Detecting Network Events by Analyzing Dynamic Behavior of Distributed Network

Haishou Ma[1], Yi Xie[2(✉)], and Zhen Wang[1]

[1] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
[2] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
xieyi5@mail.sysu.edu.cn

Abstract. Detecting network events has become a prevalent task in various network scenarios, which is essential for network management. Although a number of studies have been conducted to solve this problem, few of them concern about the universality issue. This paper proposes a General Network Behavior Analysis Approach (GNB2A) to address this issue. First, a modeling approach is proposed based on hidden Markov random field. Markovianity is introduced to model the spatio-temporal context of distributed network and stochastic interaction among interconnected and time-continuous events. Second, an expectation maximum algorithm is derived to estimate parameters of the model, and a maximum a posteriori criterion is utilized to detect network events. Finally, GNB2A is applied to three network scenarios. Experiments demonstrate the generality and practicability of GNB2A.

Keywords: Behavior analysis · Event detection · Network modeling

1 Introduction

Detecting network events has emerged as a common task in various network scenarios. Wireless Sensor Network (WSN) is applied to different fields including monitoring environment [4] and tracking targets [2], the collected sensed data is often analyzed to find interesting events. The WannaCry ransomware strikes across the globe and worm propagates throughout the mobile communication network lead to a critical problem of detecting malicious events. These diverse network scenarios carry out a uniform task of detecting network events. It watches what's happening to network, identifies the nature of network events, which is of great importance of network management.

This work is supported by the Natural Science Foundation of Guangdong Province, China (No. 2018A030313303), the Fundamental Research Funds for the Central Universities (No. 17lgjc26) and the Natural Science Foundation of China (No. U1636118).

X. Liu et al. (Eds.): ChinaCom 2018, LNICST 262, pp. 645–655, 2019.
https://doi.org/10.1007/978-3-030-06161-6_63

In order to detect network events, a lot of studies have been conducted [1]. These works mostly focus on detecting network events at a single node of network, such as at the border of network [6]. Due to the dynamic characteristics of network event, for example, a worm outbreaks in computer network [8], a single node's information may not be sufficient to identify events clearly. To overcome the limitation of single node's detection scheme, data fusion is considered in some researches, which collecting a collection of nodes' data to detect events [3,5]. However, the literatures review only considers the data features of network events, the correlation characteristics between network events and time continuity of network events are rarely applied. Moreover, the works mentioned in the literatures are applied to a specific network scenario instead of a general approach to deal with the uniform problem of detecting network events.

To solve limitations of the literatures review, this work is motivated to propose GNB2A to detect network events, which utilizes the correlation between network events and time continuity of network events. In GNB2A, a network is divided into three layers: topology layer is the actual network topology, event layer denotes the network events occur in network and behavior layer describes the external behavior of network nodes that is driven by underlying network events. Since the network events can not be measured directly, but the external behavior feature of network is measurable, therefore, event layer can be inferred from the behavior layer, the detecting task is map to an inference problem, and the goal is to infer the current event types of network nodes through measurable behavior feature. A two-layer mathematical model is introduced to model this inference problem, an observable random field denotes the measurable feature of behavior layer, and an unobservable Markov random field that describes the underlying event layer. In this work, an expectation maximum (EM) algorithm is applied to estimate parameters of model from the training data, and a maximum a posteriori (MAP) criterion is utilized to infer the event layer.

Contributions of this work include:

- Proposing GNB2A by leveraging hidden Markov random field, a spaio-temporal context is introduced to model the correlation of network events.
- Deriving algorithms for parameter estimation and network events detection.
- Evaluating the performance of GNB2A by designing three network scenarios in a simulated environment.

The rest of this paper is organized as follows. Section 2 describes GNB2A, the modeling approach and algorithms of this work are provided. In Sect. 3 experiments are designed to validate GNB2A. Finally, Sect. 4 includes the conclusion to this paper.

