Exact Markov Chain of Random Propagation of Malware With Network-Level Mitigation

Rodrigo Matos Carnier, *Member, IEEE*, Yue Li, Yasutaka Fujimoto, *Senior Member, IEEE*, and Junji Shikata, *Member, IEEE*

***Abstract*—In the age of Internet of Things (IoT), exploitation of security vulnerabilities is increasing, including self-propagating IoT malware. As an answer, specific research on IoT malware is being developed. Many studies use Markov chain models of malware propagation to predict the behavior of epidemics qualitatively and quantitatively. However, most studies approximate random propagation as a simple multiplicative term and no exact derivation of the Markov chain for random propagation was done so far. Moreover, systems of malware mitigation operating at the network level are rare and the majority of proposals focus on local networks like wireless sensor networks. In this article, we present a simple derivation of the exact Markov chain for random propagation of malware. Our model assumes a binomial form, compatible with binomial distributions in stochastic studies. To validate this derivation we implemented a stochastic simulation for the simplest compartmental epidemic model, susceptible–infected–susceptible (SIS). Predictions of the proposed Markov chain match simulation results with less than 0.2% error, well within stochastic variability and much smaller than the error of literature models. To complement our model of propagation, we developed and derived the Markov chain of a new system of malware mitigation, based on grouping random devices with identified infections during malware cleaning. Our mitigation system works at the network level and counteracts the vulnerability of mass deployment of IoT devices with aggressive but calculated mass disconnection. The system is able to artificially reduce R_0 (the basic reproduction number) below 1 and prevent malware taking over the network—all without changing the rate of detection.**

***Index Terms*—Cyber security, Internet of Things (IoT), malware, Markov chain model.**

I. Introduction

THE Internet of Things (IoT) is a relatively recent trend in development and deployment of Internet-connected devices that exploded in numbers and is not stopping soon [1]. From surveillance cameras to industrial remote sensors, from smartphones to SmartHouses, low-cost connectivity to the Internet suddenly created countless hardware applications. These devices have relatively low computational power compared to PCs or laptops and sometimes have only basic Internet connectivity, but their low specifications are exactly what made them viable as a product.

Such growth of Internet-connected low-spec devices led to widespread security vulnerabilities and diversification of cyber attack surfaces. They are caused mostly by three factors [2], [3], [4], [5]: 1) reduced computational or networking resources in IoT devices, which prevent or hinder the use of local advanced security software or intense remote data processing; 2) use of security measures and software designed for standard networks, disregarding limitations of IoT hardware or inadequacy of IoT firmware; and 3) widespread neglect by IoT manufacturers to implement proper security measures and maintenance. These are chronic problems caused by the business model of the IoT technology: mass deployment for profit through low-cost quantity instead of aggregated value, discouraging manufacturers to increase costs by implementing strong security measures. This setup-and-forget scenario is perfect for cyber criminals, which are increasing their attacks on IoT networks, particularly by developing IoT malware.

Between the different types of IoT malware, botnets, and worms have drawn much attention for exploiting weak IoT securities in a self-replicating manner, creating massive botnets, and performing highly disruptive DDoS attacks [6]. The most successful IoT malware to date, Mirai, transformed 400 000 devices into bots and used them to take down hundreds of websites for hours, including Twitter, Netflix, Reddit, and GitHub [5], [7], [8]. Catching up with these challenges, studies on self-replicating IoT malware increased, focusing mostly on one of these three topics: 1) different detection methods to mitigate malware at the device or LAN level [9], [10]; 2) mitigation strategies for IoT devices at the LAN level [11], [12]; and 3) mathematical or simulation models of malware propagation in the form of differential equations/Markov chains [11], [12], [13], [14], [15], agent-based models [16], [17], cellular automata [18], stochastic models [19], [20], and others.

Regarding mathematical models, many studies use Markov chains as a framework due to its simplicity and flexibility of both deterministic and stochastic application. The generality of the tool is shown in its diverse application: from simple epidemic models of malware propagation, to its detection [21], to IoT localization in SmartHouses [22], to complementary

Manuscript received 6 September 2022; revised 22 December 2022; accepted 24 January 2023. Date of publication 30 January 2023; date of current version 7 June 2023. This work was supported by the Ministry of Internal Affairs and Communications, Japan, through the contract of "Research and Development on IoT Malware Removal/Make It Nonfunctional Technologies for Effective Use of the Radio Spectrum" among "Research and Development for Expansion of Radio Wave Resources under Grant JPJ000254." *(Corresponding author: Rodrigo Matos Carnier.)*

Rodrigo Matos Carnier, Yue Li, and Yasutaka Fujimoto are with the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama 240-8501, Japan (e-mail: rodrigo.carnier@gmail.com; li-yue-np@ynu.jp; fujimoto@ynu.ac.jp).

Junji Shikata is with the Graduate School of Environment and Information Sciences, Yokohama National University, Yokohama 240-8501, Japan.

Digital Object Identifier 10.1109/JIOT.2023.3240421

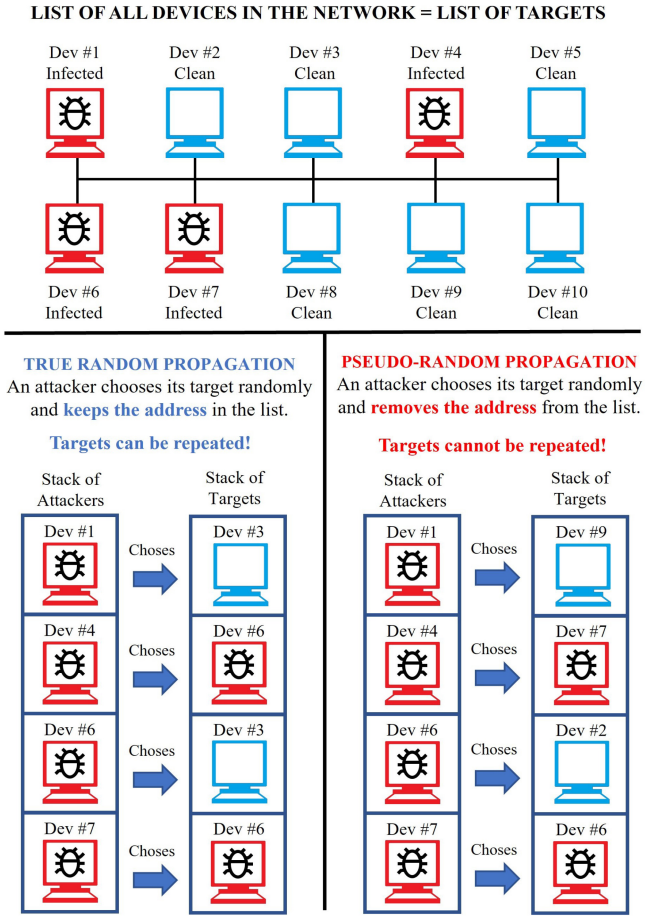

Fig. 1. Red computers: infected devices performing attacks on targets chosen randomly from the whole network. Blue computers: noninfected devices. Left side: targeting logic of our exact Markov chain model for random propagation. Right side: targeting logic of the most popular Markov chain model of literature, which is an approximation of random propagation.

frameworks of machine learning for traffic modeling [23]. The latter, for example, used Markov chains as a stochastic framework of ML-based traffic modeling to provide granular characterization of network traffic and evaluation of the machine-learning algorithm itself. In the context of malware modeling, a model possessing the core dynamics of propagation can predict many important metrics, both during transition gradients and in the equilibrium state. All reviewed Markov chain models focused their contribution in different epidemic states and parameters to properly represent the core dynamics. But given the simplification brought by many assumptions (like homogeneous networks), eliminating unnecessary inaccuracies is important. Since all papers proposing Markov chain models for malware use approximations of random propagation, we chose to focus on the exact modeling of random propagation itself. Fig. 1 summarizes the difference between the approximated and our exact random propagation model.

While reviewing the literature we also noticed a lack of studies of mitigation of malware at the network level. This is understandable, given the complexity of modeling IoT network topologies, such as ZigBee and 6LowPan, which address the need of flexibility in network configuration and the challenge of intermittent device connectivity (e.g., devices that move between connection points and devices with low battery life). While it is important to understand how malware propagate in such topologies and develop mitigation systems for them, there is a lack of studies on mitigation at the network level even for simpler scenarios: CCTV networks and SmartHouses, where the connectivity of nodes are more permanent. In this article, we are interested in these later scenarios, for which we propose a mitigation system with three characteristics: 1) malware mitigation at the network level; 2) exact mathematical models for both malware propagation and malware mitigation; and 3) a mitigation methodology based on centralized management of security and commands sent through the server. Fig. 4 summarizes of our proposed group mitigation strategy.

The remainder of this article is organized as follows. Section II lists the contributions of this article and technical points tackled by each contribution. Section III discusses the two mechanisms of propagation inefficiency in random propagation, describes our methodology of exact Markov chain derivation and presents the exact mathematical model of random propagation of malware. Section IV describes our proposed method of network-level mitigation of malware and derives its dynamics. Section V presents five different Markov chain models mixing different dynamics of propagation and mitigation, demonstrating how our proposed models can be mixed with others. Section VI shows the validation of our proposed Markov chain and compares its the performance with the two most popular literature models. Section VII evaluates the effectiveness of our proposed mitigation system, tuned by the parameter G that represents the intensity of mass mitigation. Section VIII presents final discussions, including our preliminary study on the sensitivity of our mitigation strategy to false negatives (FNs) and positives and how it relates to our design assumptions. Section IX summarizes this article and presents our conclusions.

II. Contributions

In this article, we present two contributions.

