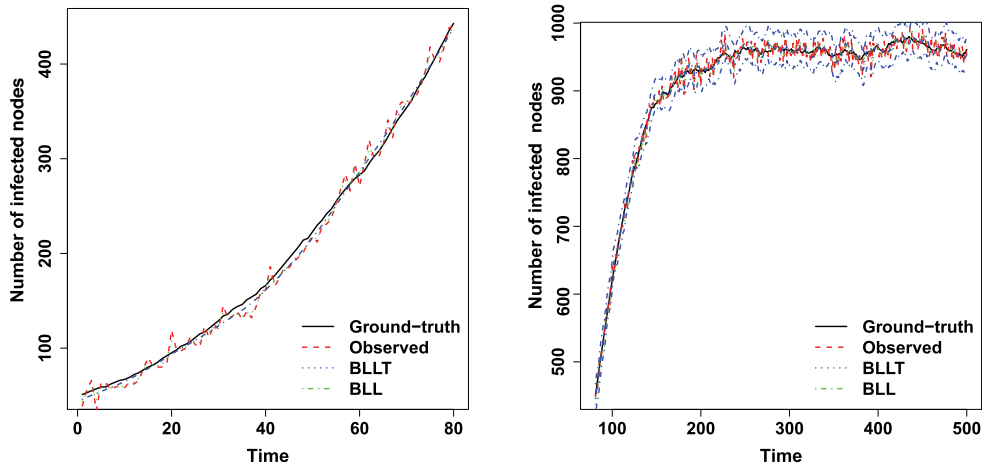

Figure 9. Using different models to fit and predict D_1 in the presence of false-positives and false-negatives, where blue curves indicate the upper and lower bounds of the 95% prediction interval. (a) Fitting $D_{1,1}$ with misclassifications. (b) Predicting $D_{1,2}$ with misclassifications.

For fitting accuracy, Figure 9(a) shows that the fitted BLLT and BLL models generally follow the increasing trend of the ground-truth, but can oscillate, owing to observation errors. For predicting accuracy with respect to $D_{1,2}$ with observation errors, Figure 9(b) shows that the observed number of infected computers has large variability caused by the false-positives and false-negatives.

Table 2 reports the prediction accuracy metrics, and shows that the BLLT model outperforms the BLL model in terms of these metrics. By comparing the prediction accuracy results reported in Tables 1, we see that false-positives and false-negatives cause the much less accurate prediction results. It is also seen that the BLLT model is more robust than the BLL model in terms of the prediction accuracies. As an advantage of the Bayesian modeling approach, the prediction interval can be easily provided. In Figure 9, we show the 95% prediction interval for the BLLT model. It can be observed that the ground-truth falls into the 95% prediction interval.

To conclude, the proposed BLLT model has an excellent fitting and prediction performances on the malware propagation dynamics based on synthetic data. It can adapt to accommodate local (transient) variations in the dynamics very quickly and produce the high accurate prediction, whereas the benchmark SIS and SIR models have the drawback of not being able to adapt to accommodate the variations exhibited by the data. Further, the

Table 2. Prediction accuracy with respect to $D_{1,2}$ in the presence of false-positive and false-negative observation errors.

	MSE	MAD	MAPD	SMAPE
BLL	167.6308	10.4146	0.0115	0.0119
BLLT	145.0861	9.6581	0.0106	0.0108

BLLT model is more robust than the BLL model. If the propagation dynamics data contains misclassifications, the proposed BLLT model can accommodate the uncertainty by providing the prediction intervals.

5. Applications

In this section, we study the fitting and prediction performances of the proposed BLLT model on the propagations of the Conficker and Code Red worms. The performances of the proposed BLLT model to the other commonly used models are compared as well. We further discuss how to use the proposed BLLT model in practice.

