International Journal of Software Science and Computational Intelligence
Volume 14 • Issue 1

Machine-Learning-Based White-Hat Worm Launcher in Botnet Defense System

Xiangnan Pan, Yamaguchi University, Japan
Shingo Yamaguchi, Yamaguchi University, Japan*
Taku Kageyama, Yamaguchi University, Japan
Mohd Hafizuddin Bin Kamilin, Yamaguchi University, Japan

ABSTRACT

This article proposes a white-hat worm launcher based on machine learning (ML) adaptable to large-scale IoT network for a botnet defense system (BDS). BDS is a cyber-security system that uses white-hat worms to exterminate malicious botnets. White-hat worms defend an IoT system against malicious bots. The BDS decides the number of white-hat worms, but there is no discussion on the white-hat worms' deployment in IoT networks. Therefore, the authors propose a machine-learning-based launcher to launch the white-hat worms effectively along with a divide and conquer algorithm to deploy the launcher to large-scale IoT networks. Then the authors modeled BDS and the launcher with agent-oriented Petri net and confirmed the effect through the simulation of the PN^2 model. The result showed that the proposed launcher can reduce the number of infected devices by about 30-40%.

KEYWORDS

Botnet Defense System (BDS), Divide and Conquer Algorithm, IoT Network, Machine Learning (ML), Petri Net, White-Hat Worm, Worm Launcher

INTRODUCTION

A spree of huge distributed denial-of-service (DDoS) attacks in late 2016 basically changed the threat landscape in Internet security. Notably, the initial attack on Krebs exceeded volumes of 600 Gbps, halt major Internet service suppliers such as OVH and Dyn (Krebs, 2016; Millman, 2016; Moss, 2016). This destructive attack came from hundreds of thousands of Internet of Things (IoT) devices controlled by a brand new botnet named Mirai. The Mirai botnet is a wake-up call to the industry to better secure Internet of Things (IoT) devices or risk exposing the Internet infrastructure to increasingly disruptive distributed denial-of-service (DDoS) attacks (AsSadhan et al., 2017).

Today, the explosive growth of the Internet of Things market has connected billions of small devices to the Internet. In order to survive in such a rapidly growing market, developers usually ignore security issues and bring vulnerable products to the market. These vulnerabilities can be used for launching large-scale DDoS attacks by Mirai, threatening users' personal information and the company's data assets (Garg & Sharma, 2017; Silva, 2017). With the explosive growth of IoT devices, it is no longer realistic to respond with workforce tactics, and an innovative method is needed to greatly improve defense capabilities.

In this paper, we propose a machine learning-based white-hat worm launcher for Botnet Defense System (BDS). The main contributions of our research are as follows:

DOI: 10.4018/IJSSCI.291713 *Corresponding Author

Copyright © 2022, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

1. A machine-learning based white-hat worm launcher is designed to predict white-hat worms'
2. appropriate positions for BDS.
3. In order to apply the launcher to large-scale IoT network, the divide-and-conquer algorithm is proposed to develop the launcher's scalability.
4. To tackle the correlation problem in divide-and-conquer algorithm, the boundary overlapping method is proposed to develop the launcher's adaptivity.
5. The effect of the proposed launcher is evaluated through the simulation of PN^2 model.

Remaining part of this paper is organized as follows: Section 2 surveys the related work and BDS. Section 3 presents the launcher's methodology. Section 4 presents the simulation results and discussion. Section 5 summarizes our key points and gives future work.

RELATED WORK

Botnet and Mitigation Methods

This section reviews the related work that deal with the mitigation methods of botnets. It also reviews the *Botnet Defense System (BDS)* that uses white-hat worms to protect IoT network from Mirai botnets.

Bezerra et al. (2019) proposed a host-based method for detecting IoT botnets and implemented a system called IoTDS. IoTDS monitors the CPU usage rate and memory consumption of the device, when it detects a botnet from those data, it sends the alarm to the central server. Nagisetty and Gupta (2019) proposed a framework for detection of malicious activities in IoT Backbone Networks using Keras Deep learning Library. The proposed framework uses four different deep learning model such as Multi Layer Perceptron (MLP), Convolutional Neural Networks (CNN), Deep Neural Networks (DNN) and Autoencoder for predicting the malicious attacks. Alieyan et al. (2021) proposed a DNS rule-based approach for Botnet Detection (DNS-BD) that can detect any abnormal DNS query and response behaviors. The approach has a higher accuracy than any other approaches based on DNS. Kumar et al. (2021) proposed an ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks, and the evaluation showed the detection framework can reach a high accuracy.

Gopal et al. (2018) proposed a whitelisting based solution to prevent the spread of Mirai and showed the successful blocking of Mirai malware through experiments. Manso et al. (2019) proposed a software-defined intrusion detection system. When this system detects an attack, it uses a software-defined network controller to control the traffic. This reduces abnormal traffic and makes it possible to maintain normal traffic. Ceron et al. (2019) proposed an approach for handling the network traffic generated by the IoT malware. The approach can characterize the malware behaviors and modify the traffic at the network layer to limit the malicious activity.

We summarized the attributes and limitations of the related work in Table 1. However, we can see that the previous studies mainly focused on methods to detect botnets or to mitigate the threats of botnets, but they cannot exterminate existing botnets. In other words, the purpose of the previous studies is to detect botnet or defend against botnet, while our proposed launcher in BDS is actually an offensive way to protect our IoT network.