2 The Proposed Approach

2.1 General Network Behavior Analysis Approach

In most communication networks, a network node's behavior depends on the event it is currently encountering. For instance, a host may forward a large

number of packets to a specific destination when it is involved in a DDoS attack. Here, "forward a large number of packets to a specific destination" is a network node's behavior, while the "DDoS attack" is the event that is affecting the node. This phenomenon is common in many network scenarios. A three-layer model is used to describe the relationship of (Node, Event, Behavior). Extending to distributed scenarios, it forms a general model, as shown in Fig. 1. Hence in GNB2A, a network is divided into three layers: topology layer, event layer and behavior layer. Topology layer is network topology consists of network nodes connected by links. Event layer denotes network events occur in network, the event occurs in a node also called "hidden state" in the following, since it cannot be observed by measurement directly. Behavior layer describes the external behavior of network which is driven by the underlying network events, the behavior also called "observation" in the following since it can be measured directly.

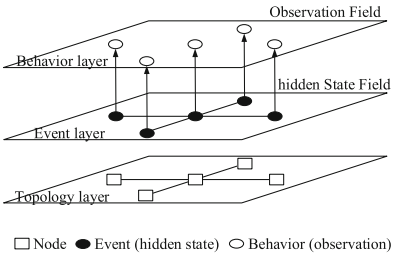


Fig. 1. The framework of GNB2A.

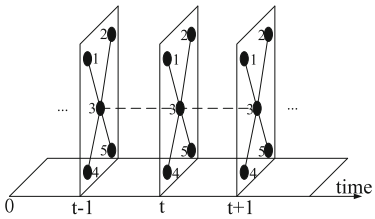


Fig. 2. Spatio-temporal context of network.

Due to the connectivity of network and the time continuity of events, an event E occurred on a node N is not only affected by N's neighbors, but also closely related to N's previous moment. To simply the modeling, this work only consider the impact of one-hop nodes, which is shown in Fig. 2, at time t, node 3 is influenced by its neighboring nodes' $\{1, 2, 4, 5\}$ events and previous event at time $t-1$. This interdependent phenomenon is called "spatio-temporal context" in this paper.

Based on the above modeling approach in GNB2A, the event layer and behavior layer of a network can be considered as a double-layer random field: a hidden State Field (SF) and an Observation Field (OF), respectively, as shown in Fig. 1. Hidden SF consists of the events of each node in network, OF represents the measurements of external behavior feature of each node in network. Thus, detecting network events from network external behavior can be mapped to infer the hidden SF from a measured OF. In the follows, how to model the relationship between hidden SF and OF and how to infer hidden SF based on a measured OF are introduced in detail.

2.2 Formulation of the Model

In network layer with N nodes, $\mathbb{N}$ denotes the set of nodes, $x_{t,n} \in \mathbb{E}$ denotes the $n^{th}(n \in \mathbb{N})$ node appearing at the t^{th} time slot, where $\mathbb{E}$ includes the collection of all $x_{t,n}$ for $\forall(t,n)$, and $|\mathbb{E}| = |\mathbb{N}| \times T$, where T denotes the number of time slot. Let $S_{t,n}$ denote the random variable of hidden state of $x_{t,n}$, i.e., the type of network event, and $s_{t,n} \in \mathbb{S}$ is an instance of $S_{t,n}$, where $\mathbb{S}$ is the set of all possible states. Then $S = \{S_{t,n} | \forall t \in [1,T], \forall n \in \mathbb{N}\}$ is a family of random variables defined on the set $\mathbb{E}$. Thus S can be used to describe the hidden SF, and $s \in \mathcal{S}$ denotes a configuration of S, where $\mathcal{S}$ is the set of all possible configurations of hidden SF. Use similar expressions, Let $O_{t,n}$ denote the random variable of observation of $x_{t,n}$, i.e., the behavior feature of network node, and $o_{t,n} \in \mathbb{O}$ is an instance of $o_{t,n}$, where $\mathbb{O}$ is the set of all possible observations. Then $O = \{O_{t,n} | \forall t \in [1,T], \forall n \in \mathbb{N}\}$ is a family of random variables defined on the set $\mathbb{E}$. Thus O can be used to describe the OF, and $o \in \mathcal{O}$ denotes a configuration of O, where $\mathcal{O}$ is the set of all possible configurations of OF.