5.1. Conficker worm

For the Conficker worm, the first 300 observations are used for the model fitting, and the rest 1,410 observations are used for assessing prediction performance. Before evaluating the model fitting and prediction accuracies, the convergence of MCMC chain during the fitting procedure is verified to confirm that the Bayesian approach can be employed. The Gelman-Rubin approach is used to determine the convergence [5]. When the chain converges, the Gelman-Rubin statistic $\hat{R}$ should be close to 1. We test five chains with size 10, 000 and with different initial values for the parameters $(\sigma_t, \sigma_\mu, \sigma_\delta)$. The corresponding Gelman-Rubin statistics $\hat{R}$ are respectively 1.0103, 1.0124, 1.0399, which are all smaller than 1.1. This indicates that the MCMC approach is suitable for the Conficker worm data.

The fitting curve of the BLLT model is shown in Figure 10(a). For comparison purposes, the fitting of the BLL model is also displayed. It is seen that both models can fit the worm propagation data very well. Prediction is performed by the rolling approach of Algorithm 3 and plotted in Figure 10(b). It is observed that both models have very good prediction performances. The prediction accuracy metrics are reported in Table 3. It is seen that the MSE and MAD of the BLLT model are much smaller than those of the BLL model. Furthermore, the BLLT model can achieve a 99.98% prediction accuracy when compared to the 99.94% accuracy of the BLL model in terms of MAPD, and a 99.98% accuracy vs. a 99.93% accuracy in terms of SMAPE.

In Figure 11, the prediction intervals for the Conficker worm at 95% and 99% levels are displayed. Figure 11(a) shows the overall prediction intervals, and Figure 11(b) displays a zoomed part of the prediction. It can be observed that the prediction intervals are very narrow at both predictive levels.

Model comparisons. We compare the fitting and prediction performances of the BLLT model on the Conficker worm with those of the statistical models proposed in the literature, including the AutoRegressive Integrated Moving Average (ARIMA) model [32,41],

Table 3. Prediction evaluation of different models for the Conficker worm.

	MSE	MAD	MAPD	SMAPE
BLL	30291.29	127.0138	0.0006	0.0007
BLLT	5609.875	45.74324	0.0002	0.0002

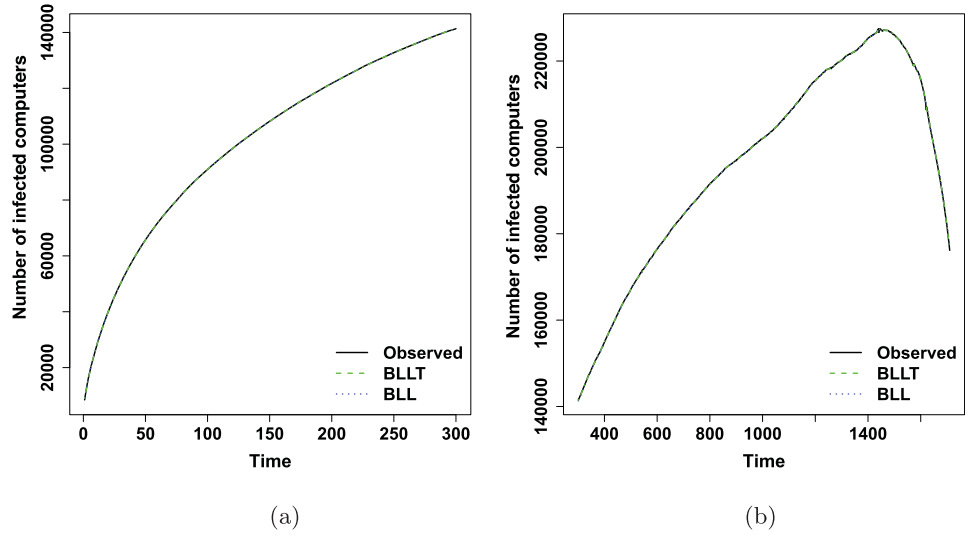

Figure 10. Fitting and prediction of different models for the Conficker worm propagation data. (a) Fitting. (b) Prediction.

