Computer Communications 206 (2023) 160–171

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Game-theoretic approach to epidemic modeling of countermeasures against future malware evolution

Hideyoshi Miura [a,*], Tomotaka Kimura [b], Hirohisa Aman [c], Kouji Hirata [d]

[a] *Graduate School of Science and Engineering, Kansai University, Japan*
[b] *Faculty of Science and Engineering, Doshisha University, Kyoto, Japan*
[c] *Center for Information Technology, Ehime University, Japan*
[d] *Faculty of Engineering Science, Kansai University, Japan*

ARTICLE INFO

Keywords:
Game theory
Evolutionary game
Epidemic model
Botnet malware
Future malware

ABSTRACT

Recently, vulnerability mining techniques that discover unknown vulnerabilities based on machine learning have been attracted much attention for protecting software. Although we benefit from these techniques for cyber security, they could be exploited by malicious attackers. For example, the literature has introduced a concept of future malware exploiting vulnerability mining techniques. It discovers vulnerabilities of hosts by performing vulnerability mining with the use of the computing resources of hosts infected with the malware. In this paper, we propose a game-theoretic approach to epidemic modeling for discussing how to counter such future malware evolution. In the proposed approach, we consider a countermeasure model that constructs a countermeasure group aiming to discover vulnerabilities earlier than malware or malicious attackers, and repair them to protect hosts not to get infected with the malware. This paper provides stochastic epidemic modeling for the countermeasure model, which represents the infection dynamics of future malware based on a continuous-time Markov chain under countermeasure environments. Furthermore, we apply evolutionary games on complex networks to the epidemic model in order to represent the selfish behavior of hosts participating in the countermeasure group. Through simulation experiments, we reveal strategies to efficiently counter the future malware evolution.

1. Introduction

In recent years, malware such as computer viruses, worms, ransomwares, botnets, and Trojan horses has become a serious threat [1, 2]. Botnets [3–5] have especially inflicted enormous damage on the Internet, which cause cyber attacks, e.g., distributed denial of service attacks, spam dissemination, and malicious payload delivery. Malicious attackers exploit security vulnerabilities in software of hosts, performing these cyber attacks. They often perform zero-day attacks that take advantage of the time lag between the discovery of the vulnerability and the release of a security patch for it. Therefore, it is very important to appropriately find and repair security vulnerabilities in order to protect hosts.

In order to protect software against these attacks, vulnerability mining techniques that automatically discover unknown vulnerabilities based on machine learning techniques have been introduced in the past [6–10]. The main purpose of these works is to develop efficient methods in protecting software. However, malicious attackers could exploit these techniques to discover vulnerabilities and perform illegal attacks. Furthermore, they could incorporate the technologies into current evolving malware, making more serious malware. For example, the literature [11–13] has introduced metamorphic malware that rewrites and obfuscates its source codes and [14–16] has discussed ways to generate new malware by combining known malware source codes. Moreover, distributed machine learning techniques using the computing resources of inexpensive computers have been developed [17–19]. Integrating the idea of these evolving malware and distributed machine learning with vulnerability mining techniques will generate future malware whose threats will become far more serious than ever before.

Based on these facts, the authors in [20] have introduced a new concept of botnet malware named self-evolving botnets that will appear in the near future. The self-evolving botnets assume to perform vulnerability mining using distributed machine learning exploiting the computing resources of zombie computers, namely, hosts getting infected with the botnet malware. By doing so, the botnet malware can discover unknown vulnerabilities and perform zero-day attacks against non-infected hosts to newly infect them via the discovered vulnerabilities. The botnet malware takes in newly infected hosts,

[*] Corresponding author.
E-mail addresses: k846996@kansai-u.ac.jp (H. Miura), tomkimur@mail.doshisha.ac.jp (T. Kimura), aman@ehime-u.ac.jp (H. Aman), hirata@kansai-u.ac.jp (K. Hirata).

https://doi.org/10.1016/j.comcom.2023.05.001
Received 28 November 2022; Received in revised form 17 March 2023; Accepted 1 May 2023
Available online 6 May 2023

and therefore makes itself bigger, which results in the increase in its infectivity and the enhancement of vulnerability mining performance. By repeating vulnerability discovery and infection, the botnet malware evolves autonomously. In [20], the authors have provided stochastic epidemic modeling of self-evolving botnets based on a continuous-time Markov chain. Through numerical and simulation experiments, they revealed the tremendous infectivity of self-evolving botnets compared with conventional cyber attacks.

In order to protect future Internet, it is absolutely essential to develop countermeasure strategies against such future malware evolution. In this paper, we therefore propose a game-theoretic approach to epidemic modeling of future malware under countermeasure environments. In the proposed approach, we consider a countermeasure model that constructs a countermeasure group performing vulnerability mining similar to self-evolving botnets, but it uses the computing resources of hosts participating in the group [21,22]. The countermeasure group aims to discover unknown vulnerabilities earlier than malware or malicious attackers, and repair them to protect non-infected hosts not to get infected with the malware. This paper provides stochastic epidemic modeling for the countermeasure model, which represents the infection dynamics of future malware based on a continuous-time Markov chain under countermeasure environments where a countermeasure group exists in a network. In this paper, we consider two scenarios for the countermeasure model to counter future malware evolution. Furthermore, we apply the concept of evolutionary games on complex networks [23] to the epidemic model in order to represent the selfish behavior of hosts. Specifically, we suppose situations where hosts change their strategies (i.e., joining or leaving the countermeasure group) based on the evolutionary games with the use of payoffs of the strategies. By applying the concept of evolutionary games on networks, we can model the decision-making of each host that decides whether the host joins/leaves the countermeasure group. Through simulation experiments, we reveal countermeasure model scenarios to efficiently counter the future malware evolution

The contributions of this paper are as follows.

(1) We introduce epidemic modeling of countermeasures to counter future malware evolution.
(2) We incorporate the decision-making of each host with game theory into the future malware epidemic model based on a continuous-time Markov chain.
(3) We provide two different countermeasure model scenarios: ally model and volunteer model. Through simulation experiments, we reveal the characteristics of these models for suppressing the malware spreading.

The rest of this paper is organized as follows. In Section 2, we review related works. In Section 3, we explain the proposed model. Section 4 explains evolutionary games on networks in the proposed model. In Section 5, we discuss the results of simulation experiments. Section 6 provides some discussions on our countermeasure model. We conclude this paper in Section 7.

2. Related works

2.1. Epidemic model

There exist many works that discuss epidemic models of malware in the Internet. The epidemic models are categorized into the deterministic model and the stochastic model. The deterministic model represents the dynamics of malware propagation with ordinary differential equations, similar to traditional epidemiology-based models representing communicable diseases propagation such as a Susceptible–Infected–Recovered (SIR) epidemic model [24]. For example, Xiao et al. [25] have proposed a deterministic model in order to analyze the behavior of worm malware in Wi-Fi environments. On the other hand, the stochastic model represents the stochastic behavior of hosts, using some ways such as simulation experiments and Markovian analysis. For example, Okamura et al. [26] have introduced stochastic malware propagation models, including a Susceptible–Infected–Susceptible (SIS) model and a Kill-Signal (KS) model, which are represented by a continuous-time Markov chain. The SIS model is a simple epidemic model to characterize the malware propagation. The KS model considers situations where warning signals named kill signal are used to remove malware.

Some epidemic models consider network structures such as the degree and the adjacent state of nodes, which affect the malware propagation. Karyotis [27] has proposed a stochastic SIS model that considers a malware propagative chain network and represents malware spreading with a Markov random field. Yang et al. [28] have introduced a deterministic epidemic model using a heterogeneous node-based Susceptible–Infected–Recovered–Susceptible (SIRS) model. They have considered situations where nodes in a network have different infection rates and investigated the impact of the network structure on the malware spreading. Similarly, Qu and Wang [29] have introduced a stochastic SIS epidemic model with heterogeneous infection rates. Through simulation experiments based on a Markov chain, they have examined the impact of infection rates. Furthermore, some researches consider epidemic models for mobile wireless networks. Peng et al. [30] have presented a stochastic epidemic model to observe the propagation of smartphone worms on social relationship graphs based on a semi-Markov process. Shen et al. [31] have proposed a deterministic epidemic model for mobile wireless sensor networks. They have provided differential equations to represent node states while considering the mobility of sensor nodes. Conzalez et al. [32] have proposed a stochastic model to analyze the propagation of Bluetooth worms in smartphones, which is based on cellular automata and epidemiological compartmental models.

In order to suppress the malware spreading, some models have been discussed recently. Ren et al. [33,34] have proposed a deterministic epidemic model to counter malware spreading, assuming the deployment of honeypots, which are security resources to analyze the attacks of malware by being attacked on purpose. The authors in [21,22] have introduced the concept of countermeasure models that consist of a countermeasure group to protect hosts by discovering unknown vulnerabilities prior to malware and malicious attackers. They have provided stochastic and deterministic epidemic models under situations where the countermeasure group works and hosts randomly join/leave the countermeasure group.