The goal of this work is detecting network events from the external network behavior. This problem is equivalent to infer a configuration of hidden SF s given a measured OF o. According to the maximum a posteriori criterion, seeking $\hat{s}$ given o satisfies Eq. (1), where Ω denotes parameters of the model.

$$\hat{s} = \arg\max_{s \in \mathcal{S}} \{\Pr[s|o, \Omega]\} \tag{1}$$

Based on the Bayes theorem in Eq. (2):

$$\Pr[s|o, \Omega] \propto \Pr[o|s, \Omega] \cdot \Pr[s|\Omega], \tag{2}$$

The likelihood probability $\Pr[o|s, \Omega]$ in Eq. (2) describes the relationship between OF and hidden SF. In this work the conditional independent assumption is adopted to make the model solvable and tractable. Hence, $\Pr[o|s, \Omega]$ can be calculated by Eq. (3):

$$\Pr[o|s, \Omega] = \prod_{(t,n)} \Pr[o_{t,n}|s_{t,n}, \Omega]. \tag{3}$$

For the typical Gaussian distribution, the random variables of OF have the following probability density functions:

$$a_m(o_{t,n}) = \Pr[O_{t,n} = k | S_{t,n} = m, \theta_m] = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp(-\frac{(k-\mu_m)^2}{2\sigma_m^2}), k \in \mathbb{O}, m \in \mathbb{S}, \tag{4}$$

with the parameters $\theta_m = (\mu_m, \sigma_m)$, and k, m are the values of observation and hidden state, respectively. Prior probability $\Pr[s|\Omega]$ in Eq. (2) describes the interaction of hidden states between network nodes. Pseudolikelihood and first-order Markovianity are utilized to simplified the joint probability:

$$\Pr[S = s|\Omega] \simeq \prod_{(t,n)} \Pr[s_{t,n}|s_{\mathbb{N}^S_{t,n}}, s_{\mathbb{N}^T_{t,n}}, \lambda], \tag{5}$$

where $\mathbb{N}_{t,n}^S$ denotes the spatial neighboring nodes of node $x_{t,n}$, $\mathbb{N}_{t,n}^T$ denotes the corresponding one-hop temporal neighbor of node $x_{t,n}$. Based on the Hammersley-Clifford theorem, the partial probability can be written as

$$b_{t,n}(m) = \Pr[s_{t,n}|s_{\mathbb{N}_{t,n}^S}, s_{\mathbb{N}_{t,n}^T}, \lambda] = \frac{1}{Z_{t,n}(\lambda)} \exp(-U_{t,n}(m|\lambda)), m \in \mathbb{S}, \quad (6)$$

where $Z_{t,n}(\lambda) = \sum_{m \in \mathbb{S}} \exp(-U_{t,n}(m))$ and $U_{t,n}(m)$ are the marginal partition function and marginal energy function, respectively. And $U_{t,n}(m)$ has the form

$$U_{t,n}(m) = \varepsilon_{t,n} \sum_{n' \in \{\mathbb{N}_{t,n}^S, \mathbb{N}_{t,n}^T\}} V_{t,n}(m, s_{n'}), \quad (7)$$

where $\varepsilon_{t,n} = 1/(|\mathbb{N}_{t,n}^S| + |\mathbb{N}_{t,n}^T|)$ denotes the normalized factor of node energy, $|\mathbb{N}_{t,n}^S|$ and $|\mathbb{N}_{t,n}^T|$ are the number of spatial and temporal neighbor of node $x_{t,n}$, respectively. Potential function $V_{t,n}(m, s_{n'})$ defined by

$$V_{t,n}(m, s_{n'}) = \begin{cases} 0, & (s_{n'} = m) \\ \beta, & (s_{n'} \neq m) \end{cases}, n' \in \{\mathbb{N}_{t,n}^S, \mathbb{N}_{t,n}^T\}, \quad (8)$$

where β denotes the parameter correspond to the pairwise interactions between two nodes.