Botnet Defense System

Research has shown (Yamaguchi, 2020a) that a promising method is to use a kind of IoT worms called Hajime as white-hat worms against Mirai. Yamaguchi (2020b) has proposed a new kind of cyber-security system, named Botnet Defense System (BDS). BDS uses white-hat botnets to exterminate malicious botnets like Mirai. A BDS consists of four components: *monitor*, *strategy planner*, *launcher*, and *command&control server*, they are respectively responsible for detecting botnet, making strategy, launching white-hat worms, and commanding white-hat worms. He evaluated BDS and the proposed

Table 1. The Comparative table of the related work

Authors	Attributes	Limitations
Bezerra et al., 2019	Monitoring CPU usage and memory consumption to detect botnets	Unable to handle concept drifts automatically
Nagisetty & Gupta, 2019	Detecting malicious activity based on network traffic and Machine Learning Library	Only four different machine learning algorithms were evaluated
Alieyan et al., 2021	Detecting any abnormal DNS query and response behaviors	Unable to apply in other type of botnet
Kumar et al., 2021	Using ensemble learning IDS based on fog-cloud architecture to detect malicious activity	Other mechanisms like the verified boot, remote attestation have same effect
Gopal et al., 2018	Preventing the spread of Mirai by whitelisting	Other mechanisms like the verified boot, remote attestation have same effect
Manso et al., 2019	Detecting botnet and controlling the traffic by SDN-based IDS	Scalability and adaptability
Ceron et al., 2019	Identifying malware behaviors at the network layer to limit the malicious activity	The test malware is executed in the sandbox
This Work	Launching white-hat worms by machine learning to exterminate the botnets	An efficient divide method need to be considered

strategies through the simulation of agent-oriented Petri net model. This is a promising way to exterminate malicious botnets.

However, the BDS only decides the number of white-hat worms but dose not decide where those white-hat worms should be launched. In other words, the previous approach only launch white-hat worms in a random manner. Thus, the authors propose a machine-learning-based white-hat worm launcher for *Botnet Defense System (BDS),* which can launch the white-hat worms for fighting in an effective manner.

MACHINE-LEARNING-BASED WHITE-HAT WORM LAUNCHER

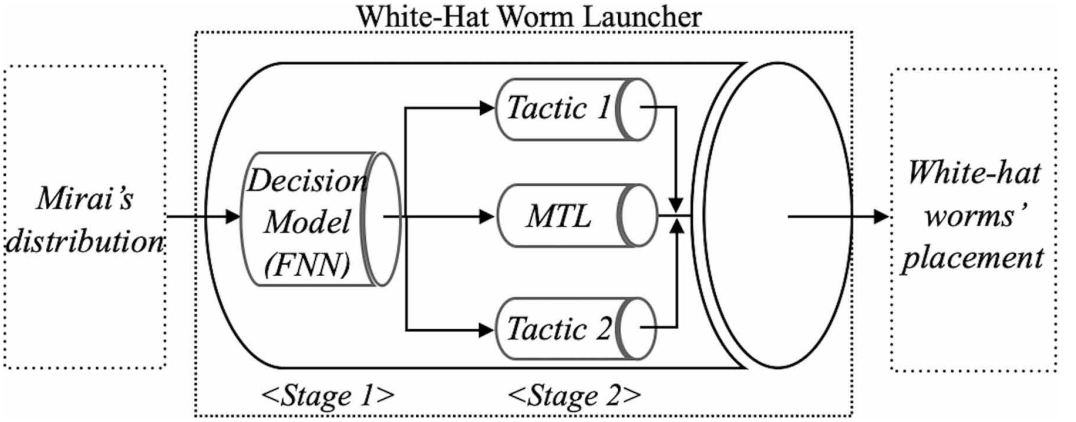
An Overview of the Proposed Launcher

We present a structure of the white-hat worm launcher in Figure 1. The launcher consists of two stages: The Stage 1 utilizes a Feed-forward Neural Network (FNN) to decide an appropriate placement tactic. The Stage 2 utilizes a Multi-Task Learning Neural Network (MTL) to predict white-hat worms' appropriate position for fighting against the bots. This launcher can learn where is the right place (appropriate tactic) to deploy the white-hat worms for fighting malicious botnets (Pan et al., 2021).

Divide-And-Conquer Algorithm

We found from our preliminary experiment that the performance decreases with the size of network (e.g. the number of nodes is 100). In addition, a large-scale network usually means more time and more expensive hardware are required to train learning model and analyze IoT network. This will undoubtedly increase economic and time cost especially for small organizations when using the white-hat worm launcher. Thus, we need to improve to make the launcher adaptive. In this section,

Figure 1. The overview of the white-hat worm launcher



we proposed a divide-and-conquer algorithm into the white-hat worm launcher to adapt to large-scale IoT network.

As a preparation, we train a decision model M to predict an appropriate tactic for a small network like 5x5 nodes (corresponding to each divided sub-network).

Input: Mirai's distribution D_M on network N, Machine learning models M_1 and M_2.
Output: White-hat worm's deployment D_W for D_M.
Step 1: We divide N into n sub-networks N_1, N_2, … N_n
Step 2: For each sub-network N_i in N_1, N_2, … N_n
Step 2-1: Predict a tactic T_i for N_i with M_1
Step 2-2: Obtain the white-hat worms' deployment D_{Wi} by using M_2
Step 3: Output the white-hat worm's whole deployment D_W by combining D_{W1}, D_{W2}, … D_{Wn}

We utilize an example to illustrate our proposed algorithm as shown in Figure 2. In step 1, we get the Mirai's distribution on a 10×10 IoT network and divide the network into 4 sub-networks. Each sub-network is a 5×5 network. In steps 2-1, we predict a tactic for each 5×5 sub-network by using the FNN learning model. In steps 2-2, we obtain the white-hat worms' deployment for each 5×5 sub-network by using the MTL learning model. In step 3, we combine all sub-networks to output the white-hat worm's whole deployment.