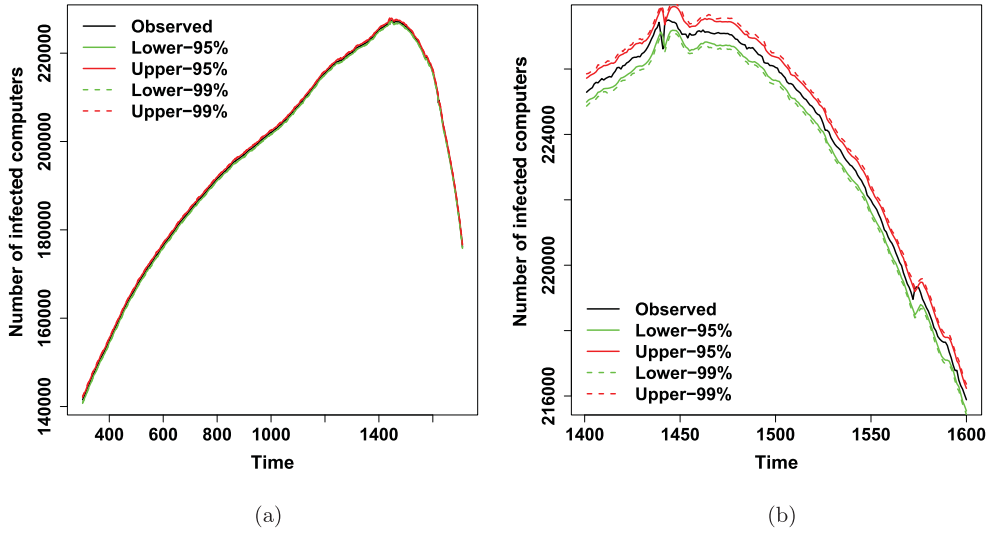

Figure 11. Prediction intervals for the Conficker worm at levels 95% and 99%. (a) Prediction intervals. (b) Zoomed prediction intervals.

the Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) model [42], and the Bayesian Negative Binomial (BNB) model [3].

Model fitting is performed on the first 300 observations. To develop an ARIMA model, the ADF test is used to determine where a unit root is present in the Conficker worm data. The p-value of ADF test is smaller than 0.01, which suggests that no difference is needed for modeling the Conficker worm data. The AIC suggests that ARIMA(1,0,1) model is sufficient for the modeling purpose. For the GARCH model, the AIC is also used to select the model, which suggests that the mean part can be modeled as ARMA(1,1) and the variability

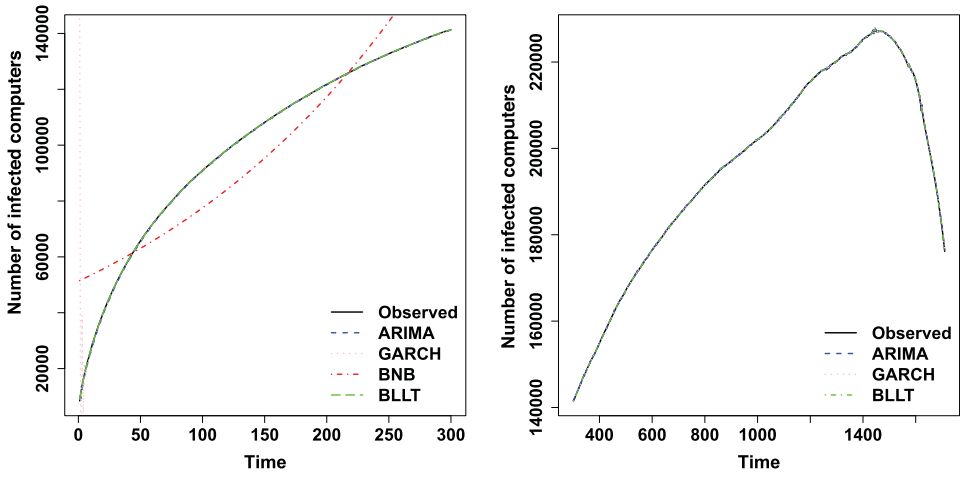

Figure 12. Comparing the fitting and prediction performances of different models based on the Conficker worm propagation data. (a) Fitting. (b) Prediction.

part can be modeled as GARCH(1,1). The BNB model [3] is also used for fitting the Conficker worm data. The fitting performances of different models are shown in Figure 12(a). It is seen that when compared to the BLLT model, the ARIMA model also has a good fitting performance; the GARCH model has a poor fitting performance for the first few observations, then shows a good fitting performance for the rest of the observations; the BNB model has a very poor fitting performance, which suggests that the Conficker worm data does not follow the negative binomial distribution. We further compare the prediction performances of the ARIMA and GARCH models to that of the BLLT model, based on the rolling prediction of rest 1410 observations. The predictions of the three models are shown in Figure 12(b), and it is seen that all of the models have good prediction performances.