2.3 Algorithm

Infer hidden SF according to the MAP criterion satisfies Eq. (1), the pseudocode is shown in **Algorithm 1**.

Algorithm 1 Infer Hidden SF Algorithm

function Infer_Hidden_SF(o, Ω)
Initialize : $s^{(0)}$;
for all $x_{t,n} \in \mathbb{E}$ **do**
  for all $m \in \mathbb{S}$ **do**
    $a_m(o_{t,n}) = \Pr[O_{t,n} = k|S_{t,n} = m, \theta_m]$;
    $b_{t,n}(m) = \Pr[S_{t,n} = m|s_{\mathbb{N}_{t,n}^S}, s_{\mathbb{N}_{t,n}^T}, \lambda]$;
    $\xi_{t,n}(m) = a_m(o_{t,n}) b_{t,n}(m)$;
  end for
  $s_{t,n} \leftarrow \arg\max_{m \in \mathbb{S}} \xi_{t,n}(m)$;
end for
$\forall x_{t,n} \in \mathbb{E} : \hat{s}_{t,n} \leftarrow s_{t,n}$;
return $\hat{s}$;
end function

Input of the algorithm are the observation of network behavior o and parameter of model Ω, output is the hidden state of the network. In the initialization process (2^{nd} line), $s^{(0)}$ can be obtained by prior knowledge on OF and hidden SF. For every node in network (3^{rd} line), algorithm traverses all the potential hidden states (4^{th} line), then choose the state that has maximum probability as infer result (9^{th} line). Note that the probability of a potential hidden state includes two parts based on Eq. (2), the first part is likelihood probability in Eq. (4), denoted by $a_m(o_{t,n})$ in algorithm (5^{th} line), and the second part is partial prior probability in Eq. (6), denoted by $b_{t,n}(m)$ in algorithm (6^{th} line).

Similar to most machine learning-based applications, parameter learning is required before using model. This work uses EM algorithm to estimate parameters. The core of EM algorithm is Q function, which is defined by

$$Q(\Omega|\Omega^{(i)}) = E_s\{\ln \Pr[o, s|\Omega]|o, \Omega^{(i)}\}, \tag{9}$$

where $\Omega^{(i)}$ and Ω denote the parameter sets obtained in the i^{th} iteration and to be estimated in the $(i+1)^{th}$ iteration, respectively.

A computable form of Q function of this work is shown in Eq. (10), where $Q_A(\mu_m^{(i+1)}, \sigma_m^{(i+1)})$ and $Q_B(\beta^{(i+1)})$ denote the first term and the second term on the right-side of the second equal sign, respectively. Then the model's parameters can be estimated by maximizing $Q_A(\mu_m^{(i+1)}, \sigma_m^{(i+1)})$ and $Q_B(\beta^{(i+1)})$ independently since they are not related.

$$\begin{aligned} &Q(\Omega|\Omega^{(i)}) = E_s\{\ln \Pr[o, s|\Omega]|o, \Omega^{(i)}\} \\ &= \sum_{m\in\mathbb{S}} \sum_{t,n} \Pr[S_{t,n} = m|O_{t,n} = k, \Omega^{(i)}] \cdot \ln \Pr[O_{t,n} = k|S_{t,n} = m, \Omega] + \\ &\quad \sum_{m\in\mathbb{S}} \sum_{t,n} \Pr[S_{t,n} = m|O_{t,n} = k, \Omega^{(i)}] \cdot \ln \Pr[S_{t,n} = m|\Omega] \\ &= Q_A(\mu_m^{(i+1)}, \sigma_m^{(i+1)}) + Q_B(\beta^{(i+1)}) \end{aligned} \tag{10}$$