2.2. Cyber attacks with machine learning

With the recent development of machine learning, especially deep learning [35], cyber attacks using machine learning techniques have appeared [36,37]. As an example of such cyber attacks, Shi et al. [38] have presented an approach of spoofing wireless signals by using generative adversarial networks (GAN) to imitate users' behaviors in a target network. The authors in [39,40] have proposed ways to evade the detection by malware detection systems, using GAN. Kudo et al. [20] have introduced the concept of self-evolving botnets that perform vulnerability mining using distributed machine learning exploiting the computing resources of zombie computers in order to discover vulnerabilities of hosts. Through numerical and simulation experiments based on a continuous-time Markov chain, they revealed the tremendous infectivity of self-evolving botnets. In the near future, cyber attacks using machine learning will further evolve and become more serious threats. Thus, it is very important to counter such evolution of cyber attacks. In our proposed approach in this paper, we consider a game-theoretic approach based on countermeasure models discussed in [21,22] to counter future malware evolution.

2.3. Game theory for cyber security problems

Game theory has been commonly used for modeling and resolving security problems. Dijk et al. [41] have proposed a FLIPIT game, which is applied to the situation where attackers and defenders scramble for computing resources on a network. In [42,43], the authors have presented how to apply reinforcement learning to the FLIPIT game. They analyze the optimal behavior of network defenders by learning the timing at which the attacker attacks. Spyridopoulos et al. [44] have provided a unified malware proliferation model, which clarifies the optimal behavior of each host by applying ordinary differential equations and game theory. The FLIPIT game provides a simple and elegant framework that considers the interaction between attackers and defenders in practical scenarios.

In addition, evolutionary games on networks have been used to analyze social interactions. Evolutionary games provide a framework to study strategic decision-making in various varieties of complex networks [23]. Antal et al. [45] have investigated the population evolves in two species with different advantages in heterogeneous networks. Eksin [46] has analyzed the rational behavior of hosts against infectious disease propagation over a network. Kabir and Tanimoto [47] have suggested a novel framework for a vaccination game based on a Susceptible–Vaccinated–Infected–Recovered (SVIR) epidemic model.

Furthermore, after the emergence of COVID-19 epidemic, epidemic models considering the behavior of people have been actively discussed [48–51]. These models adopt evolutionary game theory to represent the behavior of people, e.g., whether they take a cooperative strategy imposed by a government to suppress the spreading of COVID-19. In fact, adopting evolutionary game theory enables us to deeply discuss the effect of cooperation of hosts on malware spreading. Thus, in this paper, we use evolutionary game theory to represent the selfish behavior of hosts, i.e., whether the hosts cooperate in the countermeasure group.

3. Stochastic epidemic modeling for the countermeasure model

In this paper, we assume self-evolving botnets as an example of future malware and apply our game-theoretic approach to the epidemic model of the self-evolving botnets. Note that the idea of our proposed approach can be applied to various malware epidemic models. In what follows, we first briefly explain the epidemic model of self-evolving botnets without any countermeasure strategies. We then discuss the countermeasure model assumed in this paper. We further explain the epidemic model of self-evolving botnets under countermeasure environments where the countermeasure model works.

3.1. Epidemic model for self-evolving botnets [20]

We here explain the stochastic epidemic model of self-evolving botnets introduced in [20], which does not consider any countermeasure strategies. The epidemic model represents the infection dynamics, assuming that the botnet malware spreads on a given overlay network consisting of hosts. It constructs a continuous-time Markov chain based on a variant of SIRS model that represents the state transitions of each host. In the SIRS model, each host in the network has the three states: Susceptible (S), Infected (I), and Recovered (R), as shown in Fig. 1. “S” indicates that the host has some vulnerabilities. “I” indicates that the host is infected with the botnet malware. “R” indicates the host has no known vulnerabilities.

There are four transitions in the SIRS model. When a susceptible host gets infected by attacks of an infected host, the state of the host transitions from S to I, which means that the host is embedded in the botnet malware. When an infected host eliminates the botnet malware from itself, the state of the host transitions from I to R. When a susceptible host repairs its vulnerabilities, the state of the host transitions from S to R. When the botnet malware discovers a new vulnerability, all recovered hosts transition from R to S. The vulnerability discovery capability of the botnet malware increases with the number of infected hosts because it discovers vulnerabilities using the computing resources of the infected hosts. In the stochastic epidemic model introduced in [20], the infection dynamics of self-evolving botnets is represented by a continuous-time Markov chain where these events of the SIRS model occur according to a Poisson process.

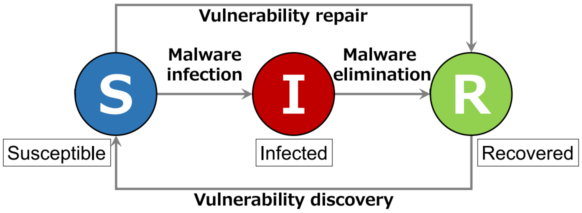


Fig. 1. SIRS model.

3.2. State transitions of each host

Taking the tremendous infectivity of future malware such as self-evolving botnets into account, it is difficult for each host to counter against the future malware individually. Thus, this paper adopts a countermeasure model that extends the concept discussed in [21,22]. In the countermeasure model, we resist the future malware, utilizing the computing resources of hosts in a group of a network. We call this group *countermeasure group* and a host participating in the group *countermeasure host*. In this paper, we construct the countermeasure model under the following assumptions:

(i) There exist K hosts and one countermeasure group.
(ii) Any hosts can join/leave the countermeasure group.
(iii) The countermeasure group discovers new vulnerabilities by performing distributed machine learning using the computing resources of countermeasure hosts.
(iv) The information on new vulnerabilities discovered by the countermeasure group is shared with hosts.
(v) An overlay network consisting of all the hosts is constructed.
(vi) Each host determines whether it joins/leaves the countermeasure group based on evolutionary games on the overlay network.

Furthermore, for the assumption (iv), we consider the following two scenarios:

(I) *Ally model scenario*: The newly discovered vulnerability information is shared with only the countermeasure hosts.
(II) *Volunteer model scenario*: The newly discovered vulnerability information is shared with all the hosts in the network.

The ally model scenario assumes that non-countermeasure hosts cannot receive the vulnerability information discovered by the countermeasure group. Therefore, those hosts do not have efficient ways to counter against the botnet malware. In this scenario, only countermeasure hosts can resist the future malware. In contrast, the volunteer model scenario assumes that all the hosts receive benefits from the countermeasure group, regardless of whether they are countermeasure hosts or not. We compare these two scenarios in this paper.

In this paper, we assume that hosts are commonly used computers of users on the Internet, and they perform vulnerability mining in the form of a public distributed computing system. In this scenario, user hosts can freely join/leave the countermeasure group, and our proposed countermeasure model adopts the game theory to represent decision-making of user hosts. By using evolutionary games, we can examine how user hosts act against infection spreading of serious malware and what factors affect users’ decision on joining/leaving the countermeasure group.

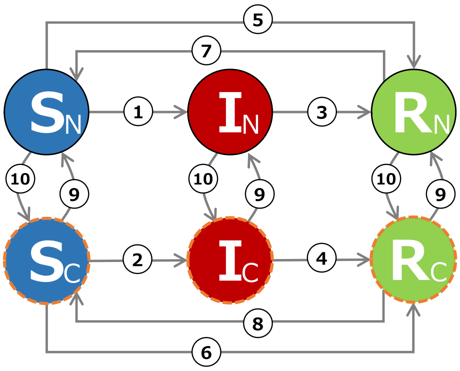


Fig. 2. The state transition of each host in the countermeasure model.

In Fig. 2, we show the state transitions of each host for the countermeasure model. In this model, each host can have six states: "$\mathrm{S_N}$", "$\mathrm{S_C}$", "$\mathrm{I_N}$", "$\mathrm{I_C}$", "$\mathrm{R_N}$", and "$\mathrm{R_C}$". The state $\mathrm{S_N}$ (resp. $\mathrm{I_N}$ and $\mathrm{R_N}$) represents that the host is a non-countermeasure host in the susceptible state (resp. infected and recovered states). On the other hand, the state $\mathrm{S_C}$ (resp. $\mathrm{I_C}$ and $\mathrm{R_C}$) represents that the host is a countermeasure host in the susceptible state (resp. infected and recovered states). The transitions between these states occur according to the following events:

(1) A susceptible host gets infected by attacks of an adjacent infected host on the overlay network and is embedded in the botnet malware (①, ② in Fig. 2). In this case, the host transitions to the infected state.
(2) An infected host eliminates the botnet malware from itself (③, ④). In this case, the host transitions to the recovered state.
(3) A susceptible host repairs its vulnerabilities (⑤, ⑥). In this case, the host transitions to the recovered state.
(4) The botnet malware discovers a new vulnerability. In this case, hosts in the recovered state transition to the susceptible state (⑦, ⑧).
(5) A host joins/leaves the countermeasure group based on evolutionary games on the overlay network where the host compares its strategy with the strategies of adjacent hosts on the overlay network (⑨, ⑩).

The occurrences of the malware propagation in event (1) and the review of the strategy (i.e., joining/leaving the countermeasure group) in event (5) depend on relationships among hosts such as their friendships, frequently accessed websites, and network environments to which they connect. Therefore, in this paper, we consider an overlay network consists of all hosts. On the overlay network, infected hosts can infect only adjacent susceptible hosts as described in event (1). Similarly, hosts change their strategies according to adjacent hosts on the overlay network as described in event (5).