Table 4 presents the prediction accuracy metrics. It is observed that the BLLT model outperforms the other models in terms of all of the metrics. Particularly, the BLLT model significantly outperforms the other models in terms of the MSE and MAD metrics.

5.2. Code red worm

Similar to the previous subsection, the first 300 observations are used for model fitting, and the rest 1, 512 observations are used for assessing the prediction performance. The Gelman-Rubin statistic is used to determine the convergence as well. We test five chains with size 10, 000 and with different initial values for the parameters $(\sigma_t, \sigma_\mu, \sigma_\delta)$. The corresponding Gelman-Rubin statistics $\hat{R}$ are respectively 1.0018, 1.0374, 1.008, which are all

Table 4. Prediction evaluation of different models for the Conficker worm.

	MSE	MAD	MAPD	SMAPE
ARIMA	16285.51	92.0162	0.0005	0.0005
GARCH	12737.8	71.0670	0.0004	0.0004
BLLT	5609.875	45.74324	0.0002	0.0002

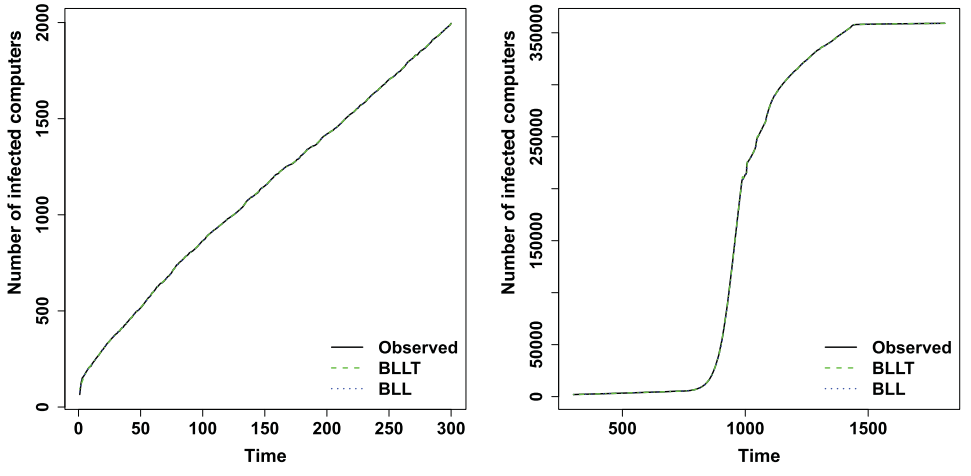

Figure 13. Fitting and prediction of different models for the Code Red worm propagation data. (a) Fitting. (b) Prediction.

smaller than 1.1. This indicates that the MCMC approach is suitable for the Code Red worm data.

The fitting plot of the BLLT model is shown in Figure 13(a). For comparison purposes, the fitting of the BLL model is displayed as well. It is again seen that both models can fit the worm data very well. The prediction is shown in Figure 13(b), and it is observed both models have a very good prediction performance. The prediction accuracy metrics are reported in Table 5. It is seen that the MSE and MAD values of the BLLT model are much smaller than that of the BLL model. The BLLT model can predict with a 99.97% accuracy, when compared to the 99.85% accuracy of the BLL model in terms of MAPD (or 99.93% vs. 99.59% accuracy in terms of SMAPE).

For the Code Red worm propagation data, the prediction intervals are shown in Figure 14 at both 95% and 99% levels. Figure 14(a) shows the overall prediction intervals, and Figure 14(b) displays the zoomed part of the prediction. Again, it can be observed that the prediction intervals are very narrow at both levels.

In summary, we conclude that the BLLT model can predict the dynamics of Code Red worm propagation very well.

Model comparisons. Similar to the Conficker worm, we perform the model comparisons as follows. The AIC suggests that ARIMA(0,1,2) model is suitable for the modeling

Table 5. Prediction evaluation of different models for the Code Red worm propagation data.