Deployment Tactics

Kageyama and Yamaguchi (2021) focused on the distribution patterns of devices infected by Mirai malware and proposed two placement tactics for disinfecting Mirai efficiently. We apply these two tactics to the proposed launcher for better deployment:

Tactic One (Uniform Deployment): Put white-hat worms evenly.
Tactic Two (Clumped Deployment): Put white-hat worms around Mirai bots.

Through the two tactics, white-hat worms can be launched in a targeted manner to improve the effectiveness on devices infected with fixed patterns shown as Figure 3; When Mirai's distribution is uniform, white-hat worms will be launched evenly; when Mirai's distribution is clumped, white hat worms will be launched around clumped Mirai bots to limit further infection.

Figure 2. The illustration of the divide-and-conquer algorithm

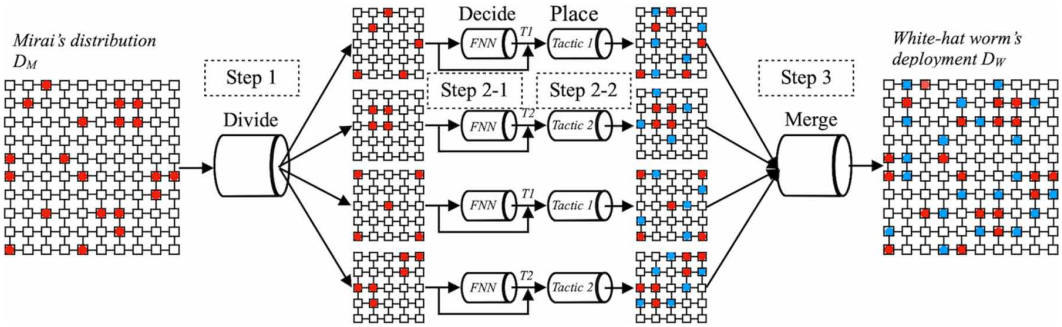
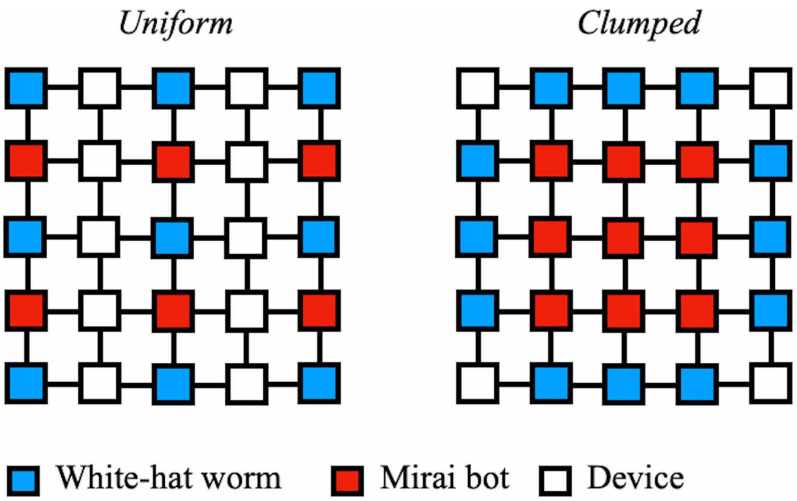


Figure 3. The illustration of the proposed tactics



Tactics Selection

The Stage 1 utilizes a Feed-forward Neural Network (FNN) as shown in Figure 4. The network architecture used to predict an appropriate tactic for white-hat worm launcher. The input layer corresponds to i units (= i devices), and there are 2 hidden layers, each of the layers has the number of units of i^2, i. All the layers use a rectified linear unit (ReLU) activation function except for the last layer which connected to the 3 output units with Softmax activation function. Each output unit represents a different class of tactics.

Worm Placement

The Stage 2 utilizes a Multi-Task Learning neural network as shown in Figure 5 to predict white-hat worms' appropriate position for fighting against the Mirai bots. Compared to the machine learning model shown in Figure 4, this model utilizes the Sigmoid activation function for the final layer in the task-specific layers.