3.3. Description by a continuous-time Markov chain

Let $\mathcal{K} = \{1, 2, \dots, K\}$ denote a set of hosts. Each host belongs to one of the states, i.e., "$\mathrm{S_N}$", "$\mathrm{S_C}$", "$\mathrm{I_N}$", "$\mathrm{I_C}$", "$\mathrm{R_N}$", or "$\mathrm{R_C}$". We define a set Ω of system states, which are combinations of the states of hosts, as

$$\Omega = \{(x_1, x_2, \dots, x_K); \quad x_i \in \mathcal{X}, i \in \mathcal{K}\}, \tag{1}$$

where $\mathcal{X} = \{\mathrm{S_N}, \mathrm{S_C}, \mathrm{I_N}, \mathrm{I_C}, \mathrm{R_N}, \mathrm{R_C}\}$ and x_i denotes the state of host i. This paper describes the infection dynamics of the botnet malware in countermeasure environments as a continuous-time Markov chain on Ω, under the following assumptions (a)-(e), which correspond to the events (1)–(5) described in Section 3.2, respectively.

(a) Each infected host infects an adjacent susceptible host on the overlay network according to a Poisson process with the infection rate α.
(b) Each infected host eliminates its botnet malware according to a Poisson process with the malware elimination rate β.
(c) The user of each susceptible host repairs vulnerabilities of the host according to a Poisson process with the repair rate δ.
(d) The botnet malware discovers a vulnerability according to a Poisson process with the vulnerability discovery rate $f(n)$ when the number of infected hosts is n ($n = 1, 2, \dots$), where $f(n)$ is a function of n, which determines the vulnerability discovery capability of the botnet malware.
(e) An evolutionary game is done according to a Poisson process with the executing rate θ to determine whether a host joins/leaves the countermeasure group.

In assumption (d), the botnet malware discovers a new vulnerability with the computing resources of infected hosts. Therefore, the vulnerability discovery capability of the botnet malware increases with the number n of infected hosts. We represent this with the function $f(n)$, which will be defined later.

3.4. System state transition

In what follows, we describe the system state transition based on the Markov chain in detail. Let $\mathcal{A}_i$ denote a set of adjacent hosts of host $i \in \mathcal{K}$ on the network. Let $h(X)$ ($X \in \mathcal{X} = \{\mathrm{S_N}, \mathrm{S_C}, \mathrm{I_N}, \mathrm{I_C}, \mathrm{R_N}, \mathrm{R_C}\}$) denote the number of hosts in state X. Let $\mathbb{1}\{x_i = X\}$ denote an indicator function that becomes 1 if the state x_i of host i is $X \in \mathcal{X}$; otherwise, 0 (e.g., $\mathbb{1}\{x_i = \mathrm{S_N}\} = 1$ if the state x_i of host i is $\mathrm{S_N}$). Note that $\mathbb{1}\{x_i = \mathrm{S_N}\} + \mathbb{1}\{x_i = \mathrm{S_C}\} + \mathbb{1}\{x_i = \mathrm{I_N}\} + \mathbb{1}\{x_i = \mathrm{I_C}\} + \mathbb{1}\{x_i = \mathrm{R_N}\} + \mathbb{1}\{x_i = \mathrm{R_C}\} = 1$ holds for each $i \in \mathcal{K}$.

We now consider the situation where the current system state is $\boldsymbol{\tau} = (x_1, \dots, x_i \dots, x_K)$. The system state transitions for the events (1)–(5) are represented as follows.

- When event (1) occurs for host i in S_k ($k = \mathrm{N}, \mathrm{C}$), the state x_i of the host transitions to I_k (①, ②). The total transition rate $\lambda_{\boldsymbol{\tau}}^{[k]}$ ($k = \mathrm{N}, \mathrm{C}$) for events that any host in S_k gets infected with the botnet malware by contact with an infected host is given by

$$\lambda_{\boldsymbol{\tau}}^{[k]} = \alpha \sum_{i \in \mathcal{K}} \Bigg\{ \mathbb{1}\{x_i = \mathrm{S}_k\} \cdot \sum_{j \in \mathcal{A}_i} \left(\mathbb{1}\{x_j = \mathrm{I_N}\} + \mathbb{1}\{x_j = \mathrm{I_C}\} \right) \Bigg\}. \tag{2}$$

- When event (2) occurs for host i in I_k ($k = \mathrm{N}, \mathrm{C}$), the state x_i of the host transitions to R_k (③, ④). The total transition rate $\mu_{\boldsymbol{\tau}}^{[k]}$ ($k = \mathrm{N}, \mathrm{C}$) for events that any host in I_k eliminates its botnet malware is given by

$$\mu_{\boldsymbol{\tau}}^{[k]} = \beta h(\mathrm{I}_k). \tag{3}$$

- When event (3) occurs for host i in S_k ($k = \mathrm{N}, \mathrm{C}$), the state x_i of the host transitions to R_k (⑤, ⑥). The total transition rate $\sigma_{\boldsymbol{\tau}}^{[k]}$ ($k = \mathrm{N}, \mathrm{C}$) for events that any host in S_k repairs its vulnerabilities is given by

$$\sigma_{\boldsymbol{\tau}}^{[k]} = \delta h(\mathrm{S}_k). \tag{4}$$

- For event (4), we assume that the vulnerability discovery capability of the botnet malware increases with the number of infected hosts. Thus, the discovery rate $\gamma_{\boldsymbol{\tau}}$ of a new vulnerability by the botnet malware is defined by

$$\gamma_{\boldsymbol{\tau}} = f(n) = \eta n, \tag{5}$$

where η denotes the discovery rate of a new vulnerability by each infected host and $n = h(\mathrm{I_N}) + h(\mathrm{I_C})$ represents the number of infected hosts [20]. In the countermeasure model proposed in this paper, the countermeasure group can discover unknown vulnerabilities the same as the botnet malware, and thus hosts can repair the discovered vulnerabilities before the botnet malware exploits them. We can regard this feature as the decay of the vulnerability discovery capability of the botnet malware. To represent this feature, we here consider two cases about new vulnerability discovery.

- The first case is that the botnet malware discovers a new vulnerability that has not been discovered by the countermeasure group. In this case, all hosts in $\mathrm{R_N}$ and $\mathrm{R_C}$ transition to $\mathrm{S_N}$ and $\mathrm{S_C}$, respectively (⑦, ⑧). We define the occurrence rate $\gamma_{\tau}^{[1]}$ of the first case as

$$\gamma_{\tau}^{[1]} = f^{[1]}(n,m) = \eta n\left(1 - \frac{m}{N}\right), \tag{6}$$

  where $n = h(\mathrm{I_N}) + h(\mathrm{I_C})$ represents the number of infected hosts and $m = h(\mathrm{S_C}) + h(\mathrm{I_C}) + h(\mathrm{R_C})$ denotes the number of countermeasure hosts. The occurrence rate increases with the number n of infected hosts. On the other hand, it decreases as the number m of countermeasure hosts increases.
- The second case is that the botnet malware discovers a vulnerability that has been already discovered by the countermeasure group. The occurrence rate $\gamma_{\tau}^{[2]}$ of the second case is defined as

$$\gamma_{\tau}^{[2]} = f^{[2]}(n,m) = \eta n\frac{m}{N}. \tag{7}$$

  Note that $\gamma_{\tau} = \gamma_{\tau}^{[1]} + \gamma_{\tau}^{[2]}$ holds. In this case, the transition events for hosts in $\mathrm{R_N}$ and $\mathrm{R_C}$ occur according to the scenarios, i.e., volunteer model and ally model scenarios discussed in Section 3.2.

  (I) *Ally model scenario*: In the ally model scenario, we assume that vulnerability information discovered by the countermeasure group is shared with only the countermeasure hosts, and hosts in $\mathrm{R_N}$ do not know the vulnerability information. Therefore, in the second case, all hosts in $\mathrm{R_C}$ transition to the state $\mathrm{S_N}$ (⑦). On the other hand, no transitions occur for hosts in $\mathrm{R_C}$.

  (2) *Volunteer model scenario*: In the volunteer model scenario, we assume that each vulnerability information discovered by the countermeasure group is shared with all the hosts. In this case, no transitions occur because each host has already repaired the discovered vulnerability.

- For event (5), an evolutionary game is done with the rate θ to determine whether a host joins/leaves the countermeasure group by comparing its payoff with the payoffs of the adjacent hosts, which will be discussed in Section 4. When host i in $\mathrm{S_C}$ (resp. $\mathrm{I_C}$ and $\mathrm{R_C}$) changes its strategy to leave the countermeasure group, the state x_i of the host transitions to $\mathrm{S_N}$ (resp. $\mathrm{I_N}$ and $\mathrm{R_N}$) (⑨). In contrast, when host i in $\mathrm{S_N}$ (resp. $\mathrm{I_N}$ and $\mathrm{R_N}$) changes its strategy to join the countermeasure group, the state transitions to $\mathrm{S_C}$ (resp. $\mathrm{I_C}$ and $\mathrm{R_C}$) (⑩).

When the system state is $\boldsymbol{\tau}$, the sojourn time of system state follows an exponential distribution with parameter $\lambda_{\tau}^{[\mathrm{N}]} + \lambda_{\tau}^{[\mathrm{C}]} + \mu_{\tau}^{[\mathrm{N}]} + \mu_{\tau}^{[\mathrm{C}]} + \sigma_{\tau}^{[\mathrm{N}]} + \sigma_{\tau}^{[\mathrm{C}]} + \gamma_{\tau} + \theta$ in the Markov chain. After the sojourn time, any one of the events (1)–(5) occurs with probabilities proportional to the transition rates.