	MSE	MAD	MAPD	SMAPE
BLL	377225.7	282.9071	0.0015	0.0041
BLLT	77339.74	59.9474	0.0003	0.0007

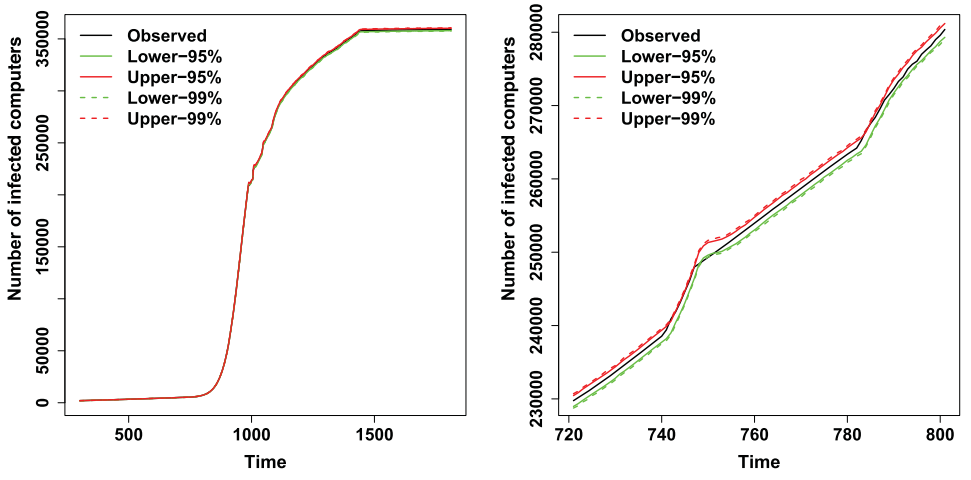

Figure 14. The prediction intervals for the Code Red worm at the 95% and 99% levels. (a) Prediction intervals. (b) Zoomed prediction intervals.

Table 6. Prediction evaluation of different models for the Code Red worm propagation data.

	MSE	MAD	MAPD	SMAPE
ARIMA	132476	108.9276	0.0006	0.0014
GARCH	146288.3	148.5909	0.0008	0.0021
BLLT	77339.74	59.9474	0.0003	0.0007

the Code Red worm data. For the GARCH model, the AIC suggests to use the ARMA(1,1)+GARCH(1,1) model. For fitting, it is found that both the BLLT and ARIMA models have good fitting performances, while the GARCH and BNB models have poor performances. This observation is similar to what is observed in the case of the Conficker worm data. Prediction results are reported in Table 6, which is based on the rolling prediction of the rest 1,512 observations. It is again observed that the BLLT model significantly outperforms the other models.

5.3. Using BLLT in practice

We propose the following three-step procedure for using the BLLT model to predict malware propagation in practice.

(1) Data collection. For modeling real-world malware propagation, the first step is to collect data, and then aggregate the data into time windows of a desired time unit (e.g. second, minute, or hour).
(2) Modeling. Use Algorithm 1 to generate a large number of simulated sequence of parameters for the BLLT model (e.g. 10, 000).
(3) Prediction. Use Algorithm 3 to predict the distribution of the future value and calculate the predictive quantities of interest.

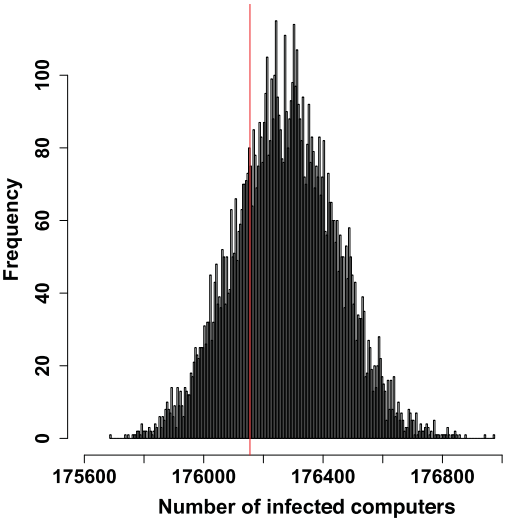

Figure 15. The histogram of the predicted number of infected computers by the Conficker worm, where the vertical line represents the real observation.