Figure 4. The structure of the machine learning model M1 for Stage 1

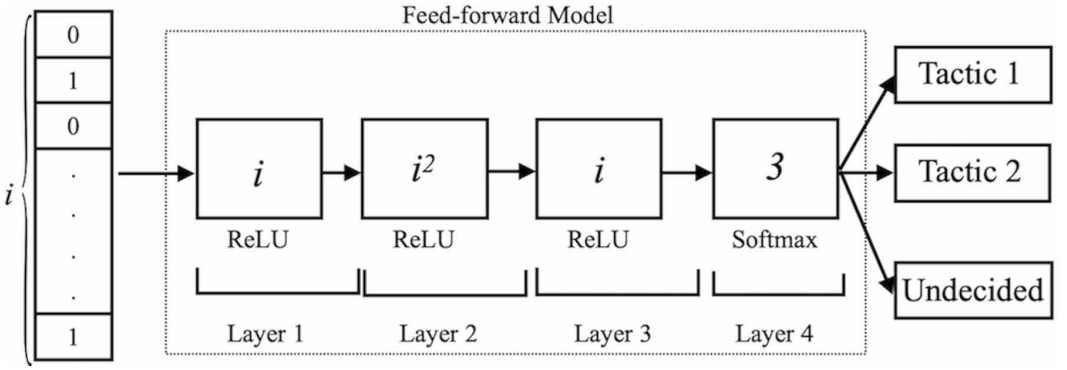
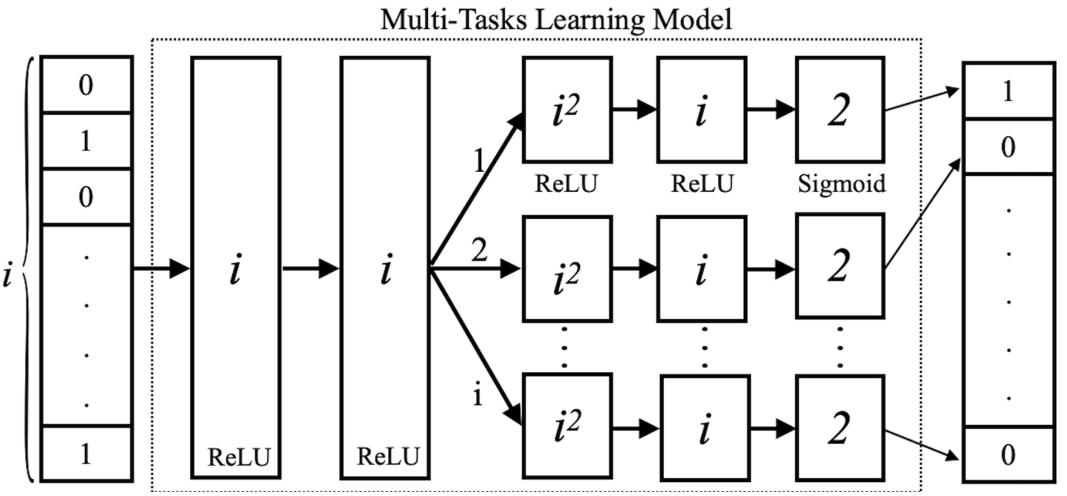


Figure 5. The structure of the machine learning model M2 for Stage 2



SIMULATION EVALUATION

Simulation

We confirm the effectiveness of the machine learning through the simulation of the PN^2 model shown in Figure 6. In this experiment, we used a grid structured IoT network composed of 25 (=5×5) IoT devices, where 5 devices are infected with the Mirai botnet. To exterminate the Mirai botnet, 5 white-hat worms will be deployed by the proposed white-hat worm launcher.

The infection rate of performance is indicated in Table 2. This shows the simulation result in different initial distribution patterns of Mirai bots. Each value shows the Mirai's infection rate of 1,000 steps for 1,000 trials.

Next, we conducted a simulation experiment with different Mirai's distribution patterns to evaluate the proposed algorithm with PN2Simulator (Nakahori & Yamaguchi, 2017). In this experiment, we used a grid structured IoT system composed of 100 IoT devices, where 20 devices are infected with the Mirai botnet. In order to exterminate the bots, 20 white-hat worms will be launched to fight against Mirai bots. We simulate 3 Mirai worms' infection distribution patterns to confirm the divide-and-

Figure 6. The PN² Model representing a battle between Mirai and white-hat worm

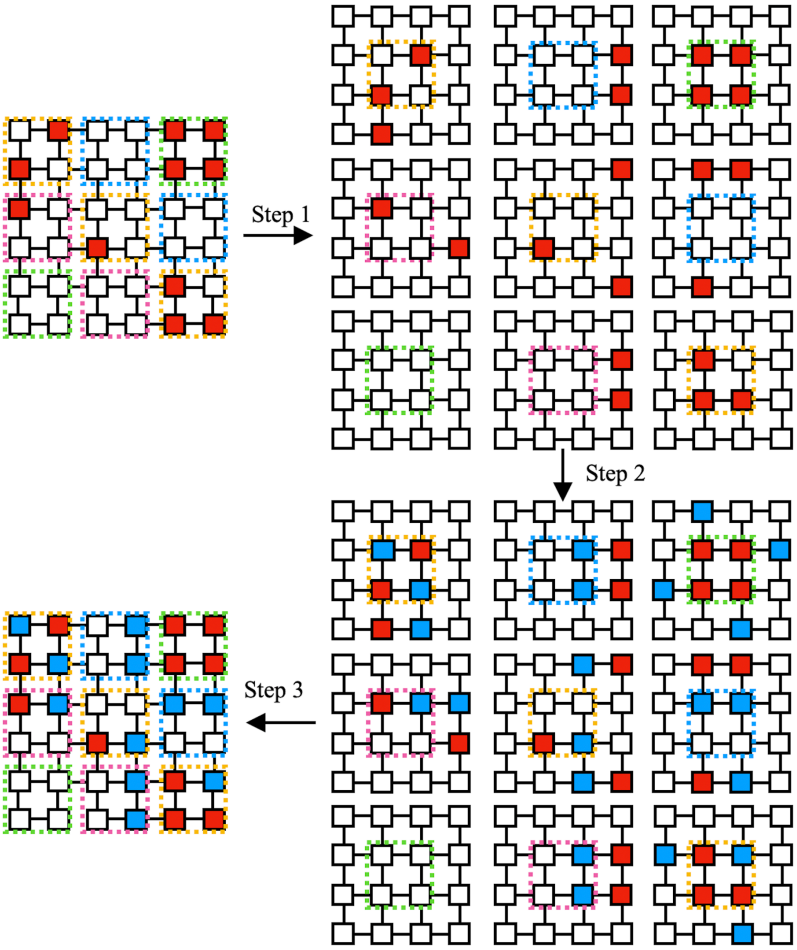


Table 2. Mirai infection rate with machine-learning placement compared to random placement