1) The *exact Markov chain* of random malware propagation.
2) A *system of group mitigation* that exploits the mass-deployment of IoT devices, and its Markov chain model.

To achieve them, we tackled the following technical points.

1) *Proper Modeling of Random Malware Propagation:* When malware spreads randomly, two dynamics of inneficient target selection occur and waste opportunities of infection (see Fig. 1 and the detailed explanation in Section III). These dynamics were never explained in details or explicitly before. We did so in this study and developed the first exact Markov chain model of random propagation as far as we know.
2) *New Mitigation System Specific for IoT Networks:* One of the big challenges of the IoT technology is the mass deployment of devices with low specs and insufficient security capabilities, which is readily exploited by malware that propagates through worm approaches. Our proposed system of mitigation exploits the same tradeoff

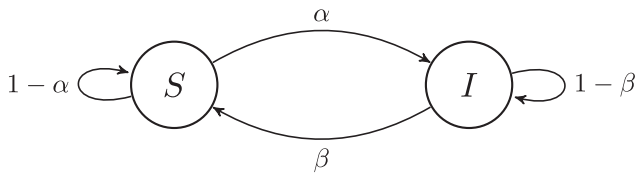

Fig. 2. Markov process of SIS model, the simplest compartmental model. *S* is the susceptible state and *I* is the infected state. The figure shows a linear model, where α (the infection rate) is the probability of transition of *S* to *I* and β (the detection rate) is the probability of transition of *I* to *S*.

between mass deployment and individual low impact of devices, grouping a designed number of random devices with every identified infected device and then performing the cleaning action. This strategy is derived from our previous work [24], which exploited the constraints of local propagation of malware to clean devices in local groups. In this work the propagation is random, therefore, we developed a grouping strategy that is also random but nonetheless repeats the gains of performance of the previous study.

III. Random Propagation of Malware

A. Basic Definitions and Symbols

Historically malware modeling has used the same compartmental models of biological diseases. These models define specific states for each phase of infection and describe the dynamics of disease as a cycle—from healthy, to infected, to either healthy again or deceased. Moreover, epidemic models are compartmental, which means there is no overlapping of states and the transitions are complete from one state to another. The state of individuals is not considered, instead all individuals in the same phase of infection are grouped in populations, whose size varies according to the malware epidemic's evolution.

In this study, we used the simplest compartmental model: the SIS model or susceptible–infected–susceptible. It represents a closed cycle where infected individuals can recover and the total population can be completely cleaned of malware. Fig. 2 presents a Markov process diagram of the SIS model. The network model is self-contained and has no exterior contact. It has a fixed total number of devices *N*. The two possible populations are the susceptible population *S* (non-infected) and the infected population *I*. The most important parameters are the infection rate α, which is the chance of a susceptible device being infected after an attack, and the detection rate β, which is the chance of an infected device being identified and cleaned by an anti-malware system. Since the network has no exterior contact, a study of the dynamics of malware propagation starts with a percentage of devices already infected. This initial infected population I_0 is what starts the propagation of malware and spread it to the rest of network. Table I summarizes all symbols used in this article.

All models studied in this article are discrete: time is normalized as turns (natural numbers) and changes incrementally from one turn $[T]$ to the next $[T+1]$. Equation (1) shows the relationships of *N*, *S*, and *I* with *T*

$$N = S[T] + I[T]. \tag{1}$$

The system states are $S[T]$ and $I[T]$. The system dynamics will be presented as a system of difference equations, exemplified in

$$\begin{cases} S[T+1] - S[T] = f_1(S[T], I[T]) & (2) \\ I[T+1] - I[T] = f_2(S[T], I[T]). & (3) \end{cases}$$

TABLE I
Definitions and Symbols

Independent Variable

T	Turn (discrete time)

States

N	Total number of devices
$S[T]$	Susceptible devices
$I[T]$	Infected devices

Special Terms and Quantities

$I_n[T]$	Successful infections in a turn
$M_n[T]$	Cleaned infections in a turn
$C[T]$	Cost of mitigation (total reset devices)
$I[T_f]$	Value of convergence of I
T_c	Time constant of I (95% of $I[T_f]$).
$F_n[T]$	Quantity of false negative detections
$F_p[T]$	Quantity of false positive detections

Parameters

α	Infection rate
β	Detection rate
R_0	Basic reproduction number
G	Mitigation multiplicator
FPR	False positive rate

B. Review of Literature Models

Random propagation happens when infected devices randomly attack other devices without coordination, spreading malware in a network. In a completely random propagation, the targets of attack are chosen randomly from the entire population *N*, and this choice is done independently for every attack. Modeling randomness in malware propagation has proven difficult, given the disparity between simple compartmental models and the complex computer networks they represent. This caused studies on malware to widely adopt approximated models or simplify the logic of randomness.

Below, we discuss the two common Markov chains for random propagation of malware: 1) the linear model and 2) the most popular nonlinear model in the literature (from now on we will call it the "standard nonlinear model.")

In the linear model [11], [12], random infection is modeled by simply taking the population of susceptible devices and multiplying it by the rate of infection

$$I_{n/\text{lin}}[T] = \alpha S[T] \tag{4}$$

where $I_{n/\text{lin}}[T]$ represents the population of newly infected devices for the linear model and α is the rate of infection. Note that regardless of the model, $I_n \neq \Delta I$, where ΔI is the total variation of I from one turn to the next. ΔI actually depends on both I_n and the subtraction of cleaned devices.

While this approximation is reasonable and easy to transform into a closed-form solution, it can incur in significant error (>10%) because it does not take into consideration the availability of infected devices to perform attacks.

In the standard nonlinear model [13], [14], [15], [16], [17], this mutual availability is taken into consideration by multiplying the previous linear term by the percentage of infected devices in respect to the total population N

$$I_{n/\text{std}}[T] = \alpha S[T]\frac{I[T]}{N}. \tag{5}$$

The logic behind this is to treat new cases of infection as the number of encounters of susceptible devices with infected ones [13]. If there are few infected devices in the network, there will not be many infections even if there is a big availability of susceptible devices and α is high. By considering the availability of both populations, when one becomes small the number of new infections also becomes small (which is what actually happens in real scenarios). However, this logic does not account for the possibility of infected devices attacking the same target (see Fig. 1). In true random propagation, this can happen and wastes an opportunity of infection if the previous attack already succeeded. As a consequence, *the standard nonlinear model overestimates the number of infections*. Take for example the optimal value for propagation, when there is balanced availability ($S = I = 0.5$ N) and guaranteed infection after attacks ($\alpha = 1$). There is a 50% chance of infected devices attacking susceptible devices, which multiplied by αS yields 25% of chance of new infections. But this chance only applies if infected devices are not allowed to attack the same targets. Else the chance of attack repetition must be calculated and subtracted from 25%—this calculation is what we will do below.

In this article, we propose to model the propagation of malware exactly. To do so, first we will consider every possible scenario of attack in our model, then derive the Markov chain for random propagation after every attack, extrapolating the results to N numbers of attack according to the evolution of the Markov chain.

C. Modeling Random Propagation

In random propagation of malware, malware attacks, and successful infections (which do not necessarily match in numbers) follow a random pattern. But since some attacks fail to infect, the number of random attacks is bigger or equal than the number of infections they generate. To clarify this point, we define an attack as *efficient* if it can change the state of the target to infected state, and *wasted* if it cannot do so (either because it already started the turn as infected or because it was infected this turn before the current attack). To understand all the possibilities of wasted and efficient attacks, we need to list the different circumstances (or scenarios) of attack in respect to the states of its players: the attacker and the target. While there is only one type of attacker (infected devices), there are four different types for targets, each one generating a different scenario of attack. These scenarios are described below in details and depicted in Fig. 3.

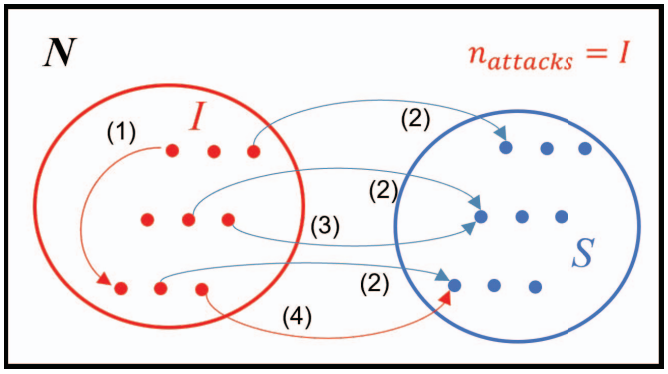

Fig. 3. Four types of attack in a scenario of random targeting. Blue arrows: efficient attacks. Red arrows: wasted attacks.

1) *Attacking another infected device*. This attack is wasted since the target is already infected.
2) *Attacking a susceptible device for the first time*. Every first attack is efficient because the target is always clean and can be infected.
3) Repeating an attack on a susceptible device after failure of all previous attempts. This attack is efficient because it still can change the target' state to infected.
4) Repeating an attack on a susceptible device after success of one of the previous attempts. This attack is wasted because the target' state is already compromised and will change in the next turn regardless of the current attack.

The quantity of scenarios presents a dilemma: there is only two states in the system, but an infected device can attack four types of targets. To represent them, we define additional states created from the division of the susceptible population into three virtual populations, as shown in (6). They correspond to the targets of scenarios (2)–(4) listed above

$$S[T] = S_1[T] + S_f[T] + S_i[T] \tag{6}$$

where S_1 are susceptible devices not attacked before, therefore, targets for 1st time attacks; S_f are susceptible devices that already suffered one or more attacks but they all *failed*; and S_i are susceptible devices that were attacked and already *infected* this turn by one successful attack (note: remember that these devices will change state to I only in the next turn).