Table 7. Predicted intervals of the number of infected computers by the Conficker worm.

Interval	Prob
[175600,175800)	0.0021
[175800,176000)	0.0509
[176000,176200)	0.2765
[176200,176400)	0.4369
[176400,176600)	0.2051
[176600,176800)	0.0265
[176800,177000)	0.002

For example, for the Conficker worm data, the distribution of the last observation can be predicted. The histogram of predicted infection is shown in Figure 15. The last observation is 176, 155, which corresponds to the 24.53% percentile of the predicted distribution. In practice, one can create the prediction intervals as shown in Table 7, which would be more straightforward for practitioners to use. From Table 7, it is seen that the last observation falls into the interval [176000, 176200), which occurs with a 0.2765 probability.

In summary, the proposed BLLT model has very satisfactory fitting and prediction performances for the real-world Conficker and Code Red worm propagation data. Since it is a Bayesian model, it can naturally handle the noise in the data by providing predictive distribution.

6. Conclusion

We have proposed using the BLLT model to model the data-driven, macro-level computer malware propagation dynamics in cyberspace, without requiring the various kinds of information about the micro-level attack-defense interactions. The proposed model not only possesses the parsimony property (i.e. using only three parameters), but also can

provide the predictive distribution (i.e. accommodating uncertainty that is often encountered in practice). Both simulation and empirical studies show that the proposed Bayesian approach has satisfactory fitting and prediction accuracies. When compared with traditional time series models, the proposed approach achieves a substantially higher prediction accuracy. For the Conficker worm propagation dynamics, the prediction accuracy of the proposed approach is 65.56% higher than that of the ARIMA model and 55.96% higher than that of the GARCH model, both measured in the MSE metric; for the Code Red worm propagation dynamics, the prediction accuracy of the proposed approach is 41.62% higher than that of the ARIMA model and 47.13% higher than that of the GARCH model, both measured in the MSE metric. We hope this study will motivate more studies on statistical approaches to modeling the computer malware propagation dynamics in cyberspace.

Future studies include the investigation of computer malware propagation dynamics with complex propagation patterns. For example, the proposed model may be adjusted or extended to accommodate covariates to improve the fitting and prediction accuracies when additional information on the propagation dynamics is available (e.g. network traffic flow information).

Acknowledgments

The authors are grateful to the AE and the anonymous referees for their insightful and constructive comments, which guided them in revising and improving the paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Shouhuai Xu was supported in part by National Natural Science Foundation (NSF) Grants 1814825 and 1736209 and Army Research Office (ARO) Grant W911NF-17-1-0566. The opinions expressed in the paper are those of the authors' and do not reflect the funding agencies' policies in any sense. Peng Zhao was supported by National Natural Science Foundation of China (11871252), and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions. Taizhong Hu was supported by Anhui Center for Applied Mathematics.

References

[1] H. Andersson and T. Britton, *Stochastic Epidemic Models and Their Statistical Analysis*, Vol. 151, Springer Science & Business Media, New York, 2012.
[2] M. Bailey, E. Cooke, F. Jahanian, A. Myrick, and S. Sinha, *Practical darknet measurement. Information Sciences and Systems*, *2006 40th Annual Conference on*, March 2006, pp. 1496–1501.
[3] J.Z. Bakdash, S. Hutchinson, E.G. Zaroukian, L.R. Marusich, S. Thirumuruganathan, C. Sample, B. Hoffman, and G. Das, *Malware in the future? forecasting of analyst detection of cyber events*, J. Cybersecurity 4 (2018), pp. 1–10.
[4] K. H. Brodersen, F. Gallusser, J. Koehler, N. Remy, and S. L. Scott, *Inferring causal impact using bayesian structural time-series models*, Ann. Appl. Stat. 9 (2015), pp. 247–274.
[5] S.P. Brooks and A. Gelman, *General methods for monitoring convergence of iterative simulations*, J. Comput. Graph. Stat. 7 (1998), pp. 434–455.
[6] CAIDA. *Center for applied internet data analysis*, preprint (2019). Available at. https://www.caida.org/home/.

