Prediction Method of Infection Spreading with CNN for Self-evolving Botnets

Keita Kishioka*, Koki Hongyo*, Tomotaka Kimura†, Takanori Kudo‡, Yoshiaki Inoue§, and Kouji Hirata¶
* Graduate School of Science and Engineering, Kansai University, Osaka 564-8680, Japan
Email: {k266175, k896955}@kansai-u.ac.jp
† Faculty of Science and Engineering, Doshisha University, Kyoto 610-0321, Japan
Email: tomkimur@mail.doshisha.ac.jp
‡ Faculty of Science and Engineering, Setsunan University, Osaka 572-8508, Japan
Email: t-kudo@ele.setsunan.ac.jp
§ Graduate School of Engineering, Osaka University, Osaka 565-0871, Japan
Email: yoshiaki@comm.eng.osaka-u.ac.jp
¶ Faculty of Engineering Science, Kansai University, Osaka 564-8680, Japan
Email: hirata@kansai-u.ac.jp

***Abstract*—Recently, machine learning has been extensively used and achieved significant results in many research areas. Accordingly, the literature has suggested the appearance of self-evolving botnets, which autonomously discover vulnerabilities and evolve by performing machine learning with computing resources of zombie computers. Our previous work have shown that the infection dynamics of self-evolving botnets depend on connection relations among hosts, through simulation experiments based on a Markov chain. This paper proposes a prediction method of the infection dynamics of self-evolving botnets, which uses a convolutional neural network (CNN). The proposed method predicts the level of infection spreading of self-evolving botnets, which depends on network structures and initial infected hosts, by using adjacency matrices of hosts as input data to CNN. In this paper, we show the effectiveness of the proposed method through performance evaluation based on data obtained from simulation experiments.**

I. Introduction

In recent years, malware has rapidly evolved and become sophisticated. For example, metamorphic malware rewrites and obfuscates its own source codes whenever it infects a new host [3]. Furthermore, there exist malware generation methods that generate new malware by combining known malware source codes [4], [11]. There also exist botnets, which consists of many hosts (named zombie computers) infected by the botnet malware [12]. The botnets are controlled by attackers to perform illegal attacks, and thus they have become serious cyber security threats.

Machine learning technologies also has rapidly evolved in the past. Deep learning [7], [5] especially has produced remarkable results in various research fields. Some researches use deep learning to discover bugs and vulnerabilities in software [13], [16]. Such researches have been done from the viewpoint of helping software creation and protection. However, malicious attackers can also use them to discover vulnerabilities and attack software. Furthermore, the malicious attackers can enhance the performance of vulnerability discovery by using distributed machine learning techniques [5], [10], [14], which perform learning with computing resources of inexpensive hosts.

Under these circumstances, in [8], the authors have suggested the appearance of a new type of botnets named self-evolving botnets. The self-evolving botnets discover unknown vulnerabilities by performing distributed machine learning with computing resources of zombie computers and evolve autonomously accordingly. Based on the discovered vulnerabilities, they infect other hosts, and then make themselves bigger by taking in the infected hosts. The authors have shown the threat of the self-evolving botnets, using an epidemic model with a continuous-time Markov chain. In addition, in [9], the authors have proposed an epidemic model considering overlay networks that represent connection relations among hosts. Through simulation experiments, they have shown that the spreading behavior of the self-evolving botnets strongly depends on initial infected hosts and overlay network structures. However, in the simulation experiments, the computation time greatly increases as the network size increases.

This paper proposes a prediction method of the infection spreading of self-evolving botnets, which aims at predicting the level of their spreading with not simulation experiments but a convolutional neural network (CNN) [6]. In the proposed method, we use adjacency matrices of hosts as input data to CNN. By doing so, the proposed method predicts the level of the infection spreading of the self-evolving botnets, which differs depending on initial infected hosts and structures of overlay networks constructed according to connection relations among hosts. Although CNN takes a long time to learn, it can rapidly identify the level of the infection spreading after learning. Furthermore, even in cases where connection relations among hosts change, the proposed method can predict the level of the infection spreading by adopting the corresponding adjacency matrix. In this paper, we show the effectiveness of the proposed method through performance evaluation based on data obtained from simulation experiments.