If we calculate S_i after all attacks in a turn, we find the total number of new infections in the end of the turn, which yields

the term I_n of our proposed Markov chain

$$I_n[T] = {}^{+}S_i[T] \tag{7}$$

where ${}^{+}S_i[T]$ represents $S_i[T]$ after all attacks performed in a turn. All subsequent appearances of the superscript $^{+}$ will represent the same thing: the change in a variable after all attacks were performed in that turn.

After all attacks are performed, the newly infected devices $I_n[T]$ will be moved from $S[T]$ to $I[T]$ to compose $S[T+1]$ and $I[T+1]$. If there is no mitigation, $I[T+1]$ is given by

$$I[T+1] = I[T] + I_n[T]. \tag{8}$$

If there is mitigation, $I[T+1]$ is given by

$$I[T+1] = I[T] + I_n[T] - M_n[T] \tag{9}$$

where M_n is the group of devices removed from I by a mitigation strategy. We will derive M_n in Section IV.

Having established all possibilities of random attacks and how they are used to calculate I_n, we can now present the development of our exact Markov chain model.

D. Exact Markov Chain of Random Propagation

In a real-world scenario, attacks can happen at the same time, but to facilitate the derivation of our Markov chain we assumed an order of attack: the sequence of network addresses. To calculate *Si* we analyzed the evolution of all four populations of (6) during a single turn, calculating how each one changed after every attack. Since our simulated network has 10^6 devices, calculating the size of all four populations for all attacks is an immense effort. Therefore, we limited the calculations for the first three attacks and investigated the mathematical pattern of the evolving probability of each population.

From the evolving pattern in Populations Size after every attack, we observed that a new term was added after every attack. The coefficients of the expression representing population S_1 followed a famous pattern: the rows of extended Pascal's triangle, where alternate elements are negative. The absolute value of these coefficients is represented by the operator *n chooses p*, which is then corrected by the iterative operator $(-1)^i$ to turn the alternate elements into a negative value. Observing the pattern of addition of terms and its coefficients, as well as the relationship between the terms for S_1 and the other populations, we derived a generalized expression for the size of every population after an arbitrary number of attacks (I) are performed

$${}^{+}S_1[T] = N\sum_{i=0}^{I}(-1)^i\binom{I}{i}\frac{S}{N^{i+1}} \tag{10}$$

$${}^{+}S_f[T] = N\sum_{i=1}^{I}(-1)^i\binom{I}{i}I\left(1-\alpha^i\right)\frac{S}{N^{i+1}} \tag{11}$$

$${}^{+}S_i[T] = N\sum_{i=1}^{I}(-1)^i\binom{I}{i}I\alpha^i\frac{S}{N^{i+1}} \tag{12}$$

$${}^{+}I[T] = I[T]. \tag{13}$$

However, the computational cost of this summation is still too heavy. To derive a simpler formulation, we refer to the Binomial Theorem, which shows the equivalence between a binomial and its algebraic expansion into a series of terms with coefficients equal to the elements of a row of Pascal's Triangle. Therefore, (10)–(13) can be transformed into

$${}^{+}S_1[T] = S[T]\left(1-\frac{1}{N}\right)^{I[T]} \tag{14}$$

$${}^{+}S_f[T] = S[T]\left(\left(1-\frac{\alpha}{N}\right)^{I[T]} - \left(1-\frac{1}{N}\right)^{I[T]}\right) \tag{15}$$

$${}^{+}S_i[T] = S[T] - S[T]\left(1-\frac{\alpha}{N}\right)^{I[T]} \tag{16}$$

$${}^{+}I[T] = I[T]. \tag{17}$$

Substituting (16) in (7), we find the exact Markov chain for random malware propagation

$$I_n[T] = S[T] - S[T]\left(1-\frac{\alpha}{N}\right)^{I[T]}. \tag{18}$$

This binomial form of the exact derivation of random propagation was not found in any other Markov chain model for malware propagation in the literature. However, it is noteworthy that stochastic studies of malware epidemics produced binomial distributions for the system populations [19], [20], strengthening our conclusion that this model represents the exact dynamics of random propagation. Nonetheless, we performed an independent validation of our model, whose results are presented in Section VI.

IV. Mitigation System

Carnier et al. [24] proposed a strategy of group mitigation that exploits mass deployment of IoT devices as malware does. The scenario of simulation was an IoT network with a binary-tree topology where each leaf represented an IoT WAN. The infection of a leaf meant that an unknown number of devices inside the WAN was infected and, therefore, the entire WAN was compromised. To investigate the impact of local constraints on communication and how malware would spread in the intermediary levels of the Internet topology, every WAN could only infect its neighboring WANs. The proposed mitigation strategy was based on cleaning every WAN identified as infected and all its neighbors, regardless of actual detections in the neighbors. This "area" of mitigation could be expanded by increasing the number of hops from the central WAN with a detected infection. We parameterized this number of hops as a control variable of mitigation, naming it local routing depth (LRD) (see Fig. 4(a) for an example of local group mitigation based on LRD). In our simulations, we confirmed that applying group mitigation is a valid tradeoff strategy that increased the effectiveness of malware mitigation at a smaller collateral cost of downtime of noninfected units caught in the group mitigation. Nonetheless propagation of malware was eradicated faster and with smaller detection rates. If the IoT network is composed of nonessential devices that can be mass deactivated, such strategy counteracts the main vulnerability of IoT networks: big numbers of vulnerable devices that cannot afford proper individual protection.

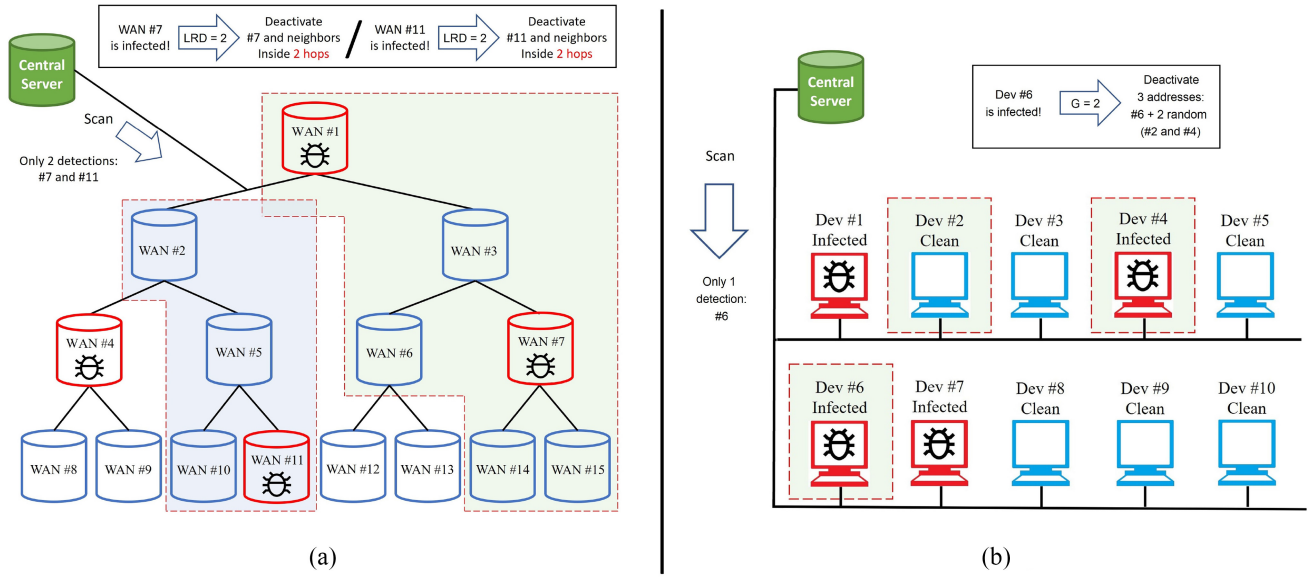

Fig. 4. (a) Group mitigation of earlier work [24]: in this study, random propagation of malware is local, between neighboring WANs, in a network with binary tree topology. The group mitigation strategy is based on the concept of LRD, which includes neighboring WANs in the cleaning action of each identified infection. (b) Group mitigation of present paper: in this study, random propagation of malware is global between any device connected to the network, which does not have specific topology (hub-like). The group mitigation strategy is based on the concept of Mitigation Multiplicator (G), which includes random devices in the cleaning action of each identified infection.

In this work, we propose a modified group mitigation strategy, designed to counteract random global propagation (in contrast to the previous work where we designed a strategy of local mitigation to counteract random local propagation). Since malware propagates now without local constraints in a random manner, the grouping of mitigation targets is also designed to be random. First, we perform a network-wide sweep of all devices, checking for infection with a rate of detection β. After the total number of detections βI is determined, we perform their cleaning plus the cleaning of $G\beta I$ other devices chosen randomly, regardless of their current state. This parameter G is a control variable that can be changed according to the desired escalation of mitigation. We will call it the *MitigationMultiplicator* (see Fig. 4(b) for an example of global group mitigation based on G).

An important difference between random propagation and our proposed mitigation system is that the former has two inefficient mechanisms, but our mitigation strategy has only one (random targeting of already clean devices). It does not need to implement the repetition of targets, which is unavoidable only in decentralized phenomena without coordination, like propagation of worm malware. In other words, the targets of mitigation are chosen *randomly* and *removed* from the list of targets. Therefore, the number of devices targeted by mitigation will be

$$C[T] = (1 + G)\beta I[T] \tag{19}$$

where C is the number of devices to undertake a cleaning procedure in turn T, G is the Mitigation Multiplicator, and β is the rate of detection. C represents the *cost* of mitigation, which will be explored in Section VII.