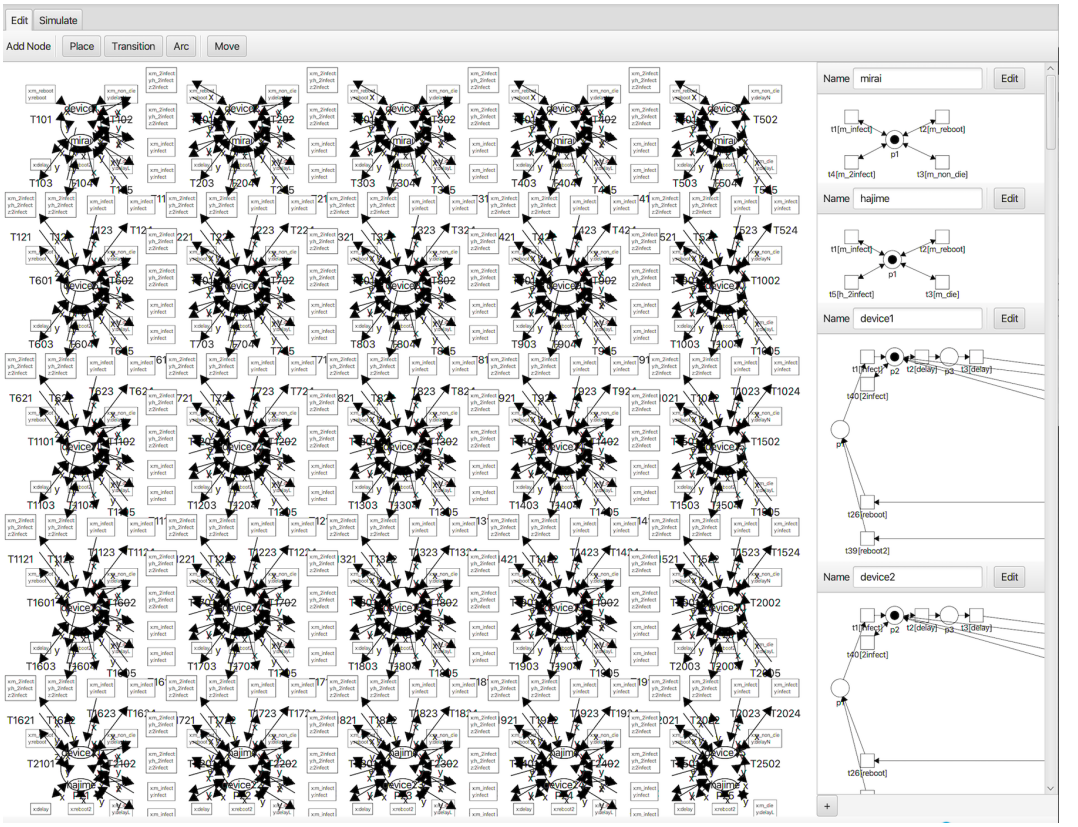
	Without ML [%]	With ML [%]
Random	20.1	15.2
Uniform	24.6	20.4
Clumped	16.7	7.6

conquer algorithm, they are random pattern, uniform pattern, and clumped pattern. Their distributions are shown respectively in Figure 7 to Figure 9.

The simulation result is shown in Table 3 and Table 4. Table 3 shows the result when using white-hat worms of delay time δ=7 lifespan ℓ=2 and secondary infectivity ρ=50%. Table III shows the result when using white-hat worms of delay time δ=11 lifespan ℓ=3 and secondary infectivity ρ=50%. Each cell shows the mean Mirai infection rate for 1,000 trials.

By referring to the results in Table 3 to Table 4, we can see that no matter which distribution pattern the Mirai belongs to, the infection rate can be decreased 30~*40%* when using the proposed

Figure 7. Launch white-hat worms with divide-and-conquer algorithm when Mirai's random distribution pattern: sub-network I with tactic 1, sub-network II with tactic 2, sub-network III with tactic 1, sub-network IV with tactic 2



algorithm, which shows that our algorithm is effective. Notably, our proposed launcher can perform a more obvious effect when the white-hat worm's capability is relatively weak.