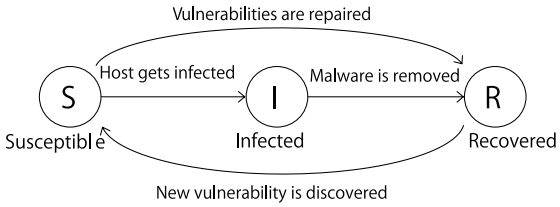

Fig. 1. SIRS model

II. Epidemic model for self-evolving botnets [9]

A. Modeling with a continuous-time Markov chain

The epidemic model for self-evolving botnets proposed in [9] represents the state of each host with a Susceptible-Infected-Recovered-Susceptible (SIRS) model shown in Fig. 1. In the SIRS model, "S" means a state where some vulnerabilities exist in the host (susceptible state), "I" means a state where the host is infected with the botnet malware (infected state), and "R" means a state where the host has no known vulnerabilities (recovered state). Note that hosts in the recovered state are protected from known vulnerabilities, but they are not protected from unknown vulnerabilities.

The infection process of a self-evolving botnet is formulated as a continuous-time Markov chain where the following events occur.

a) The self-evolving botnet discovers a new vulnerability by using distributed machine learning. When the number of hosts in the infected state I is v ($v = 1, 2, \dots$), an event that the self-evolving botnet discovers an unknown vulnerability occurs according to a Poisson process with the rate $\lambda_v = \eta(v+1)$, where η denotes the vulnerability discovery rate of each infected host and λ_v is proportional to v. In this case, all hosts in the recovered state R transition to the susceptible state S because they can get infected by being attacked the discovered vulnerability.

b) According to a Poisson process with the rate δ_{S}, each host in the susceptible state S repairs its own vulnerabilities and then transitions to the recovered state R.

c) According to a Poisson process with the rate α, each host in the infected state I infects each adjacent host in the susceptible state S and then the host getting infected transitions to the infected state I.

d) According to a Poisson process with the rate δ_{I}, each host in the infected state I removes the botnet malware from itself and then transitions to the recovered state R.

In the above event c), hosts in the susceptible state S get infected, and then transition to the infected state I. The epidemic model assumes that hosts in the infected state can only adjacent hosts on an overlay network consisting of hosts, which are constructed based on relationships among the hosts, because the infection of malware depends on the relationships such as frequently accessed web sites, their friendships, and physical network environments. In order to generally represent the structure of the overlay network, we use an adjacency matrix. Let $\boldsymbol{A} = \{a_{i,j} \mid i, j \in \mathcal{N}\}$ denote the adjacency matrix, where $a_{i,j}$ denotes (i, j) element in $\boldsymbol{A}$. $a_{i,j}$ is defined by

$$a_{i,j} = \begin{cases} 1, & \text{if host } i \text{ is adjacent to host } j, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

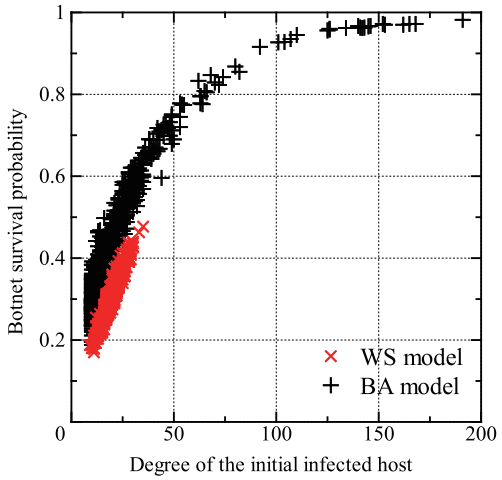

Fig. 2. Botnet survival probability against the degree of the initial infected host.

B. The level of the infection spreading of self-evolving botnets

In [9], the authors showed the infection dynamics of self-evolving botnets through simulation experiments, based on a continuous-time Markov chain. In what follows, we briefly explain the experiments. In the experiments, as overlay networks, two types of networks that are constructed based on the Watts-Strogatz (WS) model [15] and the Barabasi-Albert (BA) model [2], respectively, are used. They have special characteristics, i.e., the small world property and the scale-free property. It is known that many actual networks have these properties. The small world property is a property that any two hosts are connected with a small number of hops compared with the network size. The scale-free property is a property that most hosts connect to only a few other hosts while a few hosts directly connect to many other hosts. The WS model has the small-world property and the BA model has both properties. In each network, the number N of hosts is set to be $1,000$ and the parameters are set to be $\alpha = 0.1$, $\delta_{\mathrm{S}} = 1$, $\delta_{\mathrm{I}} = 0.1$, and $\eta = 0.01$. As the initial state at time $t = 0$, we assume situations where one host is infected and the remaining $N - 1$ hosts are in the susceptible state. We refer to the infected host at time $t = 0$ as the initial infected host.