In this work, parameter β is obtained by empirical approach, and the parameters $\Omega = \{\mu_m, \sigma_m\}, m \in \mathbb{S}$ can be estimated by by maximizing Q_A, and obtained by the following equation:

$$\begin{cases} \mu_m^{(i+1)} = \frac{\sum_{t,n} \Pr^{(i)}[s_{t,n}|o_{t,n}] o_{t,n}}{\sum_{t,n} \Pr^{(i)}[s_{t,n}|o_{t,n}]} \\ (\sigma_m^{(i+1)})^2 = \frac{\sum_{t,n} \Pr^{(i)}[s_{t,n}|o_{t,n}](o_{t,n} - \mu_m^{(i+1)})^2}{\sum_{t,n} \Pr^{(i)}[s_{t,n}|o_{t,n}]}. \end{cases} \tag{11}$$

Algorithm 2 Parameter Estimation Algorithm

1: **function** Parameter_Estimation(o)
2: **Initialize** : $i \leftarrow 0, \Omega^{(i)} \leftarrow \{\mu_m^{(i)}, \sigma_m^{(i)}, \forall m \in \mathbb{S}\}, \mathcal{L}^{(i)} \leftarrow 0, C_{em}$;
3: **repeat**
4: $s^{(i)} \leftarrow$ Infer_Hidden_SF$(o, \Omega^{(i)})$;
5: $\{\hat{\mu}_m, \hat{\sigma}_m\} \leftarrow$ Update$(Q_A, \Omega^{(i)}, s^{(i)}), \forall m \in \mathbb{S}$;
6: $i \leftarrow i + 1$;
7: $\Omega^{(i)} \leftarrow \{\hat{\mu}_m, \hat{\sigma}_m, \forall m \in \mathbb{S}\}$;
8: $\mathcal{L}^{(i)} \leftarrow \sum_m \sum_{t,n} \ln \xi_{t,n}(m|s^{i-1}, \Omega^{(i)})$;
9: **until** $|\mathcal{L}^{(i)} - \mathcal{L}^{(i-1)}| \leq C_{em}$
10: $\hat{\Omega} \leftarrow \Omega^{(i)}$;
11: **return** $\hat{\Omega}$;
12: **end function**

Pseudocode of Parameter Estimation Algorithm is shown in **Algorithm 2**. Input of the algorithm is historical observation of network nodes, i.e. the training data of the model, output are the parameters of model. Parameters $\{\hat{\mu}_m, \hat{\sigma}_m\}$ update (5^{th} line) based on Eq. (11). C_{em} denotes the given convergence condition for the iteration of algorithm, which measures the fitting degree of the model to the training data. To control the iteration process of algorithm, let $\mathcal{L} = \sum_s \ln \Pr[o, s|\Omega]$ denote the overall logarithmic likelihood.

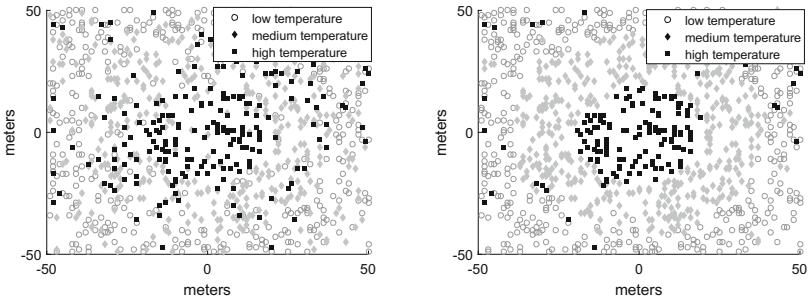
3 Experiment

In this section, three network scenarios are designed to validate GNB2A. The scenarios are selected to demonstrate GNB2A is suitable from a wireless physical network to a logical connected network.

In WSN scenario, GNB2A is applied to detect events in the environment from the unreliable sensed data. The gradual depletion of the sensors' energy or sensors are compromised by attacker, the information transmitted by the sensors is inevitably subject to a degree of unreliability and error. In this experiment, WSN monitoring environmental variables (temperature here), 1000 sensor nodes are randomly distributed in the environment, and the base temperature data originates from the Intel Berkeley Research Lab[1]. A Gaussian noise is randomly added to the temperature data to model external disturbance. Neighboring nodes of node k consist of all nodes that are within a distance d from node k.