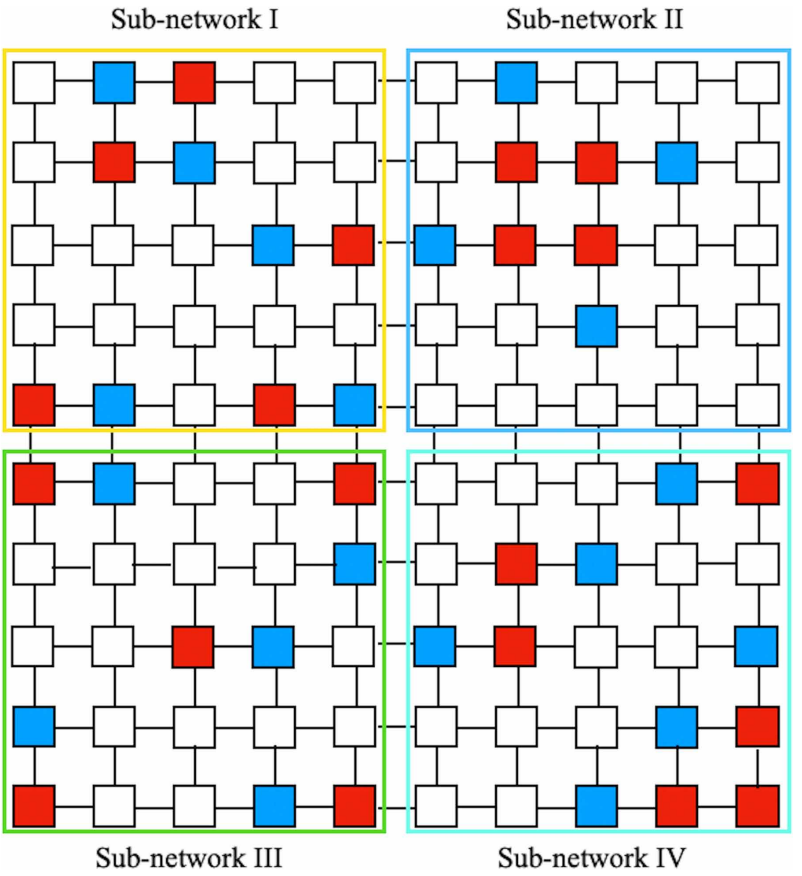
Discussion

First, let us discuss the effect of machine learning. Table 2 shows the simulation result using machine learning in a 5×5 network, the infection rate can all be reduced obviously in different initial infection patterns. That means machine learning is an available method which can predict the appropriate placement for white-hat worms. Next, let us discuss the effect of divide-and-conquer algorithm. Table 3 to Table 4 show the simulation result using divide-and-conquer algorithm in a 10×10 network, the number of infected devices can be reduced by about 30~40% in different initial infection patterns. This shows that it is available to apply machine learning to large-scale networks through the divide and conquer algorithm.

In addition, the divide-and-conquer algorithm may not be very effective in some specific infection patterns. Because some sub-networks may all be infected with Mirai botnets, there is no room in such a sub-network to launch any white-hat worm. In order to avoid this situation, we have developed a division method called boundary overlapping to strengthen the correlation between the divided sub-networks.

Step 1: We extend each sub-network by one layer of boundary to obtain each shared sub-network $N_1, N_2, \ldots N_n$

Figure 8. Launch white-hat worms with divide-and-conquer algorithm when Mirai's uniform distribution pattern: sub-network I with tactic 1, sub-network II with tactic 1, sub-network III with tactic 1, sub-network IV with tactic 1



Step 2: For each sub-network N_i in N_1, N_2, … N_n
Step 2-1: Predict a tactic T_i for N_i with M_1
Step 2-2: Obtain the white-hat worms' deployment D_{Wi} by using M_2
Step 3: Output the white-hat worm's whole deployment D_W by combining D_{W1}, D_{W2}, … D_{Wn} without overlapped boundaries.

We use a 6x6 network to illustrate the process of this developed method as shown in Figure 10. We first select a 2×2 sub-network to divide the network into 9 sub-networks, and then adopt the boundary overlapping method to transform into 9 overlapped sub-networks, respectively predict a tactic for these nine sub-networks and launch white hat worms, and output the white-hat worm's whole deployment by combining these 9 sub-networks (ignore the placement on the overlapped boundary).

We conducted a simulation experiment to confirm the effect of boundary overlapping in a 6×6 network shown in Figure 11 where 11 devices are infected by Mirai. These 11 infected devices are distributed in a specific way (there are some divided sub-networks where all devices are infected by Mirai). In order to exterminate Mirai bots, 11 white-hat worms of delay time δ=7 lifespan ℓ=2 and secondary infectivity ρ=50% will be deployed using boundary overlapping method. The result in Table 4 shows that the boundary overlapping method can further reduce the infection rate compared to the divide-and conquer algorithm.

Figure 9. Launch white-hat worms with divide-and-conquer algorithm when Mirai's clumped distribution pattern: sub-network I with tactic 2, sub-network II with tactic 2, sub-network III with tactic 2, sub-network IV with tactic 2

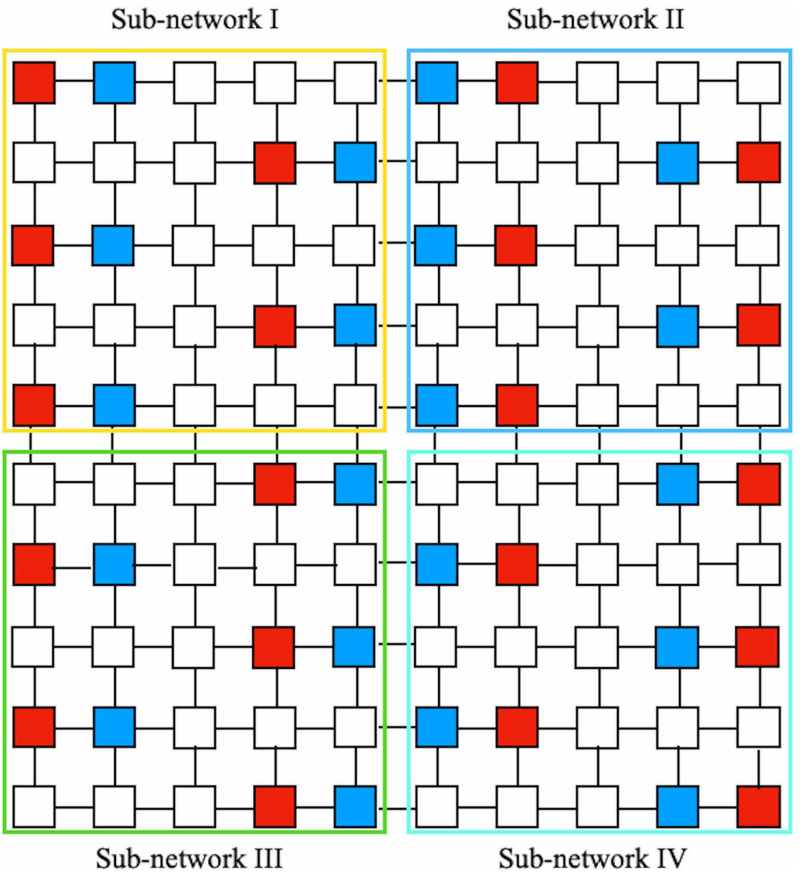


Table 3. Mirai infection rate with machine-learning placement compared to random placement when delay time δ=7 lifespan ℓ=2 and secondary infectivity ρ=50%.