4. Application of evolutionary games on networks to the epidemic model

We here discuss how to determine whether a host joins/leaves the countermeasure group in event (5) discussed in the previous section. In this paper, we suppose situations where hosts review their strategies (i.e., joining or leaving the countermeasure group) based on the evolutionary games on networks. In what follows, we describe the procedure of the strategy review of hosts based on the evolutionary games. Moreover, we explain the payoff definition.

4.1. Strategy review based on an evolutionary game

In event (5), each host compares its payoff with the payoffs of the adjacent hosts based on an evolutionary game on the network, and then reviews its strategy according to the payoffs. The detailed procedure is as follows, where we assume that each host knows the payoff and strategy of its adjacent hosts.

Step 1 Each host $i \in \mathcal{K}$ calculates its payoff $U_i(t)$ at time t when the event occurs, which will be discussed in Section 4.2.

Step 2 A candidate host is selected from among all the hosts in the network with the probability based on the payoff. The probability $P_i(t)$ that host $i \in \mathcal{K}$ is selected is defined as

$$P_i(t) = \frac{1 - U_i^{\mathrm{norm}}(t)}{\sum\limits_{j\in\mathcal{K}} \left(1 - U_j^{\mathrm{norm}}(t)\right)}, \tag{8}$$

where $U_i^{\mathrm{norm}}(t)$ is given by

$$U_i^{\mathrm{norm}}(t) = \frac{U_i(t)}{\max\limits_{j\in\mathcal{K}} U_j(t)}.$$

Step 3 The selected host i updates its strategy according to the Pairwise-Fermi rule [52]. Specifically, the host i chooses the host j^* having the largest payoff from among its adjacent hosts (i.e., $j^* = \arg\max_{j\in\mathcal{A}_i} U_j(t)$). Then, the host i adopts the same strategy as the host j^* with the probability $P_{i\leftarrow j^*}$, which is given by

$$P_{i\leftarrow j^*} = \frac{1}{1 + \exp\left[\dfrac{U_i(t) - U_{j^*}(t)}{\kappa}\right]}, \tag{9}$$

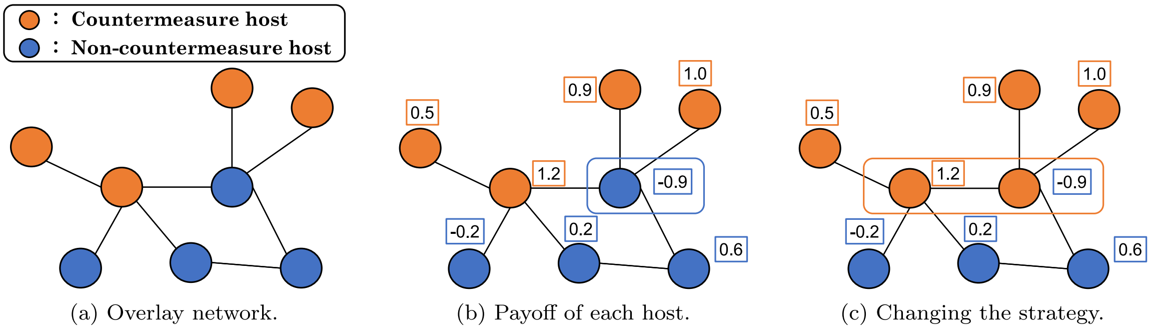
where κ is a parameter. Meanwhile, with the probability $1 - P_{i\leftarrow j^*}$, the host does not change its strategy.

Fig. 3 shows an example of the procedure. There exists an overlay network consisting of 8 hosts (Fig. 3(a)). In Step 1, each host calculates its payoff. Then, in Step 2, a host i is selected with the probability $P_i(t)$ defined in (8), which increases as the payoff decreases. We here assume that the host having the smallest payoff is selected (Fig. 3(b)). The host chooses the host j^* that has the largest payoff, and copies the strategy with the probability $P_{i\leftarrow j^*}$ given by (9) (Fig. 3(c)). This procedure is done every time event (5) occurs.

4.2. Payoff definition

We suppose that each host compares its strategy with the adjacent hosts to determine whether it changes the strategy. Thus, we define the payoff $U_i(t)$ of host $i \in \mathcal{K}$ at time t as

$$U_i(t) = \sum_{j\in\mathcal{A}_i} \left(B_i(t) - B_j(t)\right), \tag{10}$$



(a) Overlay network. (b) Payoff of each host. (c) Changing the strategy.

Fig. 3. Example of an evolutionary game.

Table 1
List of system parameters.

Parameters	Description
K	Total number of hosts
α	Malware infection rate
β	Malware elimination rate
δ	Vulnerability repair rate
η	Vulnerability discovery rate
θ	Game executing rate.
κ	Parameter of the Pairwise-Fermi rule
T	Parameter of the benefit
w_1, w_2, w_3	Weight parameters

where $B_i(t)$ denotes the benefit that host i has earned during time period $(t-T,t]$ and $T>0$ is a parameter. The benefit $B_i(t)$ is defined as

$$B_i(t) = \frac{1}{T}\left(w_1 L_i^{\mathrm{NI}}(t-T,t]\right. \\ \left. + w_2 L_i^{\mathrm{CM}}(t-T,t] - w_3 L_i^{\mathrm{CM}}(t-T,t]\right), \tag{11}$$

where w_1, w_2, and w_3 are weight parameters. $L_i^{\mathrm{NI}}(t-T,t]$ denotes the time length that host i has not been infected during time period $(t-T,t]$. Specifically, it is the total time that host i has belonged to $\mathrm{S_N}$, $\mathrm{S_C}$, $\mathrm{R_N}$, or $\mathrm{R_C}$ during time period $(t-T,t]$. $L_i^{\mathrm{CM}}(t-T,t]$ denotes the time length that host i has belonged to the countermeasure group (i.e., $\mathrm{S_C}$, $\mathrm{I_C}$, or $\mathrm{R_C}$) during time period $(t-T,t]$.

In (11), the assumption of the first term $w_1 L_i^{\mathrm{NI}}(t-T,t]$ is based on the idea of the FLIPIT game [41]. The FLIPIT game assumes the situation where an attacker and a defender compete for a target object. The benefit of the FLIPIT game is defined as the time period during which they keep the target object under their control. In our proposed epidemic model, we can regard the attacker and the defender as the botnet malware and each host, respectively. The target object is supposed to be the computing resource of the host. Thus, we give the first term $w_1 L_i^{\mathrm{NI}}(t-T,t]$ as the benefit of the host. If the host has not been infected for a long time, the value of the first term becomes large. This term can be considered as the emotional benefit of hosts.

The second term $w_2 L_i^{\mathrm{CM}}(t-T,t]$ indicates that the host gets some rewards according to the time length that the host joins the countermeasure group. For instance, an organizer gives countermeasure hosts some kickbacks in return for joining the countermeasure group. The third term $w_3 L_i^{\mathrm{CM}}(t-T,t]$ indicates the cost required for the host to participate in the countermeasure group. For example, the host needs to provide its computing resource to the countermeasure group. Therefore, the benefit of the host decreases with the increase in the value of this term.

5. Performance evaluation

5.1. Model

We conduct simulation experiments based on the continuous-time Markov chain to examine the dynamics of our proposed epidemic model on an overlay network. The list of system parameters is shown in Table 1. We use the Barabasi–Albert (BA) model [53] to construct the overlay network where there exist $K=500$ hosts and the average degree of the hosts is 4. The BA model has the small-world and scale-free properties, which are often seen in real networks. We can construct a network based on the BA model by sequentially adding $K-k_0$ hosts to an initial complete graph with k_0 hosts ($k_0<K$). The added hosts are connected to k existing hosts with probabilities proportional to the degree of the existing hosts. We assign consecutive numbers to hosts (i.e., $i=1,2,\ldots,k_0,\ldots,K$) in order of addition.

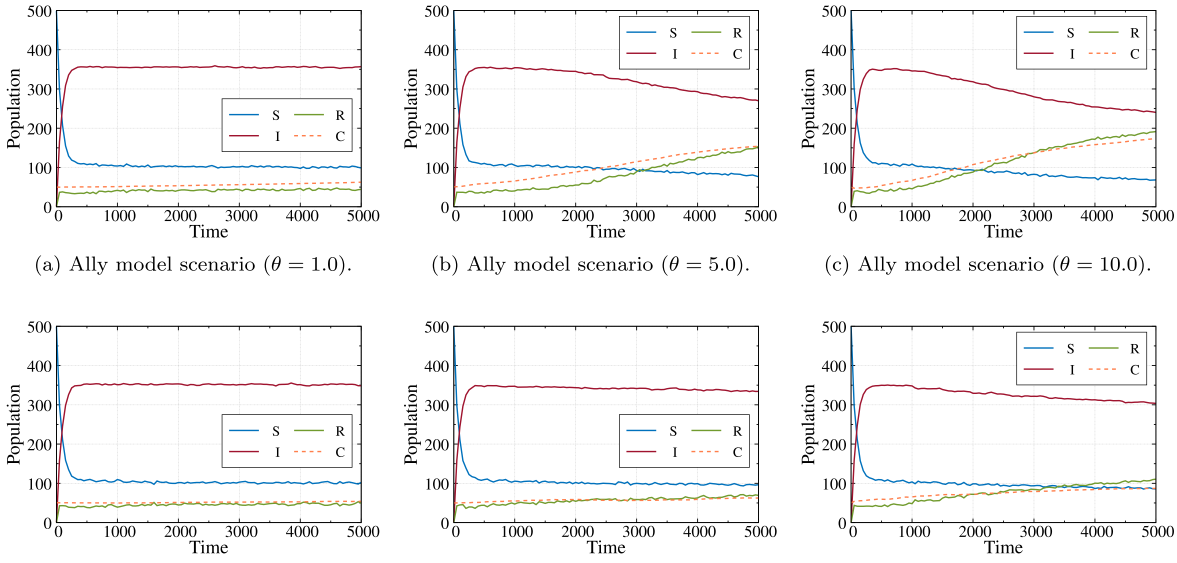
The parameters representing the infection rate α, the malware elimination rate β, the repair rate δ, the discovery rate η in (2)–(7) are set to 0.01, 0.006, 0.001, and 0.0002, respectively, based on the following reasons. In the simulation experiments, we assume situations where the botnet malware has very high infection capability in order to examine how effective the countermeasure group is against such malignant malware. Therefore, the infection rate α is set to a high value. On the other hand, we assume that each host does not actively protect itself from the malware to clearly show the impact of the countermeasure group. Thus, the malware elimination rate β and the repair rate δ are set to small values. In addition, it is difficult for each host to individually discover new vulnerabilities, compared with eliminating the malware or repairing known vulnerabilities. We assume that new vulnerabilities are effectively discovered by distributed machine learning with the use of the computing resources of many hosts. Therefore, the discovery rate η per each host is set to a very small value. The parameter κ in (9) is set to 0.1, which has been generally used in related works. The parameter T in (11) is set to 1.0. In the experiments, we examine the dynamics of our proposed epidemic model against the change in the parameters θ, w_1, w_2, and w_3 regarding the evolutionary game.