Events in this scenario defined as {high temperature, medium temperature, low temperature}, i.e., the hidden states of nodes. The observation of a node is the sensed data, due to the unreliability of sensed data, the real event may not be perceived directly from the threshold based approach, as shown in Fig. 3(a), three states are mixture in the environment. When applied GNB2A, the events are detected, and the environment are divided into three temperature regions, detection result is shown in Fig. 3(b). In this work, Accurate Rate and Macro

[1] http://db.csail.mit.edu/labdata/labdata.html.



(a) Threshold based approach (b) GNB2A

Fig. 3. Comparison of two approaches for detecting temperature events.

F1 are selected to evaluate the performance, Accurate Rate is the fraction of all correctly estimated state instances to all instances, Macro F1 is the arithmetic mean value of F1, the higher evaluation metric represents the better performance. Performance comparison is shown in the Table 1, it indicates that GNB2A outperforms the threshold based approach in WSN scenario.

Table 1. Performance comparison

Scenario	Performance / Approach	Accurate rate	Macro F1
WSN	Threshold based	72.6%	71.1%
	GNB2A	92.7%	91.6%
Internet	Kmeans	90.1%	89.2%
	GNB2A	96.7%	96.0%
SN	Kmeans	89.6%	89.6%
	GNB2A	96.6%	96.6%

In training process, training data originates from the historical sensed data. The parameters are convergent after 9 or 10 iterations, as shown in Fig. 4, it indicates algorithm converges quickly. And the parameters $\Omega = \{\mu_m, \sigma_m\}, m \in \mathbb{S}$ in this scenario describe the attributed of event. From a statistical point of view, observations can be expressed as a Gaussian mixed model, three Gaussian distribution represent three types of temperature event, as shown in Fig. 5. Due to the unreliability of sensed data, the sensed data of different temperature event are overlapping, it can not be distinguished accurately by a threshold based approach. By utilizing spatio-temporal context of network, GNB2A can achieve better performance in detecting events.

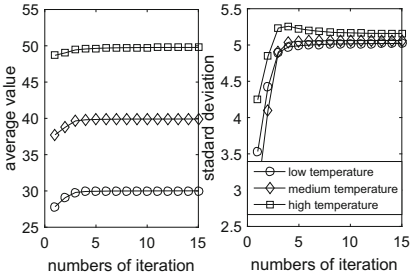


Fig. 4. Convergence of parameters.

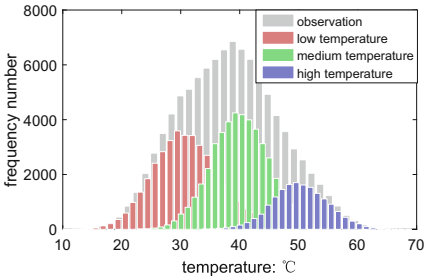


Fig. 5. Distribution of observations.

In Internet scenario, GNB2A is applied to detect network state during a DDoS attack. DDoS attack is simulated in MATLAB. A small world network with 128 nodes and 257 links is generated, some nodes are chosen as botnet, sending packets to the victim simulates a SYN flood attack. The observation of a node in this scenario is entropy of destination IP address and the arrival rate of packets. Network events define as abnormal status of the traffic passing the network node, three discrete states are used to describe the abnormal status of traffic during a DDoS attack, i.e. $\{S_1,S_2,S_3,\}$, S_1 denotes the low abnormality, and S_3 denotes high abnormality.

The training data comes from synthetic traffic data, convergence of parameters and distribution of observations are similar to WSN scenario. Table 1 shows the performance comparison, it indicates that GNB2A outperforms Kmeans approach. GNB2A can detect the abnormality of network nodes effectively. The reason for the performance gain is that GNB2A combines spatial and temporal neighbors' states, which gets more information to detect network events.