When $G = 0$, the mitigation will fall back to individual mitigation of inspected devices. When $G = 1$, for every detected infection, an additional devices will be a target of mitigation, and so on. Naturally, some of these additional devices will not be infected, incurring in an equivalent phenomenon of self targeting. Therefore, the final number of actually mitigated devices will be equal or less than $C[T]$. In the following section, we will derive the Markov chain for the exact number of infected devices actually cleaned.

TABLE II
DERIVATION OF MARKOV CHAIN FOR GROUP MITIGATION

Devices	Number of Targets	Probability of Being Infected
Detected	βI	100%
Additional	$G\beta I$	$\frac{I - \beta I}{N}$

A. Exact Markov Chain of Random Mitigation

While we can be sure of the infection in detected devices, some of the other G devices being "cleaned" are not actually infected. However, a proportion of G will have a stochastic chance of being infected, which is proportional to the number of infections in the network minus the number of detected devices. Therefore, the derivation of M_n (the actual number of infections being cleaned) requires $G\beta I$ to be separated from $(1 + G)(\beta I)$ and multiplied by its chance of being infected.

Then, multiplying the *Number of Targets* by the *Probability of Being Infected* for every device (see Table II), summing the results, and simplifying the equation, we find the exact Markov chain of Group Mitigation

$$M_n[T] = I[T]\left(\beta + GI[T]\beta\frac{(1 - \beta)}{N}\right). \tag{20}$$

A similar procedure was performed to derive the model of random propagation, but it was done attack-by-attack. For group mitigation, we just needed to make this evaluation once for the entire turn. This is due to the design of random targeting without repetition, which discards the necessity of micro analyzing every step of the mitigation sequence.

V. Exact Markov Chains

We used discrete-time Markov chains (DTMCs) to describe the dynamical equations representing the evolution of a malware epidemic. A DTMC considers only the previous state of the system to calculate the current state. Given all the steps of modeling and derivation presented in the previous sections, we will now present the complete Markov chain for the random malware model, joining together the Markov chains of random propagation and group mitigation.

The SIS model dynamics is composed of the two terms derived in the previous sections: 1) infection I_n and 2) mitigation M_n. After a turn, $S[T]$ is decreased by new infections $I_n[T]$ and increased by mitigation $M_n[T]$, and vice versa for $I[T]$. Thus, SIS dynamics assumes the general form of

$$\begin{cases} S[T+1] - S[T] = -I_n[T] + M_n[T] & (21) \\ I[T+1] - I[T] = I_n[T] - M_n[T]. & (22) \end{cases}$$

Note that I_n and M_n are independent of each other. As stated in Section I, our model of random propagation can be flexibly incorporated in the Markov chain of other epidemic models, just like the standard nonlinear model. To give an example of this, we present below five Markov chains with different combinations of infection terms (the linear propagation model, the literature standard nonlinear model, and our proposed model) and mitigation terms (the individual mitigation of detected infections and our proposed group mitigation).

A. DTMC 1: Linear Propagation Model + Individual Mitigation

This Markov chain shows the most simple SIS dynamics possible: the linear propagation model of (4) with the individual mitigation action (detected infections are cleaned one-by-one)

$$\begin{cases} S[T+1] - S[T] = -\alpha S[T] + \beta I[T] & (23) \\ I[T+1] - I[T] = \alpha S[T] - \beta I[T]. & (24) \end{cases}$$

B. DTMC 2: Literature Standard Nonlinear Propagation Model + Individual Mitigation

This Markov chain shows the usual dynamics using the literature standard nonlinear model (note how the system would be linear if $I[T]/N$ were not multiplying α)

$$\begin{cases} S[T+1] - S[T] = -\dfrac{\alpha S[T] I[T]}{N} + \beta I[T] & (25) \\ I[T+1] - I[T] = \dfrac{\alpha S[T] I[T]}{N} - \beta I[T]. & (26) \end{cases}$$

C. DTMC 3: Proposed Propagation Model + Individual Mitigation

This Markov chain substitutes the literature standard nonlinear propagation model by our proposed model and keeps the individual mitigation. We highlight how little (27) and (28) increased in complexity compared to (25) and (26). In return, the model changed from approximated to exact. Since both equations are nonlinear and cannot be transformed into a closed-form solution, our model offers an accurate alternative

$$\begin{cases} S[T+1] - S[T] = -S[T] + S[T]\left(1 - \dfrac{\alpha}{N}\right)^{I[T]} + \beta I[T] & (27) \\ I[T+1] - I[T] = S[T] - S[T]\left(1 - \dfrac{\alpha}{N}\right)^{I[T]} - \beta I[T]. & (28) \end{cases}$$

D. DTMC 4: Literature Standard Nonlinear Propagation Model + Proposed Group Mitigation

This Markov chain reverts back to the literature standard nonlinear propagation model but applies our proposed mitigation system. Note that although the propagation model is not exact (i.e., it will present errors compared to simulations, see Section VI-C), the mitigation model is

$$\begin{cases} S[T+1] - S[T] = -\alpha \dfrac{S[T] I[T]}{N} + I[T]\left(\beta + GI[T]\beta \dfrac{(1-\beta)}{N}\right) & (29) \\ I[T+1] - I[T] = \alpha \dfrac{S[T] I[T]}{N} - I[T]\left(\beta + GI[T]\beta \dfrac{(1-\beta)}{N}\right). & (30) \end{cases}$$

E. DTMC 5: Our Complete Model, With Proposed Propagation Model + Proposed Group Mitigation

This Markov chain joins together our models of propagation and group mitigation. Besides the improvement in accuracy of the propagation model, the mitigation system is more effective against malware propagation based on a single tuning parameter, the Mitigation Multiplicator G. In Section VII, we will show concrete improvements in prevention of malware propagation

$$\begin{cases} S[T+1] - S[T] = -S[T] + S[T]\left(1 - \dfrac{\alpha}{N}\right)^{I[T]} \\ \qquad\qquad + I[T]\left(\beta + GI[T]\beta \dfrac{(1-\beta)}{N}\right) & (31) \\ I[T+1] - I[T] = S[T] - S[T]\left(1 - \dfrac{\alpha}{N}\right)^{I[T]} \\ \qquad\qquad - I[T]\left(\beta + GI[T]\beta \dfrac{(1-\beta)}{N}\right). & (32) \end{cases}$$

VI. Validation of Proposed Model

In order to demonstrate the validity of our model and compare its performance with literature models, we developed a stochastic simulation that performs infections and mitigation actions device by device. Setting the same parameters for the Markov chains of the last section and the simulation, we evaluated the Markov chains and ran the simulation to check how well they matched. We used the simulation as an independent validation because while (31) and (32) represent the state of entire populations without keeping track of individual devices, the simulation keeps track of every device' state. Moreover, while the infection and detection rates are deterministic parameters in the equations, the simulation performs a random test

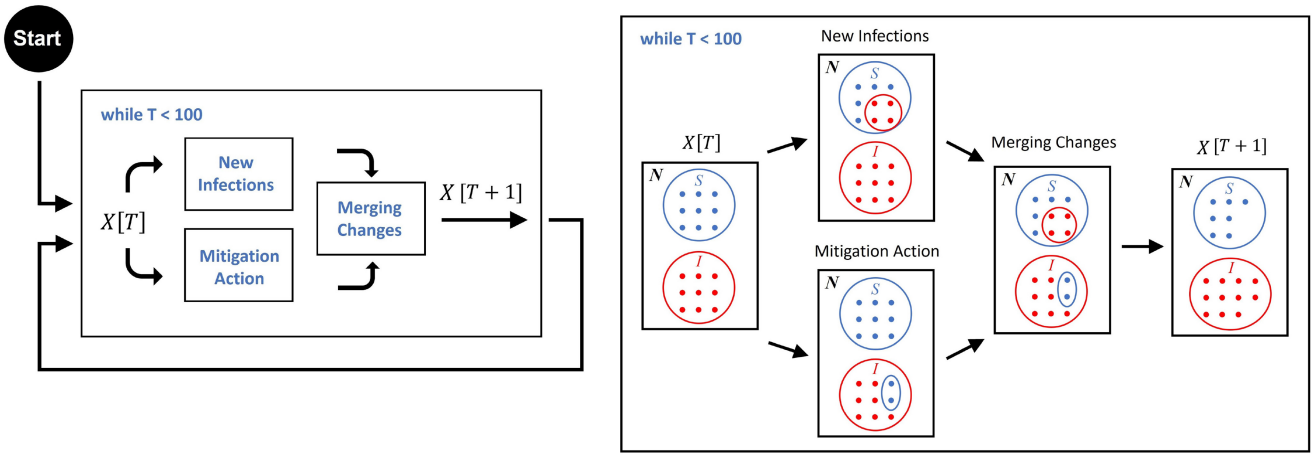

Fig. 5. Simulation setup. (a) Flowchart of simulation where every box represents the modification of a matrix containing the state of all devices. (b) Detailed flowchart boxes showing how the new infections and the mitigation action are calculated independently and then merged into the next state of the system.

against these rates at every infection and detection attempt. By running the simulation against a sufficiently big population, the variability of the stochastic approach is neutralized and a fair independent comparison becomes possible.

Fig. 5 shows how the simulation was implemented. Fig. 5(a) of figure is the fluxogram of simulation. Fig. 5(b) is a visual explanation of the nonintersectionality of infection and mitigation action. Since these two actions are independent from each other and depend only on the values of S and I at the beginning of the turn, it is necessary to make a copy at the beginning of the turn of the matrix containing the state of every device (one copy for each action). After the state of all affected devices is changed by each action in its respective copy, the copies are merged. This merged matrix contains the state of devices for the next turn. Such merge does not incur in conflict between new states because the changes caused by each action never occur on the same devices. To verify this, compare Fig. 5(b) with (18) and (20).