Figs. 2 and 3 show the botnet survival probability at time $t = 100$ as a function of the degree and the closeness centrality, respectively, of the initial infected host, where the average degree k is set to be 20 in each network. The botnet survival probability is a probability that there still exist one or more infected hosts at time t. We calculate the botnet survival probability by selecting all the hosts as the initial infected

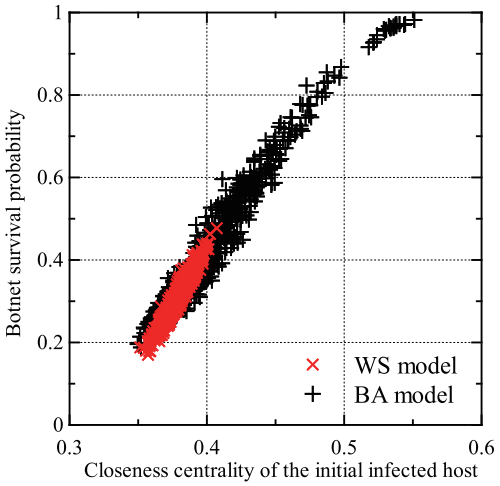

Fig. 3. Botnet survival probability against the closeness centrality of the initial infected host.

hosts and collecting 1,000 samples for each host. The botnet survival probability generally becomes stationary with time elapsed [9]. Specifically, it is almost stationary at time $t = 100$ in the simulation experiment. Therefore, we adopt the value of the botnet survival probability at time $t = 100$ as the level of the infection spreading in this paper. The closeness centrality of a host is an index indicating the distance from the host to every other host. The high closeness centrality means that the host is near from other hosts. As we can see from these figures, the botnet survival probability increases with the degree and the closeness centrality of the initial infected host. We also observe that the infectivity of the self-evolving botnet depends on the network structures.

As we have discussed above, the previous work has formulated the infection process of self-evolving botnets as continuous-time Markov chains and shown the level of the infection spreading through simulation experiments. On the other hand, the prediction method proposed in this paper aims at easily predicting the level of the infection spreading of self-evolving botnets, using CNN.

III. Prediction Method Using CNN

A. CNN

CNN is one of machine learning techniques and its special characteristic is using two layers called a convolution layer and a pooling layer. As shown in Fig. 4, a typical structure of CNN consists of some combinations of convolution layers and pooling layers. In this figure, the CNN consists of two combinations of the convolution layers and the pooling layers. After the iteration of these combinations, a fully connected layer is located and then an output layer outputs identification results for input data.

A convolution layer performs convolution operation for each element of input data and its filter. It extracts features of input data through filter processing. In CNN, learning is done by updating the filter by using an error back propagation method based on training data whose correct answer is known. A pooling layer adjusts the size of output data by performing subsampling. There are some types of pooling layers such as max pooling and average pooling. The max pooling outputs the maximum value among values within its window. The average pooling outputs their average value.

B. Proposed prediction method

1) Making of input data: The proposed prediction method predicts the level of infection spreading of a self-evolving botnet (i.e., the botnet survival probability), based on the structures of overlay networks, without simulation experiments. Specifically, the proposed method predicts the botnet survival probability by using adjacency matrices shown in (1), which represent connection relations among hosts on the overlay networks, as input data to CNN. The adjacency matrix $\boldsymbol{A}$ is an $N \times N$ square matrix, which is expressed by

$$\boldsymbol{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{pmatrix}.$$

As shown in (1), $a_{i,j} = 1$ if host i is adjacent to host j; otherwise, $a_{i,j} = 0$. The proposed method replaces the adjacency matrix with a binary image. Specifically, it makes an $N \times N$-size binary image, each pixel of which corresponds to each element in the adjacency matrix in such a way that the corresponding pixel is black (resp. white) if $a_{i,j} = 1$ (resp. $a_{i,j} = 0$). We use the binary image as input data to CNN.