	Without ML [%]	With ML [%]
Random	1.2	0.7
Uniform	0.7	0.5
Clumped	4.8	3.0

Table 4. Mirai infection rate with machine-learning placement compared to random placement when delay time δ=11 lifespan ℓ=3 and secondary infectivity ρ=50%.

	Without ML [%]	With ML [%]
Random	6.9	4.0
Uniform	5.9	3.8
Clumped	46.6	32.0

Figure 10. The illustration of the boundary overlapping

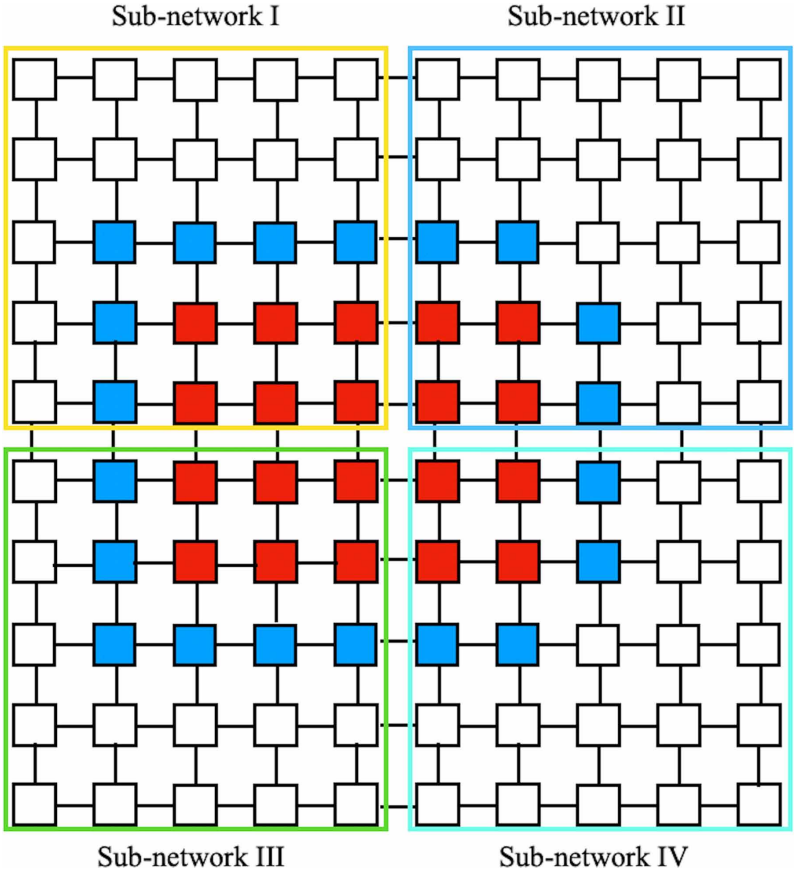
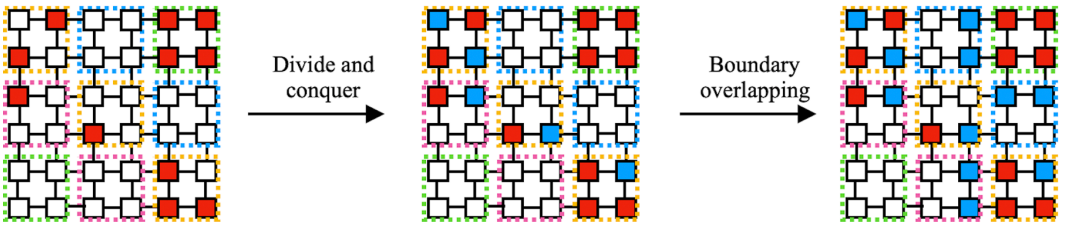


Figure 11. Launch white-hat worms with divide-and-conquer algorithm and boundary overlapping method



In summary, we have involved machine learning, divide and conquer algorithm, and boundary overlapping method. Each method is based on the former and has an independent applicable environment. Table 5 shows the limitations and attributes of them.

Table 5. Mirai infection rate with boundary overlapping method when delay time δ=7 lifespan ℓ=2 and secondary infectivity ρ=50%

Infection Pattern	Without BO [%]	With BO [%]
Specific	24.2%	11.8%

CONCLUSION

In this paper, we proposed a white-hat worm launcher based on machine learning and a divide-and-conquer algorithm. First, We gave an design of the launcher along with two effective deployment tactics into the launcher for better placement. In order to confirm the effect of the proposed launcher, we modeled BDS with PN^2 and evaluated the launcher through the simulation in a 5×5 grid IoT network. Then we also confirm the proposed launcher's scalibility in a 10×10 network. The simulation result shows that the proposed launcher has an effect in eliminating Mirai on a large-scale IoT network. In order to strengthen the correlation between the divided sub-networks for better adaptivity, we developed a division method called boundary overlapping and confirmed its effect.

Future work should aim at developing sub-network dividing methodology for different infection types of IoT networks. For different infection patterns, we can take effective division methods to divide and conquer them.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP19K11965.

REFERENCES

AsSadhan, B., Bashaiwth, A., Al-Muhtadi, J., & Alshebeili, S. (2017). Analysis of P2P, IRC and HTTP Traffic for Botnets Detection. *Peer-to-Peer Networking and Applications*, *11*, 1–14.