After a brief comment on the choice of parameters for the simulations, we will validate separately the proposed DTMCs of infection and mitigation, then compare the fitness of literature models with the fitness of our complete DTMC model.

A. Parameters

The variability of stochastic simulations is known to start subsiding when the size of population reaches 10^4. After noticing some residual variability at this size, we increased the population to 10^6, which showed consistent results with negligible variability (<0.1%).

Two important parameters are the infection rate α and detection rate β. It is well known that their ratio R_0 (the "basic reproduction number") defines the behavior of propagation of malware. In the results below, we will use a low-big range of values for α (25%–75%) and low-average values for β (12.5%–50%). With these ranges we set three values for R_0: 0.5, 1, and 2.

Since the network is closed and of fixed size, simulations start with a percentage of devices already infected from the beginning, which then propagate the infection to other devices inside the network. For the individual validation of our models of infection and mitigation, $(I[0]/N) = 50\%$. For all other comparisons, $(I[0]/N) = 10\%$. All results (simulation and Markov chain models) will be normalized by N to represent percentages of the total population and facilitate the qualitative analysis of malware propagation.

B. Validation of Infection and Mitigation Terms

Table III presents four metrics to evaluate the accuracy of prediction of five different DTMCs. In this validation section, we studied the DTMCs "Only Infection" and "Only Mitigation," each included only one of our models (meaning that (21) and (22) contained only $I_n[T]$ or $M_n[T]$). The first two metrics (Max E and RMSE) compare the results of the DTMCs with the result of the simulation. Since the SIS model has only two states, we calculated the four quantities only for the curve of infected devices I (the curve for S is symmetric in respect to the value of 50%). The first value, Max E, represents the maximum error between DTMC and simulation. RMSE represents the root-mean-square error of the whole curve of I between one of the DTMCs and the simulation, and its calculation is presented in (33). $I[T_f]$ represents the final value—or convergence value—of the infected population in the DTMC. At last, T_c represents the time constant of the convergence in the DTMC. "Time constant," here, is defined as how many turns it takes for $I[T]$ to reach 95% of its final value

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^{T_{\max}} (I_{\text{sim}}[k] - I_{\text{dtmc}}[k])^2}{T_{\max}}}. \tag{33}$$

The first two DTMC comparisons with the simulation show results using only infection or mitigation. To validate the "group" aspect of our mitigation strategy, we set $G = 2$. The results of this section use the initial population of infected

TABLE III
COLUMN 2 AND 3: VALIDATION OF PROPOSED I_n AND M_n. COLUMN 4-5: FITNESS OF DIFFERENT DTMC MODELS

DTMC model	Only Infection			Only Mitigation (G=2)			Linear			Standard Nonlinear			Proposed (G=0)		
Param / Metrics	α			β			R_0			R_0			R_0		
	25%	50%	75%	12.5%	25%	50%	0.5	1	2	0.5	1	2	0.5	1	2
Max E	(%)			(%)			(%)			(%)			(%)		
	0.0579	0.0561	0.0644	0.0663	0.0283	0.0621	33.33	47.26	46.59	0.0838	0.30	3.19	0.0503	0.11	0.15
RMSE	(%)			(%)			(%)			(%)			(%)		
	0.0154	0.0125	0.0144	0.0252	0.0103	0.0113	32.85	44.76	30.62	0.0172	0.21	2.35	0.0999	0.0435	0.0629
I[T$_f$]	100%	100%	100%	0%	0%	0%	33.3%	50.0%	66.7%	0%	2.85%	49.6%	0%	2.63%	46.6%
T$_c$	14	8	6	17	7	4	3	4	8	4	41	35	4	37	35

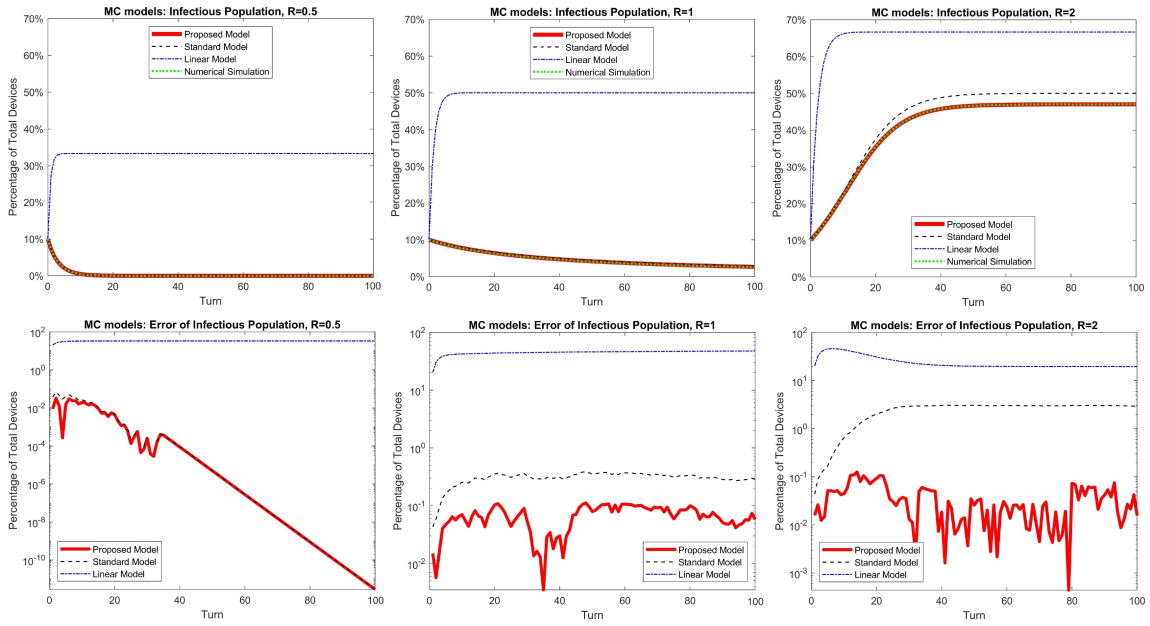

Fig. 6. Comparing fitness of DTMC models for three values of R. Top: progression of infected population. Bottom: error of infected population between DTMC models and simulation (y-axis in log scale).

devices of 50% of N. For Only Infection comparisons, we varied the parameter α to demonstrate the accuracy of the model. For Only Mitigation, we varied the parameter β.

As can be seen in Table III, the Max Error of all cases in the first two columns are smaller than 0.07% of N. The RMSE is smaller than 0.026% of N in all cases, showing that the Max Error is just a spike and the average error is much smaller. Considering that some error is expected due to the stochastic nature of the simulation, the accuracy is significantly high, especially, when compared with literature models (see next section). We conclude that both individual models of infection and mitigation are exact matches to the phenomena of random propagation and group mitigation.

C. Comparison of Proposed Model With Literature Models

After validating the proposed individual DTMC models, we compared the proposed complete model with the two most common literature models. First is the linear model [(23) and (24)], whose infection term was presented in (4). Its results are in column "Linear" of Table III.

The second is the standard nonlinear model [(25) and (26)], whose infection term was presented in (5). Its results are in column "Standard Nonlinear" of Table III.

For this comparison, we used only our propagation model [(27) and (28)] in order to evaluate the difference between propagation models isolatedly. Its results are in column "Proposed" of Table III.

In addition to the table results, we show the state trajectory of all three models and of the simulation in Fig. 6. Top side shows the curves of infected population, bottom side shows the errors between every DTMC and the simulation in log scale.

The first result to discuss is the validation of our proposed model. The Max Error of infected population in the proposed DTMC ranges between 0.05% and 0.15% of N, while the RMSE is smaller than 0.1%. Notice how this was achieved over the three conceptual values of R_0: $R_0 < 1$, $R_0 = 1$, and $R_0 > 1$. Considering how the complete model adds the

errors of variability of two stochastic models, the total error is still consistently very small, demonstrating the equivalence of the DTMC model and the simulation. Therefore, the results confirm that the proposed DTMC is an exact model for the phenomenon of random propagation. In the sequence, we will analyze how well the two most popular DTMC models in the literature fare against the simulation results.

The linear model shows huge errors, ranging from 20% to 50% of the total population. As expected, this shows that the convenience of smooth algebraic manipulation and closed-form equations are not worth the resulting inaccuracies, due to the overly simplistic modeling of random propagation. Equally problematic is the conceptual error of its $I[T_f]$ values. While malware is totally eradicated in simulations where the reproduction factor is $R_0 = 0.5$ and $R_0 = 1$ (see $I[T_f]$ of column Proposed) the infected population converges in the linear model to 33% and 50%, respectively. For $R_0 < 1$, malware replication should be slower than its mitigation, reaching 0% of N or converging to it by the end of the simulation. But in the linear model, the infected population converges to values as high as 50%. Moreover, T_c may be almost the same value (3 or 4 turns) for $R_0 < 1$ in all models, but for $R_0 \geq 1$ the two other models require at least 35 turns to converge to 95% of $I[T_f]$, while the linear model converges after only four turns. In conclusion, this model is inadequate to model random propagation.