The botnet survival probability depends on not only the structure of an overlay network but also an initial infected host. In order to distinguish the initial infected host from other hosts on the overlay network, we assume that host $i = 1$ is the initial infected host. In this case, the first row and the first column of the adjacency matrix represents the connection relations of the initial infected host. We create images for all the hosts of the overlay network as follows. We first create an image from the adjacency matrix. We then make a new adjacency matrix by updating the indices of hosts as $i \leftarrow i - 1$ for each host $i \in \mathcal{N} - \{1\}$ and $i \leftarrow N$ for host $i = 1$. The new adjacency has a shape in which the previous adjacency matrix is shifted to the upper left. By creating an image again from the new adjacency matrix where the host with index $i = 1$ is assumed to be the initial infected host, the proposed method can express a different host as an initial infected host. By repeating this process $N - 1$ times, the proposed method makes images that express all the hosts as initial infected hosts. Fig. 5 shows an example of images created by this process in the case of $N = 50$. As we can see from this figure, the arrangement patterns of the pixels are shifted from the lower right to the upper left.

2) Making of training data: The proposed method makes training data according to the following procedure.

1) Generate an overlay network.

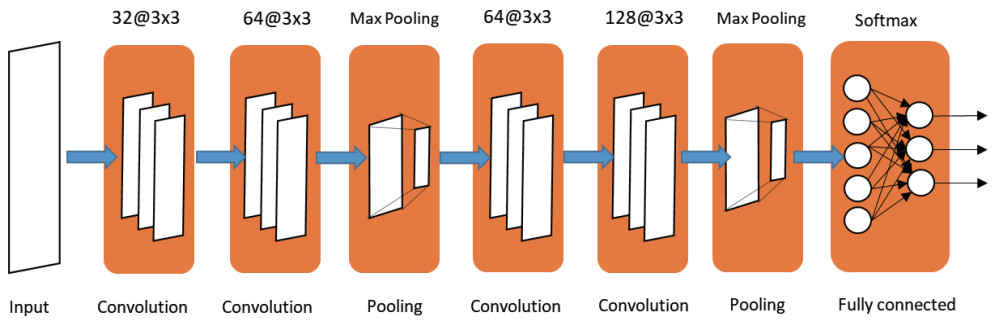

Fig. 4. Structure of CNN.

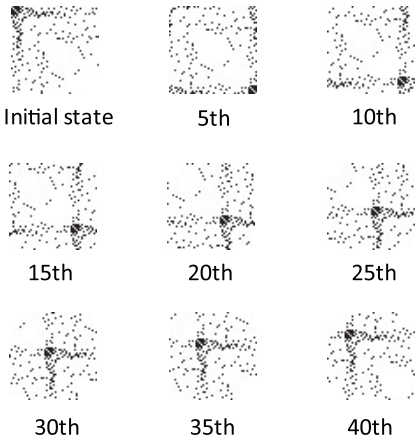

Fig. 5. Example of created images.

TABLE I
COMPUTER SPECIFICATION.

CPU	Intel(R) Xeon(R) W-2123 3.60GHz
RAM	16GB 2666MHz
GPU	NVIDIA Geforce GTX 1080Ti 11GB
OS	Windows 10 pro

2) Create an image for each initial infected host from the adjacency matrix of the overlay network.
3) Obtain the botnet survival probability for each initial infected host from the simulation experiment discussed in Section II-B.
4) Classify the corresponding image into n classes depending on the botnet survival probability.

These combinations of an image and a class are training data for CNN.

IV. EVALUATION

A. Model

We examine the prediction performance of the proposed method through performance evaluation using CNN based on training data obtained from simulation experiments. Python 3.6.2 and Keras 2.1.2 [1] are used to implement the CNN, the structure of which is shown in Fig. 4. The CNN consists of the convolution layers C1 and C2, the pooling layer P1, the convolution layers C3 and C4, the pooling layer P2, and the fully connected layer. C1 performs convolution operation with 32 filters of size 3×3, C2 and C3 perform convolution operation with 64 filters of size 3×3, C4 performs convolution operation with 128 filters of size 3×3, and P1 and P2 perform 2×2 max pooling operation. The ReLU function is used as an activation function in all the convolution layers. The fully connected layer classifies input data into n classes, using the softmax function. The computer specification used in this experiment is shown in Table I. We use GPU for learning of training data.