[7] C. Changchun Zou, L. Gao, W. Gong, and D.F. Towsley, *Monitoring and early warning for internet worms*, *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003*, Washington, DC, USA, October 27–30, 2003, pp. 190–199.
[8] C. Changchun Zou, W. Gong, and D. Towsley, *Code red worm propagation modeling and analysis*, *Proceedings of the 9th ACM conference on Computer and communications security*, ACM, 2002, pp. 138–147.
[9] Y-Z. Chen, Z-G. Huang, S. Xu, and Y-C. Lai, *Spatiotemporal patterns and predictability of cyberattacks*, PLoS. ONE. 10 (2015), pp. e0124472.
[10] G. Da, M. Xu, and S. Xu, *A new approach to modeling and analyzing security of networked systems*, *Proceedings of the 2014 Symposium on the Science of Security (HotSoS'14)*, 2014, pp. 6:1–6:12.
[11] V. Dukic, H.F. Lopes, and N.G. Polson, *Tracking epidemics with google flu trends data and a state-space seir model*, J. Am. Stat. Assoc. 107 (2012), pp. 1410–1426.
[12] J. Durbin and S.J. Koopman, *A simple and efficient simulation smoother for state space time series analysis*, Biometrika 89 (2002), pp. 603–616.
[13] J. Durbin and S.J. Koopman, *Time Series Analysis by State Space Methods*, Vol. 38, Oxford University Press, Oxford, 2012.
[14] L. Feng, L. Song, Q. Zhao, and H. Wang, *Modeling and stability analysis of worm propagation in wireless sensor network*, Math. Probl. Eng. 2015 (2015), article number 129598.
[15] A. Gelman, H.S. Stern, J.B. Carlin, D.B. Dunson, A. Vehtari, and D.B. Rubin, *Bayesian Data Analysis*, Chapman and Hall/CRC, Boca Raton, FL, 2013.
[16] Y. Han, W. Lu, and S. Xu, *Preventive and reactive cyber defense dynamics with ergodic time-dependent parameters is globally attractive*, *CoRR*, abs/2001.07958, 2020.
[17] L. Held, N. Hens, P.D O'Neill, and J. Wallinga, *Handbook of Infectious Disease Data Analysis*, Chapman and Hall/CRC, Boca Raton, FL, 2019.
[18] R.J. Hyndman and A.B. Koehler, *Another look at measures of forecast accuracy*, Int. J. Forecast. 22 (2006), pp. 679–688.
[19] S.M. Jenness, S.M. Goodreau, and M. Morris, *Epimodel: an r package for mathematical modeling of infectious disease over networks*, J. Stat. Softw. 84 (2018), article number 8.
[20] A.A. Koepke, I.M. Longini Jr, M.E. Halloran, J. Wakefield, and V.N. Minin, *Predictive modeling of cholera outbreaks in bangladesh*, Ann. Appl. Stat. 10 (2016), pp. 575–595.
[21] Z. Lin, W. Lu, and S. Xu, *Unified preventive and reactive cyber defense dynamics is still globally convergent*, IEEE. ACM. Trans. Netw. 27 (2019), pp. 1098–1111.
[22] W. Lu, S. Xu, and X. Yi, *Optimizing active cyber defense dynamics*, *Proceedings of the 4th International Conference on Decision and Game Theory for Security (GameSec'13)*, 2013, pp. 206–225.
[23] Maintainer Gábor Csárdi. Package igraph, 2018.
[24] M. Märtens, H. Asghari, M. van Eeten, and P. Van Mieghem, *A time-dependent sis-model for long-term computer worm evolution*, *Communications and Network Security (CNS), 2016 IEEE Conference on*, IEEE, 2016, pp. 207–215.
[25] D. Osthus, J. attiker, R. Priedhorsky, and S.Y. Del Valle, *Dynamic Bayesian influenza forecasting in the United States with hierarchical discrepancy (with discussion)*, Bayesian Anal. 14 (2019), pp. 261–312.
[26] R. Pastor-Satorras and A. Vespignani, *Epidemic spreading in scale-free networks*, Phys. Rev. Lett. 86 (2001), pp. 3200–3203.
[27] C. Peng, M. Xu, S. Xu, and T. Hu, *Modeling and predicting extreme cyber attack rates via marked point processes*, J. Appl. Stat. 44 (2017), pp. 2534–2563.
[28] C. Peng, M. Xu, S. Xu, and T. Hu, *Modeling multivariate cybersecurity risks*, J. Appl. Stat. 45 (2018), pp. 2718–2740.
[29] S.L. Scott and H.R. Varian, *Predicting the present with bayesian structural time series*, Int. J. Math. Model. Optim. 5 (2014), pp. 4–23.
[30] P.K. Srivastava, R.P. Ojha, K. Sharma, S. Awasthi, and G. Sanyal, *Effect of quarantine and recovery on infectious nodes in wireless sensor network*, Int. J. Sensors Wireless Commun. Control 8 (2018), pp. 26–36.