The standard nonlinear model has varying degrees of accuracy. For simulations of $R_0 \leq 1$ (malware propagation slower than mitigation) it shows convergence as accurate as our proposed model's: $I[T_f]$ is equal or converges to 0, and T_c takes 4 or 41 turns, compared to 4 or 37 turns in simulation). The quantitative errors are also acceptable, if not so accurate as our proposed model: Max Errors are 0.3% or smaller, and RMSEs are 0.21% or smaller (the $R_0 = 0.5$ case has RMSE 0.07% smaller than our model's, but due only to variability). However, for $R_0 > 1$ the errors become nonnegligible. The Max Error is 3.19% and the RMSE is 2.35%, increasing as R_0 increases. Moreover, this error shows up at the equilibrium state, making it more problematic for predictions of malware propagation in complex models with many states. Considering how the most important use of predictive models of malware propagation is when $R_0 > 1$ (i.e., when the propagation threatens to take over the susceptible population if no additional mitigation measure is taken), errors of 3% are relevant enough to justify switching models to a more complex but very accurate one.

D. In-Depth Analysis of Literature Standard Nonlinear Model

During our analysis of the standard nonlinear model, we wanted to investigate further the modeling concept of αSI: "to model random propagation as the number of encounters between susceptible and infected devices." Moreover, we wanted to understand how it is related to our conclusions about the necessity of modeling "wasted attacks" in random propagation.

Since random propagation possesses two mechanisms of wasted attacks, discussed at length in Section III, we created other simulation setups removing one of the two wasting mechanisms and comparing again the simulation with the standard nonlinear DTMC. One simulation setup in particular yielded unexpected results: a simulation with pseudo-random selection of targets for attacks. It has the waste mechanism of attacking an already infected device, but not the repetition of targets. In other words, every attacker in this simulation chooses his target randomly from a list of targets (the entire network) and removes it from the list. In a real scenario, this elimination of addresses from a list of targets can only happen with coordination between the attacks, which is impossible in self-propagation where malware replicates by attacking randomly from infected devices (it is as if every attacker chooses an address from a list of targets without removing the address). Refer to Fig. 1 for a visual explanation of the difference.

TABLE IV
ERROR BETWEEN THE STANDARD NONLINEAR DTMC MODEL IN LITERATURE (STD) AND TWO SIMULATION SETUPS: RANDOM MALWARE PROPAGATION (SIM) AND RANDOM PROPAGATION WITHOUT REPEATED TARGETS (SIM*)

Model	Sim	Sim*	Std	Sim	Sim*	Std
Param / Metrics	$R_0 = 2$			$R_0 = 4$		
Max E	3.1%	0.19%	-	6.06%	0.19%	-
RMSE	2.69%	0.09%	-	4.62%	0.06%	-
I[T_f]	47.0%	50.0%	50.0%	70.2%	75.0%	75.0%
T_c	38	37	37	15	14	14

Table IV shows the comparison of the standard nonlinear model with the original simulation setup (column "Sim") and the pseudo-random simulation setup (column "Sim*"). Since the last section showed how the inaccuracy of the standard nonlinear model becomes relevant only for $R_0 > 1$, we generated results for $R_0 = 2$ and $R_0 = 4$, and none for $R_0 \leq 1$. Compare the Max Error and the RMSE of both simulation setups. Errors between the standard nonlinear DTMC and the pseudo-random simulation setup have the same order of magnitude than errors of our proposed DTMC with the original simulation setup. $I[T_f]$ and T_c for Sim* are exactly the same too, while $I[T_f]$ increases with R_0 for Sim and T_c is delayed one turn.

This result shows that the standard nonlinear model, widespread in the literature, is not a good model for truly random propagation of malware but is an exact match for models of pseudo-random propagation. In other words, the modeling of encounters of susceptible and infected devices represented in αSI possesses one mechanism of wasted attacks (the targeting of already infected devices), but not the two present in true random propagation.

Interestingly, we could not find this conclusion in any other paper that used the standard nonlinear model, which suggests the research community is largely unaware of it despite its big popularity. Naturally, if a security threat is human coordinated and chooses targets randomly by exhauring a list of addresses, this model is a perfect fit. But if the threat is a typical uncoordinated random self-replicating malware, our propagation model (18) is an exact fit with negligible

TABLE V
IMPACT OF GROUP MITIGATION ON THE METRICS OF INFECTION

Rates (α / β)	Low-Low (25% / 12.5%)						High-Low (75% / 37.5%)			High-High (95% / 90%)		
Metrics \ G	0	2	7	9	20	40	0	2	12	0	1	5
$I[T_f]$ (%)	47.0	25.3	11.9	9.90	5.09	2.68	41.7	26.5	9.87	3.65	3.38	2.74
$D[t_f]$ (%)	5.91	3.18	1.52	1.26	0.623	0.333	15.7	9.94	3.71	3.27	3.04	2.45
$C[t_f]$ (%)	5.91	9.54	12.2	12.6	13.1	13.7	15.7	29.8	48.2	3.27	6.09	14.7
$C[t_0]$ (%)	1.25	3.70	10.0	12.5	26.6	51.0	3.74	11.3	48.6	8.99	18.0	54.0

additional computational cost, since there is only three more operations compared to the three already present in the standard nonlinear model. Moreover, (18) can simply be added to other terms in order to form more complex compartmental models, as long as the total number of devices does not change and all scenarios of attack can be grouped in the four virtual states Finally, delay mechanisms can be inserted in the infection term by simply substituting T by $T - \lambda$, where λ represents a delay in the transition of state. In future works, we will demonstrate all these algebraic manipulations and the versatility of our model.

VII. EFFECTIVENESS OF GROUP MITIGATION

Following the description of the particular vulnerabilities of IoT networks made in Section I, we proposed a mitigation system that exploits the same logic of massification to counteract the big numbers and low specifications of IoT devices, based on grouping devices for cleaning according to the parameter G. The objective of group mitigation is to improve the mitigation of malware if the detection rate is low, or equivalently low (for example when the scans are slower that the propagation of malware), and when it is not possible to increase the detection rate either because of technical or economic reasons.

The strategy of group mitigation presents a tradeoff: on the one hand, malware propagation (best represented by $I[T_f]$) is decreased by increasing G; on the other hand, in real-world scenarios the cleaning of infected devices incurs in some type of disruption of the network, usually related to the interruption of normal activities on devices. An example is the cleaning of malware stored in RAM memory: cleaning a device is as simple as resetting it, but it becomes nonoperational until reboot. In such a case, mass-cleaning devices will increase the cost of nonoperational devices. Therefore, the analysis of effectiveness of our proposed group mitigation strategy includes the analysis of this tradeoff. By assuming that every target of mitigation is reset and becomes temporarily deactivated, we can calculate the cost in a number of ways. One way is to model an additional state in the dynamical equation of our proposed DTMC to represent resetting devices. However, we want to keep the SIS model as simple as possible, in order to investigate the most basic characteristics of random propagation. Another way is to just count the number of mitigated devices by our group strategy, which can be done by calculating C in (19).

By evaluating $I[T_f]$ and $C[T_f]$ as G increases, we can understand how effective the group mitigation is at decreasing malware propagation and at the same time how much percent of the network becomes temporarily unavailable to achieve it (specifically, $I[T_f]$ gives the value of infected devices at equilibrium while $C[T_f]$ gives the value of devices being reset every turn at equilibrium). Moreover, we can define the best value for G according to a tradeoff between $I[T_f]$ and $C[T_f]$. There are many ways to define an acceptable tradeoff, according to the possibility and willingness of the network management to deactivate higher percentages of the network temporarily ($C > 25\%$) in order to decrease $I[T_f]$ or even eradicate the malware. In this study, we defined this tradeoff as the value of G that resets a number of devices equal to the number of infections. Note that the reset devices will not all be infected, because the group mitigation is random and some susceptible devices will be chosen, therefore, the malware epidemic will not be eradicated. But such a conservative tradeoff will still demonstrate its effectiveness if the mitigation system can decrease the number of infections without changing the detection rate.

To make this evaluation, we performed many simulations with a wide range of parameters and selected the most representative results, shown in Table V. The first selection criterion is the basic reproduction number R_0. Since we want to investigate the potential of stronger malware mitigation, we prefer scenarios where the initial number of infections will not be naturally decreased by effective detection, so the table results all have $R_0 > 1$. The second criterion is the relationship of group mitigation with low and high infection/detection rates. Three scenarios were selected: 1) low α with low β; 2) high α with low β; and 3) high α with high β (note that low α with high β incurs in $R_0 < 1$). These scenarios are presented in the three main columns of Table V as "Low–Low," "High–Low," and "High–High," with rates of (25%/12.5%), (75%/37.5%), and (95%/90%), respectively. Finally, we selected a progression of values of G to see how the metrics of infection changed as the group mitigation became more "aggressive" (i.e., more devices were reset for each identified infection). We started with $G = 0$ (no group mitigation) and increased G until $I[T_f]$ and $C[T_f]$ matched. As G increased, we noticed a new behavior in group mitigation: an "overshoot" of initial resets $C[T_0]$, quickly falling and converging to approximately half of the initial number. In other words, when G increases significantly, a minimal number of detections will be multiplied many times over by G and generate a very strong initial mitigation, which makes the number of infection fall and converge rapidly. This in turn also decreases the number of resets rapidly. To

investigate how the value of this overshoot increased with G, we continued to increase G until $C[T_0]$ reached 50% of all devices in the network. A final detail of the table is an additional metric: $D[T_f]$ represents the number of detections by the scanning system at equilibrium, equal to $\beta I[T_f]$. By comparing this metric with $C[T_f]$ we can see how much percent of the network is reset by mitigation despite not being infected.

Analyzing the results, we can see how $I[T_f]$ is consistently decreased as G increases over all ranges of parameters. However, there are diminishing returns as the rate of infection increases (compare the top side of Fig. 7 with the bottom side to see results without and with group mitigation). Moreover, the ability of increasing G is greatly limited in High–Low and High–High scenarios: the tradeoff limit ($I[T_f] = C[T_f]$) is reached in both scenarios for $G = 2$ and $G = 0$, while it is reached at $G = 7$ in the Low–Low scenario. The hard limit of $C[T_0] = 50\%$ is also reached faster: $G = 40$ for Low–Low, $G = 12$ for High–Low, and $G = 5$ for High–High.