TABLE II
CLASSIFICATION ACCORDING TO THE BOTNET SURVIVAL PROBABILITY.

class	botnet survival probability
A	$0 \leq$ survival probability ≤ 0.2
B	$0.2 <$ survival probability ≤ 0.4
C	$0.4 <$ survival probability ≤ 0.6
D	$0.6 <$ survival probability ≤ 0.8
E	$0.8 <$ survival probability ≤ 1.0

TABLE III
RUNNING TIME FOR SIMULATION EXPERIMENTS.

WS model	$k = 4$	$k = 6$	$k = 8$	$k = 12$
	12:15.0	26:03.5	38:00.8	55:23.5
BA model	$k = 4$	$k = 6$	$k = 8$	$k = 10$
	10:33.4	19:45.2	28:20.9	35.26.5

We construct overlay networks based on the WS model and the BA model. Each network consists of $N = 100$ hosts and the parameters in the epidemic model are set to be $\alpha = 0.5$, $\delta_S = 1$, $\delta_I = 0.1$, and $\eta = 0.01$. The procedure of the evaluation is as follows.

1) Input training data to CNN for learning.
2) For each test data, create an image from the adjacency matrix of the corresponding overlay network.
3) Input the images of test data to CNN.
4) Verify whether the results of n-class classification for

TABLE IV
TIME CONSUMED FOR LEARNING OF CNN.

M	2,000	4,000	6,000
WS model	03:34.6	06:59.3	10:00.9
BA model	03:38.4	06:50.3	10:01.2

TABLE V
TIME CONSUMED FOR IDENTIFICATION OF TEST DATA.

M	2,000	4,000	6,000
WS model	00:06.8	00:07.0	00:06.8
BA model	00:07.2	00:07.0	00:07.3

the test data identified by CNN is equal to the correct results calculated by the simulation experiment.

In this paper, we make the training data and the test data that include results for the WS model with the average degree $k = 4, 6, 8, 12$ and results of the BA model with the average degree $k = 4, 6, 8, 10$. The number M of images for the training data is 2,000, 4,000, and 6,000, which are equally created for each average degree k. Note that 100 images are created from one overlay network because the number N of hosts is 100. As the test data, we prepare different overlay networks from the overlay networks used for training data, and make 200 images for each average degree k. The correct results about the botnet survival probability are calculated from 500 samples obtained in the simulation experiment. The botnet survival probability is classified into $n = 5$ classes A-E as shown in Table II.

B. Results

Table III shows the running time in the simulation experiment for each model. The running time indicates the total time spent obtaining the botnet survival probability for 200 images of test data. Also, Tables IV and V show time consumed for learning of CNN and identification of test data, respectively, against the number M of images for training data. As we can see from these tables, the running time in the simulation experiment is very large even when the average degree k is small. On the other hand, the total time consumed for learning and identification of test data in CNN is small, compared with the running time in the simulation experiment.

Table VI shows the identification accuracy in each model against the number M of images for training data. The identification accuracy is the ratio of the number of test data that are correctly classified to the total number of test data. From this table, we observe that the identification accuracy of the WS model is high for each M. On the other hand, the identification accuracy of the BA model increases with M. The proposed method achieves high identification accuracy that is more than 80% in both models.

We then examine the ratio of the number of test data that are classified into classes adjacent to the correct class to the total number of test data that are not classified correctly. For example, if the correct class is C, adjacent classes are B and D. Table VII shows this ratio against M for each model. As

TABLE VI
IDENTIFICATION ACCURACY.

M	2,000	4,000	6,000
WS model	80% (640/800)	84.6 % (677/800)	80.9% (647/800)
BA model	68.5% (548/800)	76.9% (615/800)	80.6% (645/800)

shown in the table, almost all of the data that are not classified correctly is identified as adjacent classes.