Alieyan, K., Almomani, A., Anbar, M., Alauthman, M., Abdullah, R., & Gupta, B. B. (2021). DNS rule-based schema to botnet detection. *Enterprise Information Systems*, *15*(4), 545–564. doi:10.1080/17517575.2019.1644673

Bezerra, V. H., da Costa, V. G. T., Barbon, J., Miani, R. S., & Zarpelão, B. B. (2019). A One-Class Classification Approach to Detect Botnets in Internet of Things Devices. *Sensors (Basel)*, *19*(14), 3188. doi:10.3390/s19143188 PMID:31331071

Ceron, J. M., Steding-Jessen, K., Hoepers, C., Granville, L. Z., & Margi, C. B. (2019). Improving IoT Botnet Investigation Using an Adaptive Network Layer. *Sensors (Basel)*, *19*, 727.

Silva, K. K. (2017). How Industry Can Help Us Fight against Botnets: Notes on Regulating Private-Sector Intervention. *International Review of Law Computers & Technology*, *31*, 105–130.

Gopal, T. S., Meerolla, M., Jyostna, G., Eswari, P. R. L., & Magesh, E. (2018). Mitigating Mirai Malware Spreading in IoT Environment. *Proc. of ICACCI 2018*, 2226–2230. doi:10.1109/ICACCI.2018.8554643

Garg, S., & Sharma, R. (2017). Classification Based Network Layer Botnet Detection. *Advanced Informatics for Computing Research*, 332–342.

Krebs, B. (2016). *Krebs On Security Hit With Record DDoS*. https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-recordddos/

Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *IEEE Computer*, *50*(7), 84–87. doi:10.1109/MC.2017.201

Kumar, P., Gupta, G. P., & Tripathi, R. (2021). An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Computer Communications*, *166*, 110–124. doi:10.1016/j.comcom.2020.12.003

Kageyama, T., & Yamaguchi, S. (2021). On Tactics to Deploy White-Hat Worms in Botnet Defense System. *Proc. of IEEE GCCE 2021*, 320-323.

Liu, X., Du, X., Zhang, X., Zhu, Q., Wang, H., & Guizani, M. (2019). Adversarial Samples on Android Malware Detection Systems for IoT Systems. *Sensors (Basel)*, *19*(4), 974. doi:10.3390/s19040974 PMID:30823597

Millman, R. (2016). *OVH suffers 1.1 Tbps DDoS attack*. https://www.scmagazine.com/news/ddos/ovh-suffers-massive-1-1tbps-ddos-attack

Moss, S. (2016). *Major DDoS attack on Dyn disrupts AWS, Twitter, Spotify and more*. https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

Moffitt, T., & Tapia, A. (2016). *Source Code for Mirai IoT Malware Released*. https:// www.webroot.com/blog/2016/10/10/source-code-Mirai-iot-malware-re-leased/

Manso, P., Moura, J., & Serrão, C. (2019). SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS At-tacks. *Information (Basel)*, *10*(3), 106. doi:10.3390/info10030106

Nakahori, K., & Yamaguchi, S. (2017). A support tool to design IoT services with NuSMV. *Proc. of IEEE ICCE 2017*, 84-87. doi:10.1109/ICCE.2017.7889238

Nagisetty, A., & Gupta, G. P. (2019). Framework for detection of malicious activities in iot networks using keras deep learning library. *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 633-637.

Pan, X., Yamaguchi, S., & Kageyama, T. (2021). Machine-Learning-Based White-Hat Worm Launcher Adaptable to Large-Scale IoT Network. *Proc. of IEEE GCCE 2021*, 309-312.

US Computer Emergency Readiness Team. (2016). *Heightened DDoS threat posed by Mirai and other botnets*. https://www.us-cert.gov/ncas/alerts/TA16-288A

Yamaguchi, S. (2020). White-Hat Worm to Fight Malware and Its Evaluation by Agent-Oriented Petri Nets. *Sensors (Basel)*, *20*(2), 556. doi:10.3390/s20020556 PMID:31963954

Yamaguchi, S. (2020). Botnet Defense System: Concept, Design, and Basic Strategy. *Information, 11*, 516.

Xiangnan Pan received the B.E. degree from Xihua University, China, in 2016. He is currently a graduate student at Yamaguchi University, Japan. His research interest includes IoT and cybersecurity.

Shingo Yamaguchi is a Professor in the Graduate School of Sciences and Technology for Innovation Yamaguchi University, Japan. He received the B.E., M.E., and D.E. degrees from Yamaguchi University, Japan, in 1992, 1994, and 2002. He was a Visiting Scholar at the University of Illinois at Chicago, US, in 2007. He is currently the Director of Information and Data Science Education Center, Yamaguchi University. His research interests include AI, IoT, big data, and cybersecurity. He was the Executive Conference Chair of IEEE ICCE 2021. He is a Member of the Board of Governors of IEEE Consumer Technology Society. He is also the Editor-in-Chief of IEEE Consumer Electronics Magazine. He is a Senior Member of IEEE and IEICE.

Taku Kageyama received the B.E. degree from Yamaguchi University, Japan, in 2021. He is currently a graduate student at Yamaguchi University. His research interest includes IoT and cybersecurity.

Mohd Hafizuddin Bin Kamilin received the B.E. from Tokyo University of Science, Yamaguchi, Japan, in 2019. He received the M.E degree from Yamaguchi University, Japan, in 2021. His research interest is artificial intelligence application in IoT scheduling optimization and formulating a plan to exterminate the malicious botnet from the IoT devices.