In Social Network (SN) scenario, GNB2A is applied to detect the spammers. Considering a Short Message Service (SMS) worm propagates in social network, when a user gets infected (spammer), it sends SMS spam to others, and meanwhile the worm propagates via user's contact list. In this scenario, a scale free network with 128 nodes and 253 links is built, SMS worm propagation is modeled by Susceptible-Infected (SI) model and a hierarchical infection probability is used. The features of user according to the previous analysis [7], observation of "average SMS text length" is selected, the hidden states of a user are defined as {spammer, non-spammer}. Apply GNB2A, the worm outbreak is discovered by inferring from users' behavior.

Convergence of parameters and distribution of observations are similar to WSN scenario. In test process, when applied Kmeans approach, it may not be sufficient to estimate whether a user is spammer or not based on the user's behavior feature. While GNB2A combines user's previous and neighboring users' information to detect spammers, it gets more information and therefore better performance, Table 1 shows the performance comparison. GNB2A can tract the

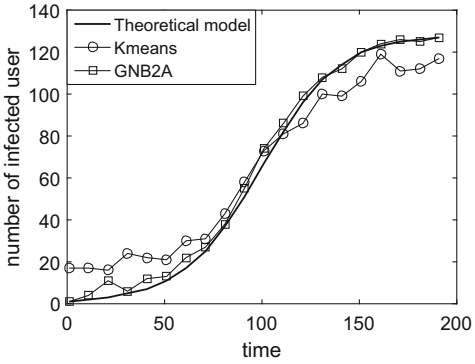


Fig. 6. The propagation of social network worm.

dynamic propagation of worm, as shown in Fig. 6, GNB2A is approximate to theoretical SI model in comparison to Kmeans approach. The result shows that GNB2A outperforms the Kmeans approach.

4 Conclusion

This work focuses on detecting network events, GNB2A is proposed to solve this problem, a hidden Markov random field is introduced to model the relationship between behavior layer and event layer of network, correlation characteristics between network events and time continuity of events are modeled by first-order Markovianity. An expectation maximum algorithm is derived to estimate parameters, and a maximum a posteriori criterion is utilized to detect network events. Experimental results demonstrate the generality and practicability of GNB2A.

References

1. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutor. **18**(2), 1153–1176 (2016). https://doi.org/10.1109/COMST.2015.2494502
2. Demigha, O., Hidouci, W.K., Ahmed, T.: On energy efficiency in collaborative target tracking in wireless sensor network: a review. IEEE Commun. Surv. Tutor. **15**(3), 1210–1222 (2013). https://doi.org/10.1109/SURV.2012.042512.00030
3. Khaleghi, B., Khamis, A., Karray, F.O., Razavi, S.N.: Multisensor data fusion: a review of the state-of-the-art. Inf. Fusion **14**(1), 28–44 (2013). https://doi.org/10.1016/j.inffus.2011.08.001, http://www.sciencedirect.com/science/article/pii/S1566253511000558
4. Othman, M.F., Shazali, K.: Wireless sensor network applications: a study in environment monitoring system. Procedia Eng. **41**, 1204–1210 (2012)

5. Ramaki, A.A., Amini, M., Atani, R.E.: Rteca: real time episode correlation algorithm for multi-step attack scenarios detection. Comput. Secur. **49**, 206–219 (2015). https://doi.org/10.1016/j.cose.2014.10.006, http://www.sciencedirect.com/science/article/pii/S0167404814001527
6. Wu, S., Liu, S., Lin, W., Zhao, X., Chen, S.: Detecting remote access trojans through external control at area network borders. In: 2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), pp. 131–141 (2017). https://doi.org/10.1109/ANCS.2017.27
7. Xu, Q., Xiang, E.W., Yang, Q., Du, J., Zhong, J.: SMS spam detection using noncontent features. IEEE Intell. Syst. **27**(6), 44–51 (2012). https://doi.org/10.1109/MIS.2012.3
8. Zhou, C.V., Leckie, C., Karunasekera, S.: A survey of coordinated attacks and collaborative intrusion detection. Comput. Secur. **29**(1), 124–140 (2010). https://doi.org/10.1016/j.cose.2009.06.008, http://www.sciencedirect.com/science/article/pii/S016740480900073X