V. CONCLUSION

This paper proposed a prediction method of the infection dynamics of self-evolving botnets. The proposed method predicts the level of infection spreading of self-evolving botnets by using adjacency matrices of hosts as input data to CNN. In this paper, we showed the effectiveness of the proposed method through performance evaluation with trained CNN. The idea of the proposed prediction method can be applied to not only the epidemic model for self-evolving botnets but also general epidemic models.

ACKNOWLEDGMENT

This work was partially supported by the Kansai University Fund for Supporting Young Scholars, 2018.

REFERENCES

[1] Keras, https://github.com/keras-team/keras
[2] A. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp.509–512, 1999.
[3] J. Borello and L. Me, "Code obfuscation techniques for metamorphic viruses," *Journal in Computer Virology*, vol. 4, no. 3, pp. 211–220, 2008.
[4] A. Cani, M. Gaudesi, E. Sanchez, G. Squillero, and A. Tonda, "Towards automated malware creation: code generation and code integration," in *Proc. Symposium on Applied Computing*, Gyeongju, Korea, Mar. 2014.
[5] J. Dean et al., "Large scale distributed deep networks," in *Proc. Neural Information Processing Systems*, Lake Tahoe, NV, Dec. 2012.
[6] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, and B. Shuai, "Recent advances in convolutional neural networks," arXiv preprint arXiv:1512.07108, 2015.
[7] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
[8] T. Kudo, T. Kimura, Y. Inoue, H. Aman, and K. Hirata, "Behavior analysis of self-evolving botnets," in *Proc. the 2016 International Conference on Computer, Information, and Telecommunication Systems (CITS 2016)*, Kunming, China, Jul. 2016.
[9] T. Kudo, T. Kimura, Y. Inoue, H. Aman, and K. Hirata, "Stochastic modeling of self-evolving botnets with vulnerability discovery," *Computer Communications*, vol. 124, pp. 101–110, 2018.
[10] E. Meeds, R. Hendriks, S. Faraby, M. Bruntink, and M. Welling, "MLitB: machine learning in the browser," arXiv:1412.2432.
[11] S. Noreen, S. Murtaza, M. Z. Shafiq, and M. Farooq, "Evolvable Malware," in *Proc. Genetic and Evolutionary Computation Conference*, Montreal, Canada, Jul. 2009.
[12] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proc. ACM SIGCOMM conference on Internet measurement*, Rio de janeiro, Brazil, Oct. 2006.
[13] R. Scandariato, J. Walden, A. Hovsepyan, and W. Joosen, "Predicting vulnerable software components via text mining," *IEEE Transactions on Software Engineering*, vol. 40, no. 10, pp. 993–1006, 2014.

TABLE VII
RATIO OF DATA INCLUDED IN ADJACENT CLASSES.

M	2,000		4,000		6,000	
	class	ratio	class	ratio	class	ratio
WS model	A	100.0% (17/17)	A	100.0% (29/29)	A	100.0% (34/34)
	B	100.0% (51/51)	B	100.0% (26/26)	B	100.0% (34/34)
	C	100.0% (43/43)	C	100.0% (32/32)	C	100.0% (55/55)
	D	100.0% (14/14)	D	100.0% (17/17)	D	100.0% (24/24)
	E	100.0% (35/35)	E	100.0% (19/19)	E	100.0% (6/6)
BA model	A	100.0% (32/32)	A	100.0% (16/16)	A	100.0% (3/3)
	B	98.7% (74/75)	B	100.0% (37/37)	B	100.0% (26/26)
	C	100.0% (42/42)	C	100.0% (30/30)	C	100.0% (37/37)
	D	100.0% (36/36)	D	100.0% (35/35)	D	100.0% (22/22)
	E	98.5% (66/67)	E	100.0% (67/67)	E	100.0% (67/67)

[14] M. Wang, H. Zhou, M. Guo, and Z. Zhang, "A scalable and topology configurable protocol for distributed parameter synchronization," in *Proc. Asia-Pacific Workshop on Systems*, Beijing, China, Jun. 2014.
[15] D. Watts and S. Strogatz, "Collective dynamics of 'small world' networks," *Nature*, vol. 393, pp.440–442, 1998.
[16] F. Yamaguchi, F. Lindner, and K. Rieck, "Vulnerability extrapolation: assisted discovery of vulnerabilities using machine learning," in *Proc. USENIX conference on Offensive Technologies*, San Francisco, CA, Aug. 2011.