As the initial system state at time $t=0$, we assume that there exist one infected host and $K-1$ susceptible hosts. There are no recovered hosts in the network at time $t=0$. The initial infected host is host $i=1$ whose degree is 33 in the constructed network where the maximum degree is 70. In each experiment, each host $i\in\mathcal{K}$ belongs to the countermeasure group with the probability p_0 while it does not belong to the group with the probability $1-p_0$ at time $t=0$. In this paper, the probability p_0 is set to 0.1 to examine the effectiveness of the countermeasure group in situations where the number of hosts participating in the countermeasure group at initial state is small. We collect 300 independent samples from experiments and the average is shown in each result.

5.2. Results

5.2.1. Impact of emotional benefits

We first consider the scenario where there are no rewards and costs to join the countermeasure group. Specifically, the second and third terms in (11) are 0 (i.e., $w_2=w_3=0$). This case only considers the emotional benefit of hosts defined by the first term in (11). Fig. 4 shows the average number of hosts in each state as a function of the



(a) Ally model scenario ($\theta = 1.0$).

(b) Ally model scenario ($\theta = 5.0$).

(c) Ally model scenario ($\theta = 10.0$).

(d) Volunteer model scenario ($\theta = 1.0$).

(e) Volunteer model scenario ($\theta = 5.0$).

(f) Volunteer model scenario ($\theta = 10.0$).

Fig. 4. Average number of hosts in each state ($w_1 = 1.0$, $w_2 = w_3 = 0$).

elapsed time t in each scenario for different values of the executing rate θ of evolutionary games, where the weight parameter w_1 is 1.0. In each figure, "S" denotes the average number of susceptible hosts, "I" denotes the average number of infected hosts, "R" denotes the average number of recovered hosts, and "C" denotes the average number of countermeasure hosts.

As shown in Figs. 4(a) and (d), when the game executing rate θ is small, the countermeasure model does not work well in both ally and volunteer model scenarios. In the ally model scenario, the number of countermeasure hosts increases with time elapsed when θ is large, as we can see from Figs. 4(b) and (c). This is because hosts have a greater opportunity to review their strategies and tend to join the countermeasure group to obtain high emotional benefits defined in (11). Note that in the ally model scenario, hosts need to join the countermeasure group to obtain high emotional benefit because vulnerability information discovered by the group is not shared with non-countermeasure hosts. As a result, the ally model scenario can relatively suppress the malware spreading.

On the other hand, from Figs. 4(e) and (f), we observe that the volunteer model scenario cannot suppress the malware spreading even when θ is large. The reason is that in the volunteer model scenario, hosts can know vulnerability information discovered by the countermeasure group without joining it. In this case, non-countermeasure hosts can earn the same benefit as countermeasure hosts, so that non-countermeasure hosts are not actively encouraged to join the countermeasure group. Note that as we can see from Fig. 4(f), the number of countermeasure hosts gradually increases with the time elapsed to enhance the emotional benefit of each host. We thus conclude that the ally model scenario works more efficiently than the volunteer model scenario in the case where we consider only the emotional benefit defined by the first term in (11).

5.2.2. Impact of rewards and costs of the countermeasure group

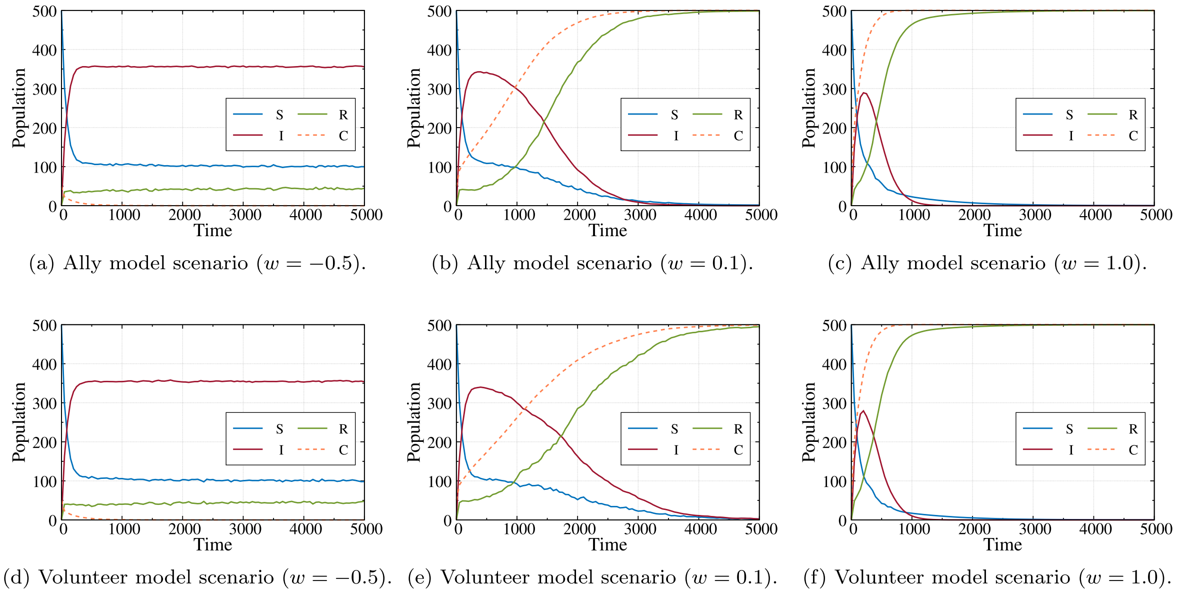
Although the ally model scenario can reduce the number of infected hosts as discussed above, its efficiency is not much high in the case where we consider only the emotional benefit. Thus, we then discuss the scenario where there are rewards and costs to join the countermeasure group, defined by the second and the third terms in (11).

Fig. 5 shows the average number of hosts in each state as a function of the elapsed time t in each scenario for different values of $w = w_2 - w_3$, where the weight parameter w_1 is 1.0 and the game executing rate θ is 5. Note that the second and third terms in (11) can be represented by $wL_i^{\mathrm{CM}}(t - T, t]$. When $w > 0$, rewards that hosts can earn outweigh their costs for joining the countermeasure group. On the other hand, when $w < 0$, costs that hosts are required are larger than their rewards for joining the countermeasure group. As we can see from Figs. 5(a) and (d), when $w = -0.5$, the number of countermeasure hosts does not increases in each model scenario. This is because the impact of the cost for joining the countermeasure group is greater than that of the emotional cost. This result implies that in order for hosts to join the countermeasure group, the value of w should be equal to or greater than 0.

On the other hand, as shown in Figs. 5(b) and (e) where $w = 0.1$, the number of countermeasure hosts markedly increases with time elapsed, compared with the results in Fig. 4. This result indicates that the reward for joining the countermeasure group efficiently encourages hosts to join the group. Because the value of w is low, the ally model scenario works more efficiently than the volunteer model scenario in this case, similar to results in the case where $w = 0$ shown in Fig. 4. From Figs. 5(c) and (f), we observe that when the value of w is large, the botnet malware is rapidly removed from the network by the countermeasure group in each scenario. We also observe that the behaviors of both scenarios are very similar, but the volunteer model scenario decreases the number of infected hosts slightly more efficiently than the ally model scenario. This result indicates that the reward for joining the countermeasure group, i.e., the second term in (11), becomes dominant for hosts to determine their strategies.

Next, we examine the transitions of the hosts in countermeasure group. Fig. 6 shows the average number of countermeasure and non-countermeasure hosts in each state as a function of the elapsed time t in each scenario for different values of $w = w_2 - w_3$, where the weight parameter w_1 is 1.0 and the game executing rate θ is 5. In each figure, "$\mathrm{S_N}$" (resp. "$\mathrm{I_N}$" and "$\mathrm{R_N}$") represents the average number of non-countermeasure hosts in the susceptible state (resp. infected and recovered state). Also, "$\mathrm{S_C}$" (resp. "$\mathrm{I_C}$" and "$\mathrm{R_C}$") represents the average number of countermeasure hosts in the susceptible state (resp. infected and recovered state).