Given the tradeoff criterion and the necessary G to achieve it, the results confirm the effectiveness of group mitigation in Low–Low and High–Low scenarios and its negligible impact in the High–High scenario. In Low–Low, $I[T_f]$ decreases from 47.0% to 11.90% at the cost of $C[T_f]$ increasing from 5.91% to 12.2% (the closest value to $I[T_f]$). In High–Low, $I[T_f]$ decreases from 41.7% to 26.5% while $C[T_f]$ increases from 15.7% to 29.8% (a big percentage of the network, although at $G = 0$ the number of resets is already big given the High-Low rates). Considering the goal of decreasing infections significantly without increasing the detection rate and without impacting the system beyond what the infections are already impacting, group mitigation is very successful in Low–Low and High–Low scenarios. But in the High–High scenario, even without mitigation ($G = 0$) $I[T_f]$ converges naturally to a small value of 3.65%. Any application of group mitigation will disproportionately reset more devices than decrease infections.

In practical terms, this analysis confirms our design assumptions and expectations. Since group mitigation was designed to counteract the mass deployment of devices with mass mitigation, circumventing the lack of proper and efficient security commonly found in IoT networks, scenarios of mitigation where the detection rate is low are significantly improved by group mitigation. However, scenarios with high detection rates do not benefit from group mitigation, since they do not really need it in the first place.

A final important note is that the quantitative values presented above ($I[T_f]$, $C[T_f]$, $C[T_0]$, and the value of G that produces them) are dependent on α and β. However, the qualitative/conceptual results in terms of Low–Low, High–Low, and High–High infection/detection rates are independent. In other words, the criterion for choosing the value of G should be based on the acceptable cost of mitigation, considering the behavior of the three groups of results discussed above.

VIII. Sensitivity To False Positives and Negatives

Given the indiscriminate nature of the group mitigation and its exploitation of mass deployment to aggressively counteract malware, an important question is how sensitive the mitigation system can be to false positives (FPs), as well as the suitability of the mitigation system to compensate FNs. As mentioned in Section I, IoT networks are creating new challenges to security because of the mass deployment of low specs devices. Considering how FPs are the result of a detection system too strong, and FNs of a detection system too weak, IoT networks are more susceptible to FNs than FPs because it hard to implement a robust and sometimes computationally heavy detection system in such networks (either by implementing it in every low spec device, or in a single server that can monitor an entire network of mass-deployed devices with high accuracy). Coupled with the lack of implementation of strong security measures and maintenance in millions of devices, this leads to a detection system that is usually undertuned, not overtuned. Moreover, a big part of the growing IoT application does not involve user interaction with webpages and dubious Internet links (e.g., CCTV systems, Wireless Sensor Networks, or SmartHouses), therefore, the detection system does not need to be overtuned to account for a variety of conditions of malware infection. This is the reason why we focused our study in low detection rates β: it is harder to keep up with malware propagation in IoT networks compared to normal networks due to the sheer number of vulnerable devices, but at the same time the detection system does not need to be overtuned as Web-browsing anti-malware.

With these assumptions, we developed our mitigation system to compensate low (or slow) detection rates with mass mitigation of devices, also assuming that the operation of a significant part of these devices (up to 25%) can be indiscriminately deactivated for a short time if this can control an aggressive malware propagation. In such a scenario, the selling point of our system of mitigation is its expected robustness to FNs. With a small increase of the parameter G we achieved big increases in the mitigation of infections even in undetected devices (see results of last Section VII). Regarding FPs, due to the very design assumptions of our mitigation strategy, a system that overcompensates FNs with indiscriminate mass deactivation is expected to compound the errors of misidentification of infection of FPs and increase the cost of group mitigation unnecessarily.

To confirm these assumptions, we performed additional simulations with an FP rate (FPR) of 10% (a common value in overtuned detection systems). The results are shown in Table VI. The table contains the recalculation of tradeoff metrics of group mitigation ($I[T_f]$, $C[T_f]$, and $C[T_0]$) and how much percent of the network represents FNs and FPs at equilibrium ($F_n[T_f]$ and $F_p[T_f]$, respectively). Simulation results confirm that our mitigation strategy is not sensitive to FNs at all, but actually compensates for it as designed. In the scenarios Low–Low and High–Low, $F_n[T_f]$ is greatly decreased, both without and with FPs. However, in the scenario High–High, the decrease of FNs is negligible, confirming the conclusions of Section VII: group mitigation is not necessary in systems with high detection rates and effective individual mitigation. FPs, on the other hand, were confirmed to be a fundamental limitation of our proposed mitigation strategy. As Table VI shows, applying an FPR of 10% almost leads to a saturation

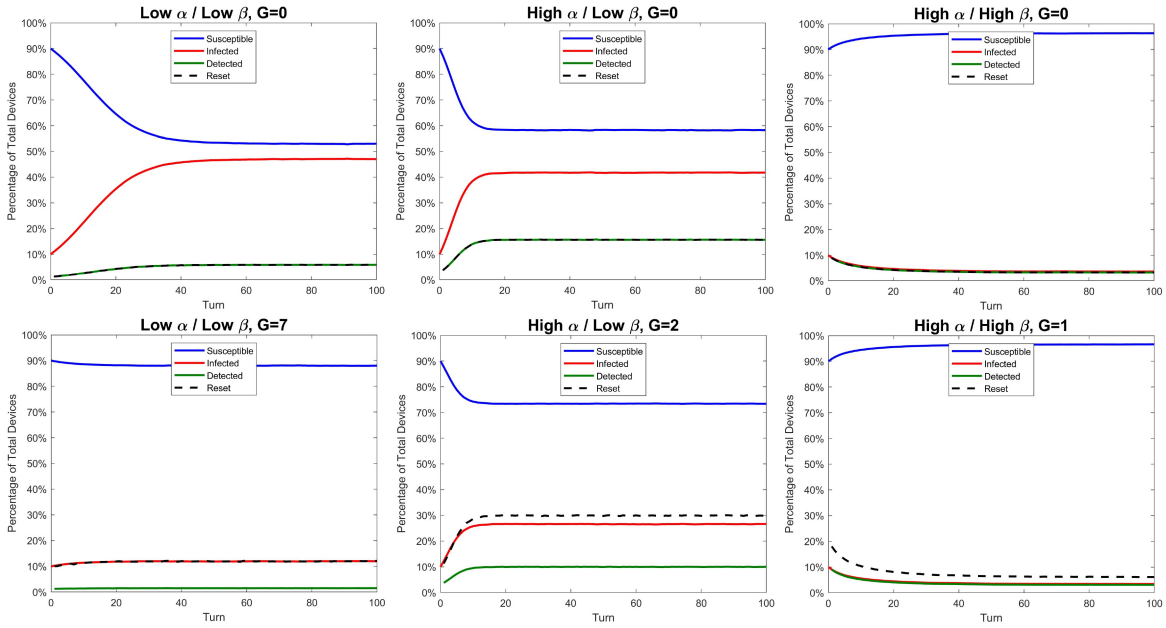

Fig. 7. Evaluation of effectiveness of group mitigation at decreasing malware propagation without improving the detection rate. Top figures: three scenarios of malware propagation regarding infections/detections (Low–Low, High–Low, and High–High). Bottom figures: same scenarios of malware propagation after applying group mitigation at a degree that matches number of infections and resets of devices at the steady state. By increasing G, the final steady-state value of infected population (red line) decreases, but the tradeoff is that the number of reset devices (black line) increases.

TABLE VI
SENSITIVITY OF GROUP MITIGATION TO FPS AND ITS IMPACT ON FNS

Rates		Low-Low		High-Low		High-high	
Metrics \ G		0	7	0	2	0	1
FPR = 0%	$I[T_f]$	47.0	11.9	41.7	26.6	3.65	3.39
	$C[T_f]$	5.91	12.2	15.7	30.0	3.27	6.09
	$C[T_0]$	1.25	10.0	3.76	11.3	8.99	18.0
	$F_n[T_f]$	41.1	10.4	26.1	16.6	0.367	0.340
	$F_p[T_f]$	0	0	0	0	0	0
FPR = 10%	$I[T_f]$	47.0	0	41.8	19.1	3.60	2.73
	$C[T_f]$	11.1	80.0	21.5	45.7	12.9	24.4
	$C[T_0]$	10.2	82.0	12.7	38.2	18.0	36.0
	$F_n[T_f]$	41.1	0	26.1	11.9	0.361	0.273
	$F_p[T_f]$	5.29	10.0	5.84	8.06	9.63	9.73

of resets in the Low–Low scenario with optimal $G = 7$ (80% of the network shuts down every turn). In the High–Low scenario with optimal $G = 2$, the resets increase from 30% to 45% when FPR is introduced, gaining in exchange only a reduction of 7% of $I[T_f]$ (from 26.6% to 19.1%). In other words, FPs severely limits the increase of G and the use of group mitigation.

Having taken these results into account, we conclude that our proposed mitigation system works well when applied to scenarios for which it was specifically designed: networks with limitations in its malware detection/mitigation system, which are not overtuned to have significant FPRs. Low-average detection rates play into the strenghts of our strategy, decreasing FNs significantly, but High detection rates nullifies the gains regarding FNs. FPRs of overtuned systems, however, introduces problems of mitigation saturation, due to the coupling between FPs and the Mitigation Multiplicator of group mitigation. A final observation comes from preliminary results of an ongoing study of how to decrease sensitivity to FPs: given the challenge of implementing detection systems in IoT networks that can keep up with malware propagation (roughly translating into low detection rates), our proposed group mitigation strategy demonstrates effectiveness as long as the detection system is implemented to have FPRs between 0% and 2% (well-tuned detection systems have FPRs of around 1%).