[31] Y. Wang, S. Wen, Y. Xiang, and W. Zhou, *Modeling the propagation of worms in networks: A survey*, IEEE Commun. Surveys & Tutorials 16 (2013), pp. 942–960.
[32] G. Werner, S. Yang, and K. McConky, *Time series forecasting of cyber attack intensity*, *Proceedings of the 12th Annual Conference on cyber and information security research*, ACM, 2017, pp. 18.
[33] H. Xia, L. Li, X. Cheng, X. Cheng, and T. Qiu, *Modeling and analysis botnet propagation in social internet of things*, IEEE Internet of Things J. 7 (2020), pp. 7470–7481.
[34] S. Xu, *Cybersecurity Dynamics: A Foundation for the Science of Cybersecurity*, Springer International Publishing, Cham, 2019, pp. 1–31.
[35] M. Xu, G. Da, and S. Xu, *Cyber epidemic models with dependences*, Internet. Math. 11 (2014), pp. 62–92.
[36] M. Xu, L. Hua, and S. Xu, *A vine copula model for predicting the effectiveness of cyber defense early-warning*, Technometrics 59 (2017), pp. 508–520.
[37] S. Xu, W. Lu, and H. Li, *A stochastic model of active cyber defense dynamics*, Internet Math. 11 (2015), pp. 23–61.
[38] S. Xu, W. Lu, and Z. Zhan, *A stochastic model of multivirus dynamics*, IEEE. Trans. Dependable. Secure. Comput. 9 (2012), pp. 30–45.
[39] M. Xu, K.M. Schweitzer, R.M. Bateman, and S. Xu, *Modeling and predicting cyber hacking breaches*, IEEE Trans. Inform. Forensics Secur. 13 (2018), pp. 2856–2871.
[40] V. Yegneswaran, P. Barford, and D. Plonka, *On the design and use of internet sinks for network abuse monitoring*, *Recent Advances in Intrusion Detection*, Springer, 2004, pp. 146–165.
[41] Z. Zhan, M. Xu, and S. Xu, *Characterizing honeypot-captured cyber attacks: statistical framework and case study*, IEEE Trans. Inform. Forensics Secur. 8 (2013), pp. 1775–1789.
[42] Z. Zhan, M. Xu, and S. Xu, *Predicting cyber attack rates with extreme values*, IEEE Trans. Inform. Forensics Secur. 10 (2015), pp. 1666–1677.
[43] C. Zhang, S. Zhou, and B.M. Chain, *Hybrid epidemics-a case study on computer worm conficker*, PLoS ONE. 10 (2015), pp. e0127478.
[44] D. Zhao, L. Wang, Z. Wang, and G. Xiao, *Virus propagation and patch distribution in multiplex networks: modeling, analysis, and optimal allocation*, IEEE Trans. Inform. Forensics Secur. 14 (2018), pp. 1755–1767.
[45] R. Zheng, W. Lu, and S. Xu, *Preventive and reactive cyber defense dynamics is globally stable*, IEEE Trans. Netw. Sci. Eng. 5 (2018), pp. 156–170.
[46] R. Zheng, W. Lu, and S. Xu, *Active cyber defense dynamics exhibiting rich phenomena*, *Proceedings of the 2015 Symposium on the Science of Security*, ACM, 2015, pp. 2.