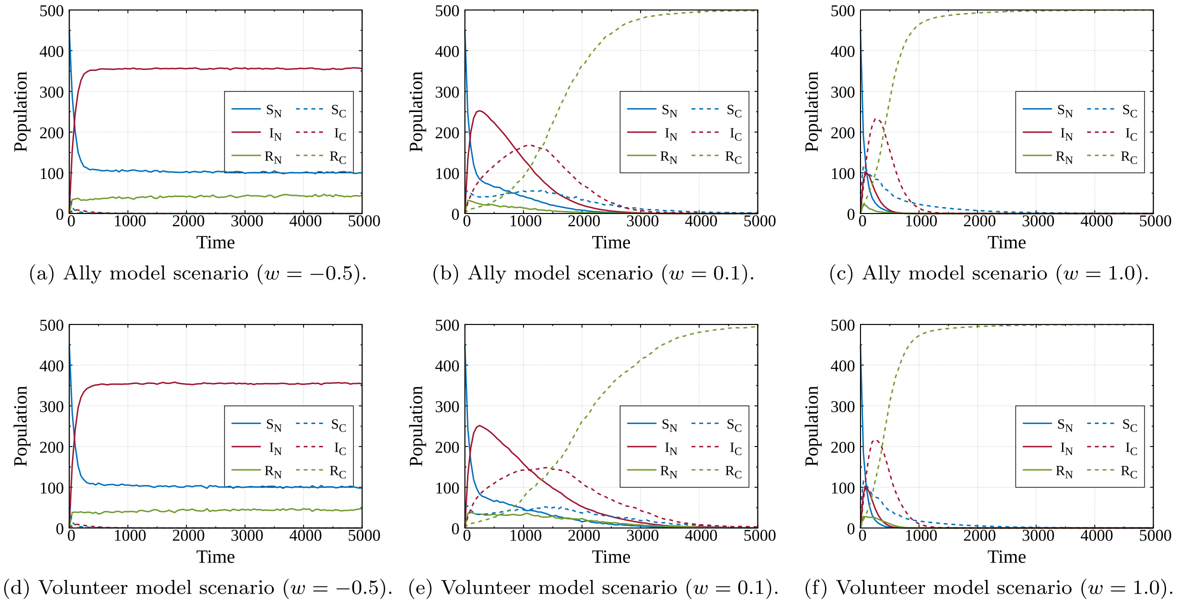
From Figs. 6(a) and (c) where $w = -0.5$, we observe that the countermeasure hosts disappear quickly due to the high cost for joining the countermeasure group. Therefore, in this case, the malware spreading cannot be suppressed. On the other hand, when $w > 0$, we can eliminate the botnet malware from the network after a certain period of time because the number of countermeasure hosts in each



(a) Ally model scenario ($w = -0.5$). (b) Ally model scenario ($w = 0.1$). (c) Ally model scenario ($w = 1.0$).

(d) Volunteer model scenario ($w = -0.5$). (e) Volunteer model scenario ($w = 0.1$). (f) Volunteer model scenario ($w = 1.0$).

Fig. 5. Average number of hosts in each state ($w_1 = 1.0$, $\theta = 5$).



(a) Ally model scenario ($w = -0.5$). (b) Ally model scenario ($w = 0.1$). (c) Ally model scenario ($w = 1.0$).

(d) Volunteer model scenario ($w = -0.5$). (e) Volunteer model scenario ($w = 0.1$). (f) Volunteer model scenario ($w = 1.0$).

Fig. 6. Average number of countermeasure hosts ($w_1 = 1.0$, $\theta = 5$).

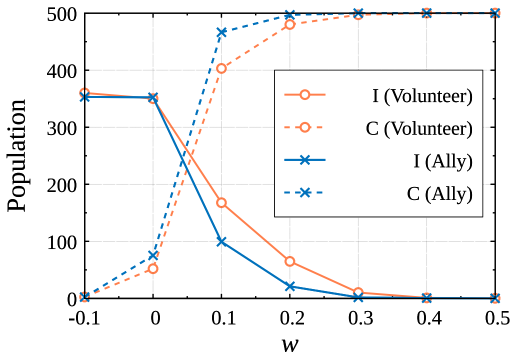
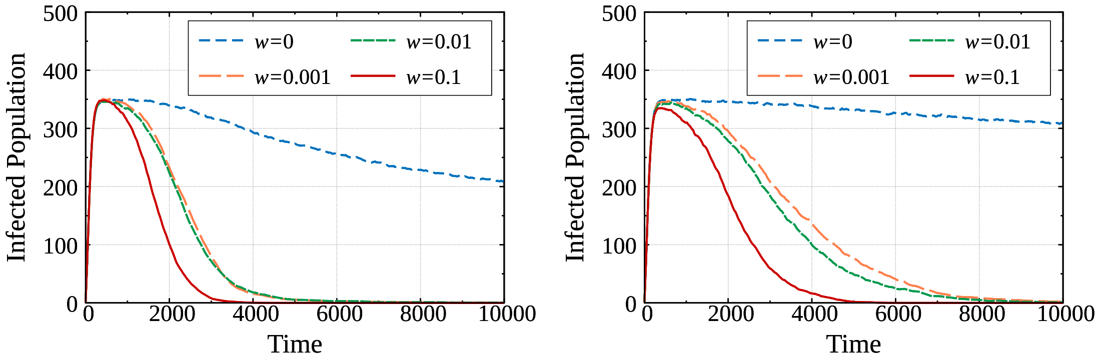


Fig. 7. Impact of the parameter $w = w_2 - w_3$.

state increases. As we can see from Figs. 6(b) and (e) where $w = 0.1$, the number of countermeasure hosts in the infected state firstly increases. The reason is that the first term with w_1 in (11) mainly affects the strategy decision of infected hosts. Then, the number of countermeasure hosts in the recovered state gradually increases with time elapsed due to the reward for joining the countermeasure group. In the ally model scenario, as discussed earlier, hosts easily tend to join the countermeasure group due to the emotional benefit, compared with the volunteer model scenario. Thus, when w (> 0) is small, the ally model scenario increases the countermeasure hosts more rapidly than the volunteer model scenario, as shown in Figs. 6(b) and (e). When w becomes large, the impact of the emotional benefit is quite low. Thus, the countermeasure hosts in the recovered state quickly increases, as shown in Figs. 6(c) and (f). As a result, the behaviors of both ally and volunteer model scenarios become similar.

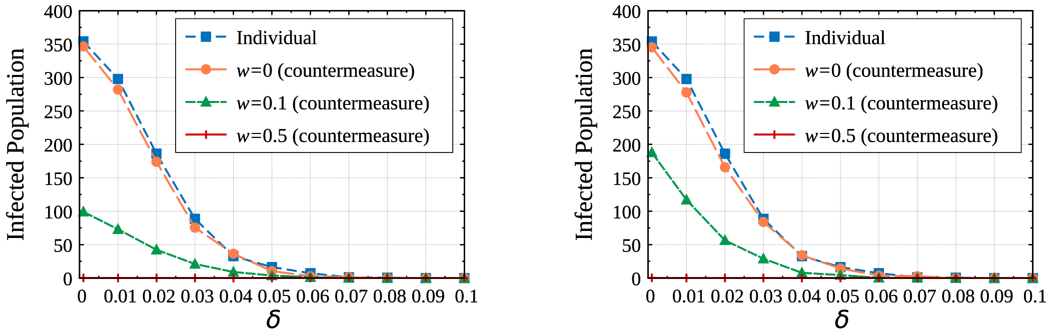
Fig. 7 shows the average number of infected hosts and countermeasure hosts in each scenario at elapsed time $t = 2{,}000$ as a function



(a) Ally model scenario.

(b) Volunteer model scenario.

Fig. 8. Average number of infected hosts.



(a) Ally model scenario.

(b) Volunteer model scenario.

Fig. 9. Average number of infected hosts against δ.

of parameter w, where $w_1 = 1.0$ and $\theta = 5$. As we can see from this figure, for $0 \leq w \leq 0.4$, the ally model scenario is more effective than the volunteer model scenario due to the emotional benefit. For $w < 0$, both scenarios do not work well because hosts do not join the countermeasure group. On the other hand, for $w \geq 0.5$, both ally and volunteer model scenarios efficiently eliminate the botnet malware from the network owing to the reward for joining the countermeasure group. We thus conclude that to effectively suppress the malware spreading with the use of the countermeasure group, we are required some rewards for joining the countermeasure group in addition to the emotional benefit.

Fig. 8 shows the average number of infected hosts as a function of elapsed time t, where the executing rate θ of evolutionary games is 5.0. As we can see from these figures, in both scenarios, giving even a few rewards can suppress the malware spreading. When w is very small, it takes time but the number of infected hosts can become close to 0. In addition, the elimination speed of infected hosts increases as the reward for joining countermeasure group increases. We also observe that the maximum number of infected hosts for each value of w is almost the same. This is because the number of countermeasure hosts increases after the number of infected hosts reaches a certain level, as shown in Fig. 5.

5.2.3. Effectiveness of the countermeasure group

Finally, we compare the performance of our countermeasure model with that in the epidemic model where each user host tries to individually counter the malware infection without the countermeasure group. In the epidemic model without the countermeasure group, to counter the malware infection, each user host needs to frequently repair discovered vulnerabilities individually. This situation can be represented by setting the vulnerability repair rate δ of each host to a high value. Thus, we can indirectly regard the repair rate δ as the cost for repairing vulnerabilities.