IX. CONCLUSION

In this article, we modeled and derived the exact Markov chain model of random propagation of malware. We used the epidemic compartmental model SIS to demonstrate how to model properly the two mechanisms of inefficiency in random propagation. We also proposed a system of group mitigation to address the vulnerabilities of IoT networks through calculated mass cleaning of random devices, and derived an exact Markov chain for this mitigation system.

To validate the two models, we created a simulation setup and evaluated independently the error of Markov chains for random propagation and group mitigation, finding errors of less than 0.07% for both. We also compared the errors of our propagation model with the two most popular Markov chains in literature, finding less than 0.15% in our model against ∼3% in the standard nonlinear model and ∼45% in the linear model. Moreover, we evaluated the effectiveness of group

mitigation, confirming its ability of improving mitigation of malware propagation for scenarios of low detection rate, and of decreasing infections without changing the detection rate itself. In the same tests we confirmed our design assumptions that group mitigation would be ineffective, but also unnecessary, in scenarios of high detection rates.

Future works will extend the results of this article to compartmental models more complex than the SIS model. We plan to add new states representing other mechanisms of infection and mitigation (e.g., incubation, immunity, and quarantine) and investigate how the parameters of these new mechanisms and the parameter G of group mitigation interact to affect propagation/mitigation behavior. Currently, we are also investigating to what extent our proposed mitigation strategy is sensitive to FPs and how to decrease this sensitivity. An important extension of our work would be the implementation of local dynamics generated by different network topologies, especially, ZigBee, 6LowPan, and variations of Wireless Sensor Networks. This would make our solution broader in its applicability, going beyond the intended use of CCTV or SmartHouse networks. For such topologies, specific group mitigation strategies that exploit the local dynamics should be developed (such as the mitigation system based on the concept of LRD, presented in our previous work on this topic [24]).

REFERENCES

[1] D. Evans. "The Internet of Things—How the next evolution of the Internet is changing everything." Cisco. 2011. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[2] S. Bagchi et al., "New frontiers in IoT: Networking, systems, reliability, and security challenges," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11330–11346, Dec. 2020, doi: 10.1109/JIOT.2020.3007690.

[3] M. B. M. Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019, doi: 10.1016/j.comnet.2018.11.025.

[4] M. Asplund and S. Nadjm-Tehrani, "Attitudes and perceptions of IoT security in critical societal services," *IEEE Access*, vol. 4, pp. 2130–2138, 2016, doi: 10.1109/ACCESS.2016.2560919.

[5] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, Jul. 2017, doi: 10.1109/MC.2017.201.

[6] E. Bertino and N. Islam, "Botnets and Internet of Things security," *Computer*, vol. 50, no. 2, pp. 76–79, Feb. 2017, doi: 10.1109/MC.2017.62.

[7] D. B. B. Herzberg and I. Zeifman. "Breaking down Mirai: An IoT DDoS botnet analysis." 2016. [Online]. Available: https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html

[8] C. Seaman, *Threat Advisory: Mirai Botnet*, Akamai Threat Advisory, Cambridge, MA, USA, 2016.

[9] M. Wazid, A. K. Das, J. J. P. C. Rodrigues, S. Shetty, and Y. Park, "IoMT malware detection approaches: Analysis and research challenges," *IEEE Access*, vol. 7, pp. 182459–182476, 2019, doi: 10.1109/ACCESS.2019.2960412.

[10] Ö. A. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[11] M. J. Farooq and Q. Zhu, "Modeling, analysis, and mitigation of dynamic botnet formation in wireless IoT networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 2412–2426, 2019, doi: 10.1109/TIFS.2019.2898817.

[12] T. Wang et al., "Propagation modeling and defending of a mobile sensor worm in wireless sensor and actuator networks," *Sensors*, vol. 17, no. 1, pp. 139–155, 2017, doi: 10.3390/s17010139.

[13] A. M. Del Rey, "Mathematical modeling of the propagation of malware: A review," *Security Commun. Netw.*, vol. 8, pp. 2561–2579, Oct. 2015, doi: 10.1002/sec.1186.

[14] A. Mahboubi, S. Camtepe, and K. Ansari, "Stochastic modeling of IoT botnet spread: A short survey on mobile malware spread modeling," *IEEE Access*, vol. 8, pp. 228818–228830, 2020, doi: 10.1109/ACCESS.2020.3044277.

[15] D. Acarali, M. Rajarajan, N. Komninos, and B. B. Zarpelão, "Modelling the spread of botnet malware in IoT-based wireless sensor networks," *Security Commun. Netw.*, Feb. 2019, Art. no. 3745619, doi: 10.1155/2019/3745619.

[16] F. K. Batista, A. M. Del Rey, and A. Queiruga-Dios, "A new individual-based model to simulate malware propagation in wireless sensor networks," *Mathematics*, vol. 8, no. 3, pp. 410–432, 2020, doi: 10.3390/math8030410.

[17] M. Karanja, S. Masupe, and M. Jeffrey, "Modelling malware propagation on the Internet of Things using an agent based approach on complex networks," *Jordanian J. Comput. Inf. Technol.*, vol. 6, no. 1, pp. 26–40, Jan. 2019, doi: 10.5455/jjcit.71-1568145650.

[18] S. White, A. M. Del Rey, and G. Sánchez, "Using cellular automata to simulate epidemic diseases," *Appl. Math. Sci.*, vol. 3, pp. 959–968, Jan. 2009.

[19] P. Van Mieghem, J. Omic, and R. Kooij, "Virus spread in networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 1–14, Feb. 2009, doi: 10.1109/TNET.2008.925623.

[20] S. Kondakci and D. D. Doruk, "Building epidemic models for living populations and computer networks," *Sci. Progr.*, vol. 104, no. 2, pp. 1–40, Jun. 2021, doi: 10.1177/00368504211017800.

[21] M. Ficco, "Detecting IoT malware by Markov chain behavioral models," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, 2019, pp. 229–234.

[22] K. Lin, M. Chen, J. Deng, M. M. Hassan, and G. Fortino, "Enhanced fingerprinting and trajectory prediction for IoT localization in smart buildings," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1294–1307, Jul. 2016, doi: 10.1109/TASE.2016.2543242.

[23] G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "Characterization and prediction of mobile-app traffic using Markov modeling," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 907–925, Mar. 2021, doi: 10.1109/TNSM.2021.3051381.

[24] R. M. Carnier, Y. Li, J. Shikata, and Y. Fujimoto, "Modeling of malware propagation in IoT network and mitigation in area based on local routing depth," in *Proc. 7th IEEJ Int. Workshop Sens. Actuation Motion Control Optim. (SAMCON)*, Mar. 2020, pp. 336–341.

Rodrigo Matos Carnier (Member, IEEE) received the B.E. degree in mechatronics engineering from the University of Sao Paulo, Sao Paulo, Brazil, in 2013, and the M.E. and Ph.D. degrees in electrical and computer engineering from Yokohama National University, Yokohama, Japan, in 2017 and 2021, respectively.

From 2020 to 2021, he was a Research Assistant with Fujimoto Laboratory, Yokohama, where he has been a Postdoctoral Researcher since 2021. His research interests include robotics, optimization, machine learning, dynamical modeling, and Internet of Things.

Dr. Carnier was also a recipient of the MEXT/Monbukagakusho Scholarship from The Ministry of Education, Culture, Sports, Science, and Technology of Japan from 2015 to 2020.

Yue Li was born in Henan, China, in 1994. He received the B.S. degree in mechanical engineering from Dalian University of Technology, Dalian, China, in 2017, and the M.S. degree in electrical and computer engineering from Yokohama National University, Yokohama, Japan, in 2020, where he is currently pursuing the Ph.D. degree in electrical and computer engineering.

His research interests include robotics, motion control, and the IoT security.

Yasutaka Fujimoto (Senior Member, IEEE) was born in Kanagawa, Japan. He received the B.E., M.E., and Ph.D. degrees in electrical and computer engineering from Yokohama National University, Yokohama, Japan, in 1993, 1995, and 1998, respectively.

He was with the Department of Electrical Engineering, Keio University, Yokohama, in 1998. Since 1999, he has been with the Department of Electrical and Computer Engineering, Yokohama National University, where he is currently a Professor. His research interests include actuators, robotics, manufacturing automation, and motion control.

Dr. Fujimoto was the recipient of the IEEE/ASME TRANSACTIONS ON MECHATRONICS Best Paper Award in 2020. He is an Associate Editor for the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and a Vice Chief for the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS. He is a member of the Robotics Society of Japan.

Junji Shikata (Member, IEEE) received the B.S. and M.S. degrees in mathematics from Kyoto University, Kyoto, Japan, in 1994 and 1997, respectively, and the Ph.D. degree in mathematics from Osaka University, Osaka, Japan, in 2000.

From 2000 to 2002, he was a Postdoctoral Fellow with the Institute of Industrial Science, the University of Tokyo, Tokyo, Japan. Since 2002, he has been with the Graduate School of Environment and Information Sciences, Yokohama National University, Yokohama, Japan, where he is currently a Professor. From 2008 to 2009, he was a Visiting Researcher with the Department of Computer Science, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. His research interests include cryptology, information theory, theoretical computer science, and computational number theory.

Dr. Shikata received several awards, including the 19th TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2004, the Wilkes Award 2006 from the British Computer Society, and the Young Scientists' Prize, the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science, and Technology in Japan in 2010.