Fig. 9 shows the average number of infected hosts at elapsed time $t = 2,000$ against the repair rate δ. When the parameter w is more than 0, the countermeasure group works well. Thus, the number of infected hosts drastically decreases as shown in the figure even though the repair rate δ is very small. On the other hand, when the countermeasure group does not exist (labeled with "Individual"), the repair rate δ should be set to a very high value to eliminate the infected hosts. For example, in our countermeasure model, when $w = 0.5$, the number of infected hosts is almost 0 even for $\delta = 0.001$. Meanwhile, when each host individually counters the malware infection, δ should be over 0.07 to decrease the number of infected hosts to the same level as our countermeasure model. This result means that joining the countermeasure group dramatically reduces the cost for repairing vulnerabilities required for suppressing the malware infection.

From Figs. 7–9, we observe that even when w is a very small positive value, the number of infected hosts can be efficiently reduced, but the time to completely eliminate infected hosts increases as the value of w decreases. We also observe that when we adopt the countermeasure model, the repair rate δ can be set to a small value for suppressing the malware infection. Therefore, to appropriately determine the value of w, we should consider the required value of the repair rate δ (i.e., the cost for repairing vulnerabilities) and the time to eliminate infected hosts.

6. Discussion

We here discuss the extensibility and feasibility of our countermeasure model from a variety of perspective.

6.1. Extensibility

In this paper, we provide a basic epidemic model to represent the countermeasure model with evolutionary games. We can extend our

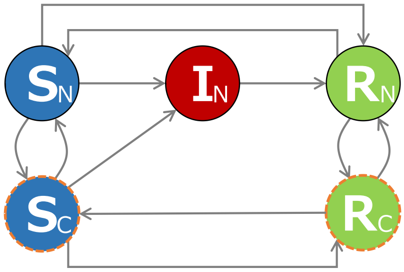


Fig. 10. Example of extension.

proposed model to some related models assuming different scenarios. In the following, we discuss the extensibility of our proposed model with some examples.

We first consider a scenario where some hosts cannot join the countermeasure group due to hardware/OS limitations. We can model this scenario by excluding the hosts from candidates of the evolutionary games. Specifically, the evolutionary games in the proposed model assume that each host $i \in \mathcal{K}$ can change its strategy. Instead of this assumption, we can consider the above scenario by applying the evolutionary games (i.e., the procedure in Section 4.1) to only hosts in the set $\mathcal{K} \setminus \mathcal{K}_L$, where $\mathcal{K}_L$ denotes the set of hosts that cannot join the countermeasure group.

Similarly, we can consider a scenario where countermeasure hosts are forced to leave the countermeasure group by getting infection with malware. This case can be modeled by the state transitions shown in Fig. 10. In this model, there is not state $\mathrm{I_C}$ that represents infected countermeasure hosts. When a susceptible host gets infected, it transitions to the state $\mathrm{I_N}$, regardless of whether it joins the countermeasure group. Accordingly, the transition rate given by (2) is appropriately changed to

$$\lambda_{\tau}^{[k]} = \alpha \sum_{i\in\mathcal{K}} \left\{ \mathbb{1}\{x_i = \mathrm{S}_k\} \cdot \sum_{j\in\mathcal{A}_i} \left(\mathbb{1}\{x_j = \mathrm{I_N}\} \right) \right\}. \tag{12}$$

By doing so, we can extend our proposed model to this scenario.

We also consider a scenario where it is difficult to find new vulnerabilities due to some prevention measures. This can be modeled by appropriately setting the system parameters of the proposed model. For example, by setting the discovery rate η to a very small value, we can represent situations where it is very difficult to find vulnerabilities due to the prevention measures. Similarly, by appropriately setting the system parameters, we can consider various scenarios, e.g., where malware has very high infectability.

6.2. Feasibility

We here discuss the feasibility of our proposed model from the viewpoints of vulnerability mining based on distributed computing, rewards for joining the countermeasure group, and motivation of users.

Our proposed model assumes that vulnerabilities are discovered by vulnerability mining techniques using distributed machine learning. Recently, some vulnerability mining techniques based on machine learning have been introduced [6–10]. For example, the methods in [6, 7] use a static code analysis, where sentences in source codes of software are segmented and used as input data to machine learning engines. Based on machine learning methods, they have shown the applicability of machine learning techniques to the discovery of unknown vulnerabilities in software. Furthermore, it is known that distributed computing with many inexpensive hosts can efficiently improve the performance of machine learning techniques [17,18]. It is then naturally expected that high-performance vulnerability mining methods with a large-scale distributed computing will be developed.

Based on these facts, we assume that user hosts discover vulnerabilities of commonly used software program whose source codes are publicly available with the use of vulnerability mining techniques, using the computing resources of users' computers in the form of a public distributed computing system such as Folding@home [54]. Folding@home simulates protein dynamics to help scientists develop new therapeutics for a variety of diseases. Volunteer users can join the system through a web browser-based application. We assume that our proposed countermeasure model works on such a distributed computing system where user hosts provide their computing resources to perform vulnerability mining.

In our proposed countermeasure model, some rewards are provided to user hosts to encourage them to join the countermeasure group as discussed in the simulation experiments. We assume that the rewards are provided by governments, cyber security companies, or some funds such as cloud funding. These providers could earn profits to cooperate in this system. For example, cyber security companies could invent a new market based on accomplishments by the countermeasure group. We believe that if the effectiveness of the countermeasure group for suppressing future serious malware infection is large enough, it is worth providing rewards to the countermeasure group.

As another idea about rewards, we can consider scores representing contributions of users. For example, Folding@home quantitatively assesses user computing contributions to the project, and adds some points to the score of users according to the contributions. This mechanism can foster friendly competition between users. Users compete with each other to get high scores, which can be the motivation for users to join the system. We thus can regard such scores as rewards for joining the countermeasure group.

We believe that our proposed countermeasure model is required for counter serious malware never before. Actually, a project in Folding@home, which is a distributed computing system has analyzed COVID-19 with the computing resources of a huge number of user computers. In this case, users have very actively joined the project to protect themselves from the threat of COVID-19. Similar to this case, it is expected that users actively join the countermeasure group to counter future serious malware because distributed computing systems including our countermeasure model could drastically enhance the computing performance.

7. Conclusion

This paper introduced a game-theoretic approach to epidemic modeling for discussing how to counter future malware evolution. In this paper, we consider a countermeasure model that constructs a countermeasure group discovering vulnerabilities earlier than malware or malicious attackers, and repairing them to protect hosts not to get infected with the malware. This paper provided stochastic epidemic modeling for the countermeasure model, which represents the infection dynamics of future malware based on a continuous-time Markov chain under countermeasure environments. Furthermore, in order to represent the strategy decision of hosts to join the countermeasure group, we applied evolutionary games on complex networks to the epidemic model. Through simulation experiments, we revealed countermeasure model scenarios to efficiently counter the future malware evolution. We concluded that some rewards for joining the countermeasure group are required in addition to the emotional benefit in order to effectively suppress the malware spreading with the use of the countermeasure group.

In this paper, we introduced a basic epidemic model to represent the countermeasure model with evolutionary games. As discussed in Section 6, our countermeasure model can be applied to some different scenarios by appropriately extending the model according to the scenarios. However, there exist scenarios that are difficult to apply our countermeasure model due to some technical reasons. For example, for a cooperate network where hosts need permission from an organizer to join the countermeasure group, game theory is not suitable for representing the behavior of the hosts. We need to modify the mechanism of the countermeasure group to tackle these problems, which we leave as future work.

Furthermore, we need to consider security problems. For example, we can consider a scenario where the countermeasure group comes under the control of a malicious attacker. In this scenario, the countermeasure group does not work well, and could lead to more serious attacks. As future work, we will tackle such security problems. Although security problems remain, we believe that hosts actively cooperate in the countermeasure group to counter future serious malware because our countermeasure model could drastically reduce the number of infected hosts and the cost for repairing vulnerabilities required for suppressing the malware infection.

CRediT authorship contribution statement

Hideyoshi Miura: Data curation, Software, Visualization, Writing – original draft, Writing – reviewing & editing. **Tomotaka Kimura:** Investigation, Conceptualization, Funding acquisition. **Hirohisa Aman:** Supervision, Methodology, Conceptualization. **Kouji Hirata:** Formal analysis, Validation, Writing – original draft, Writing – reviewing & editing, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Tomotaka Kumura, Hirohisa Aman, and Kouji Hirata reports financial support was provided by Japan Society for the Promotion of Science.

Data availability

Data will be made available on request.

Acknowledgments

This research was partially supported by JSPS, Japan KAKENHI Grant No. 20H04184.

References

[1] J. Jang-Jaccard, S. Nepal, A survey of emerging threats in cybersecurity, J. Comput. System Sci. 80 (5) (2014) 973–993.

[2] A.M. del Rey, Mathematical modeling of the propagation of malware: a review, Secur. Commun. Netw. 8 (15) (2015) 2561–2579.

[3] S.S. Silva, R.M. Silva, R.C. Pinto, R.M. Salles, Botnets: A survey, Comput. Netw. 57 (2) (2013) 378–403.

[4] P. Wainwright, H. Kettani, An analysis of botnet models, in: Proceedings of the 2019 3rd International Conference on Compute and Data Analysis, 2019, pp. 116–121.

[5] J.M. jules, H. Cheng, G.R. Regedzai, A survey on botnet attacks, Am. Acad. Sci. Res. J. Eng. Technol. Sci. 77 (1) (2021) 76–89.

[6] F. Yamaguchi, F. Lindner, K. Rieck, Vulnerability extrapolation: Assisted discovery of vulnerabilities using machine learning, in: Proceedings of the 5th USENIX Conference on Offensive Technologies, 2011, p. 13.

[7] R. Scandariato, J. Walden, A. Hovsepyan, W. Joosen, Predicting vulnerable software components via text mining, IEEE Trans. Softw. Eng. 40 (10) (2014) 993–1006.

[8] L. Luo, Q. Zeng, C. Cao, K. Chen, J. Liu, L. Liu, N. Gao, M. Yang, X. Xing, P. Liu, Tainting-assisted and context-migrated symbolic execution of android framework for vulnerability discovery and exploit generation, IEEE Trans. Mob. Comput. 19 (12) (2020) 2946–2964.

[9] G. Lin, J. Zhang, W. Luo, L. Pan, O. De Vel, P. Montague, Y. Xiang, Software vulnerability discovery via learning multi-domain knowledge bases, IEEE Trans. Dependable Secure Comput. 18 (5) (2021) 2469–2485.

[10] G. Lin, W. Xiao, L.Y. Zhang, S. Gao, Y. Tai, J. Zhang, Deep neural-based vulnerability discovery demystified: data, model and performance, Neural Comput. Appl. 33 (2021) 13287–13300.

[11] J. Borello, L. Me, Code obfuscation techniques for metamorphic viruses, J. Comput. Virol. 4 (4) (2008) 211–220.

[12] S.k. Sasidharan, C. Thomas, A survey on metamorphic malware detection based on hidden Markov model, in: 2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI, 2018, pp. 357–362.

[13] Z. Mumtaz, M. Afzal, W. Iqbal, W. Aman, N. Iltaf, Enhanced metamorphic techniques-a case study against havex malware, IEEE Access 9 (2021) 112069–112080.

[14] S. Noreen, S. Murtaza, M.Z. Shafiq, M. Farooq, Evolvable malware, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, 2009, pp. 1569–1576.

[15] A. Cani, M. Gaudesi, E. Sanchez, G. Squillero, A. Tonda, Towards automated malware creation: Code generation and code integration, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, 2014, pp. 157–160.

[16] B. Jin, J. Choi, H. Kim, J.B. Hong, FUMVar: A practical framework for generating fully-working and unseen malware variants, in: Proceedings of the 36th Annual ACM Symposium on Applied Computing, 2021, pp. 1656–1663.

[17] J. Dean, G.S. Corrado, R. Monga, K. Chen, M. Devin, Q.V. Le, M.Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, A.Y. Ng, Large scale distributed deep networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, 2012, pp. 1223–1231.

[18] K. Niwa, N. Harada, G. Zhang, W.B. Kleijn, Edge-consensus learning: Deep learning on P2P networks with nonhomogeneous data, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 668–678.

[19] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, J.S. Rellermeyer, A survey on distributed machine learning, ACM Comput. Surv. 53 (2) (2020) 1–33.

[20] T. Kudo, T. Kimura, Y. Inoue, H. Aman, K. Hirata, Stochastic modeling of self-evolving botnets with vulnerability discovery, Comput. Commun. 124 (2018) 101–110.

[21] K. Hirata, K. Hongyo, T. Kudo, Y. Inoue, T. Kimura, Consideration of a countermeasure model against self-evolving botnets, in: The 11th Intrenational Conference on Evolving Internet, 2019.

[22] K. Shimizu, Y. Kumai, K. Motonaka, T. Kimura, K. Hirata, Evaluation of countermeasure against future malware evolution with deterministic modeling, in: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2019, pp. 17–21.

[23] B. Allen, M. Nowak, Games on graphs, EMS Surv. Math. Sci. 1 (1) (2014) 113–151.

[24] W.O. Kermack, A.G. McKendrick, A contribution to the mathematical theory of epidemics, R. Soc. London Ser. A 115 (772) (1927) 700–721.

[25] X. Xiao, P. Fu, C. Dou, Q. Li, G. Hu, S. Xia, Design and analysis of SEIQR worm propagation model in mobile internet, Commun. Nonlinear Sci. Numer. Simul. 43 (2017) 341–350.

[26] H. Okamura, H. Kobayashi, T. Dohi, Markovian modeling and analysis of internet worm propagation, in: 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05), 2005.

[27] V. Karyotis, Markov random fields for malware propagation: The case of chain networks, IEEE Commun. Lett. 14 (9) (2010) 875–877.

[28] L. Yang, M. Draief, X. Yang, Heterogeneous virus propagation in networks: a theoretical study, Math. Methods Appl. Sci. 40 (5) (2017) 1396–1413.

[29] B. Qu, H. Wang, SIS epidemic spreading with heterogeneous infection rates, IEEE Trans. Netw. Sci. Eng. 4 (3) (2017) 177–186.

[30] S. Peng, M. Wu, G. Wang, S. Yu, Propagation model of smartphone worms based on semi-Markov process and social relationship graph, Comput. Secur. 44 (2014) 92–103.

[31] S. Shen, H. Zhou, S. Feng, J. Liu, H. Zhang, Q. Cao, An epidemiology-based model for disclosing dynamics of malware propagation in heterogeneous and mobile WSNs, IEEE Access 8 (2020) 43876–43887.

[32] G. Gonzalez, M.E. Larraga, L. Alvarez-Icaza, J. Gomez, Bluetooth worm propagation in smartphones: Modeling and analyzing spatio-temporal dynamics, IEEE Access 9 (2021) 75265–75282.

[33] J. Ren, Y. Xu, A compartmental model to explore the interplay between virus epidemics and honeynet potency, Appl. Math. Model. 59 (2018) 86–99.

[34] J. Ren, C. Zhang, Q. Hao, A theoretical method to evaluate honeynet potency, Future Gener. Comput. Syst. 116 (2021) 76–85.

[35] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[36] T.C. Truong, I. Zelinka, A survey on artificial intelligence in malware as next-generation threats, MENDEL 25 (2) (2019) 27–34.

[37] T.C. Truong, Q.B. Diep, I. Zelinka, Artificial intelligence in the cyber domain: Offense and defense, Symmetry 12 (3) (2020).

[38] Y. Shi, K. Davaslioglu, Y.E. Sagduyu, Generative adversarial network in the air: Deep adversarial learning for wireless signal spoofing, IEEE Trans. Cogn. Commun. Netw. 7 (1) (2021) 294–303.

[39] Y. Li, Y. Wang, Y. Wang, L. Ke, Y. an Tan, A feature-vector generative adversarial network for evading PDF malware classifiers, Inform. Sci. 523 (2020) 38–48.

[40] H. Li, S. Zhou, W. Yuan, J. Li, H. Leung, Adversarial-example attacks toward android malware detection system, IEEE Syst. J. 14 (1) (2020) 653–656.

[41] M. Dijk, A. Juels, A. Oprea, R. Rivest, FLIPIT: the game of ''stealthy takeover'', J. Cryptol. 26 (4) (2013) 655–713.

[42] L. Oakley, A. Opera, QFlip: An adaptive reinforcement learning strategy for the flipit security game, in: International Conference on Decision and Game Theory for Security, 2019, pp. 364–384.

[43] L. Greige, P. Chin, Deep reinforcement learning for FLIPIT security game, in: International Conference on Complex Networks and their Applications, 2020.

[44] T. Spyridopoulos, K. Maraslis, A. Mylonas, T. Tryfonas, G. Oikonomou, A game theoretical method for cost-benefit analysis of malware dissemination prevention, Inf. Secur. J.: Glob. Perspect. 24 (4–6) (2015) 164–176.

[45] T. Antal, S. Redner, V. Sood, Evolutionary dynamics on degree-heterogeneous graphs, Phys. Rev. Lett. 96 (18) (2006) 188104.

[46] C. Eksin, Control of stochastic disease network games via influential individuals, in: 2019 IEEE 58th Conference on Decision and Control, CDC, 2019 pp. 6893–6898.

[47] K.A. Kabir, J. Tanimoto, Dynamical behaviors for vaccination can suppress infectious disease - A game theoretical approach, Chaos Solitons Fractals 123 (2019) 229–239.

[48] D. Madeo, C. Mocenni, Evolutionary game theoretic insights on the SIRS model of the COVID-19 pandemic, IFAC-PapersOnLine 54 (17) (2021) 1–6.

[49] M.A. Amaral, M.M. de Oliveira, M.A. Javarone, An epidemiological model with voluntary quarantine strategies governed by evolutionary game dynamics, Chaos Solitons Fractals 143 (2021) 110616.

[50] D. Madeo, C. Mocenni, Identification and control of game-based epidemic models, Games 13 (1) (2022).

[51] Y. Nishihata, Z. Liu, T. Nishi, Epidemiological model of COVID-19 based on evolutionary game theory: Considering the viral mutations, in: 2022 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM, 2022, pp. 686–690.

[52] G. Szabo, G. Fath, Evolutionary games on graphs, Phys. Rep. 446 (4) (2007) 97–216.

[53] A.-L. Barabasi, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.

[54] V. Curtis, Patterns of participation and motivation in folding@home: The contribution of hardware enthusiasts and overclockers, Citiz. Sci.: Theory Pract. (2018).