Malicious Packet Detection Technology Using Reinforcement Learning

ByungWook An[1], JoongChan Lee[2], JaiSung Choi[3], Wonhyung Park[4]

[1]NCsoft, Korea

[2]Department of Artificial Intelligence Convergence, Hanyang University, Korea

[3]Security Group Moby Dick, Korea

[4]Department of Convergence Security at Sungshin Women's University, Korea

Received: December 24, 2023; Revised: February 24, 2024; Accepted: March 10, 2024; Published: March 25, 2024

Abstract

In the present day, the advancement of 5G and Internet of Things (IoT) technology has resulted in the interconnectedness of everyday objects through networks. However, attempts to exploit networked computers for malicious purposes continue to rise while the attacks utilizing malicious codes to compromise user information's confidentiality and integrity are becoming increasingly sophisticated and intelligent. To counter these evolving threats, researches have been conducted on a method to identify malicious network packets using a combination of security control systems and Artificial Intelligence (AI) technology, especially supervised learning. Unfortunately, the current cybersecurity control systems suffer from inefficiencies in terms of both manpower and cost. Moreover, the surge in remote work has created challenges in responding swiftly to security incidents. Furthermore, the existing AI technology based on supervised learning has limitations, particularly in detecting new variants of malicious code, and its accuracy in identifying malicious code depends heavily on the quantity and quality of available data. In light of these challenges, this study, reinforcement learning is employed to overcome the limitations of the original supervised learning-based malicious packet detection system, such as high dependency on training data and the failure to detect variant malicious packets. This research proposes a malicious packet detection technology capable of addressing new malicious packets or variant types effectively.

Keywords: Machine Learning, Deep Learning, Reinforcement learning, Guidance Learning, Malicious Code Detection System.

1 Introduction

With the advancement of IT technology, the increasing use of the internet, and the proliferation of key technologies such as Internet of Things (IoT), 5G, and Artificial Intelligence (AI) in the context of the Fourth Industrial Revolution, the boundaries between cyberspace and the physical world are rapidly converging [1].

According to the annual Internet Report by the U.S. telecommunications equipment company Cisco, the worldwide number of internet users is projected to increase from 51% in 2018 to 66% in 2023 [2]. With the growth in the number of internet users, it is also anticipated that cyber security incidents will increase. According to data from Cybersecurity Ventures, as of 2021, the scale of cyber damages amounts to $6.39 trillion, and it is expected to significantly increase to $10.5 trillion by 2025 [3]. Among cyberattacks, those that compromise the confidentiality and integrity of user information using malicious packets are a major concern due to their increasing sophistication and diverse tactics.

Efforts to combat security threats posed by malicious packets have led to the implementation of 24-hour intrusion detection and monitoring systems. However, these systems demand constant human oversight, which makes them inefficient in terms of both human resources and costs. Moreover, they encounter difficulties in delivering timely responses, especially in the context of remote work scenarios. To solve this problem, AI-based malicious packet detection research is actively underway to reduce the workload of controllers by 30-50% and solve missing problems that may occur during manual analysis and response. However, the malicious packet detection method using supervised learning, an existing AI technology, has an error in which the amount and quality of data have a significant impact on the result value. For example, there are propensity errors that show inaccurate detection with insufficient data learning and overfitting errors that show result values that vary sensitively to minor values due to excessive use of learning data [4]. Even worse, the supervised learning-based approach, which relies on learning known attacks in advance, faces a significant limitation. In more detail, it is ill-suited for the real world, where new and variant network packets are constantly emerging at a rapid pace. In order to effectively improve this problem, a method for detecting malicious packets through reinforcement learning among machine learning algorithms is proposed. The paper is organized as follows. Section 2 introduces the background of this study. Section 3 summarizes the implementation method, followed by the experimental process in Section 4. Finally, Section 5 gives the conclusions and presents future research directions.

2 Literature review

2.1 Advances in technology and increased security threats

With the development of 5G and IoT technologies, large and small computers are mounted on objects used in real life, and these objects are connected by networks. Since each computer is connected by a network, attempts to use it for malicious purposes are increasing. As a case in point, according to the Financial Supervisory Service, the international hacker group "Armada Collective" sent a threatening letter to a Korean bank demanding Bitcoin, but when it did not respond, it carried out a DDoS attack [5]. Also, according to Sonicwall, a U.S. network security firm, global attempts to attack IoT devices increased by about 59% from 20.2 million in January-June 2020 to 32.2 million in January-June 2021. [6]. In order to protect the system from these cybersecurity threats, research is being conducted on malicious packet detection methods using supervised learning, an AI technology. In addition, although a security control system for detecting malicious packets is in operation, there are cost and operational limitations in areas where humans have to monitor them 24 hours a day. Accordingly, there is a need to prepare a plan to detect malicious packets based on network traffic using reinforcement learning.

2.2 Malicious packet detection using supervised learning

Supervised learning is a method of machine learning to infer a function from the trained data. The training data includes a label, that is, a correct answer sheet, and uses it to infer the value. Models of supervised learning include classification and regression. The classification predicts one of several predefined or possible class labels. Regression outputs a continuous value among the inferred functions. This method is highly accurate because a person intervenes in the target value. However, there is a disadvantage that the amount of learning data is large and it takes a lot of time to label. In addition, supervised learning methods are more difficult to create training data than to create models.

2.3 Malicious Packet Detection Using Unsupervised Learning

Unsupervised learning learns without output information corresponding to a given input. That is, it is a

method used to find patterns or classify data without any information that can classify data. This method has the advantage of being very fast because it does not set a target value. However, it is more difficult than supervised learning because patterns or forms must be found from unlabeled data. In addition, the lack of labels makes it difficult to evaluate the training results of the model. Therefore, in order to compensate for the problems and limitations of supervised and unsupervised learning, this study proposes a malicious packet detection method based on network traffic using reinforcement learning, another method of machine learning.

3 Implementation

3.1 Implementation Environment

In this paper, the system configuration diagram as shown in (Figure 1) is followed.

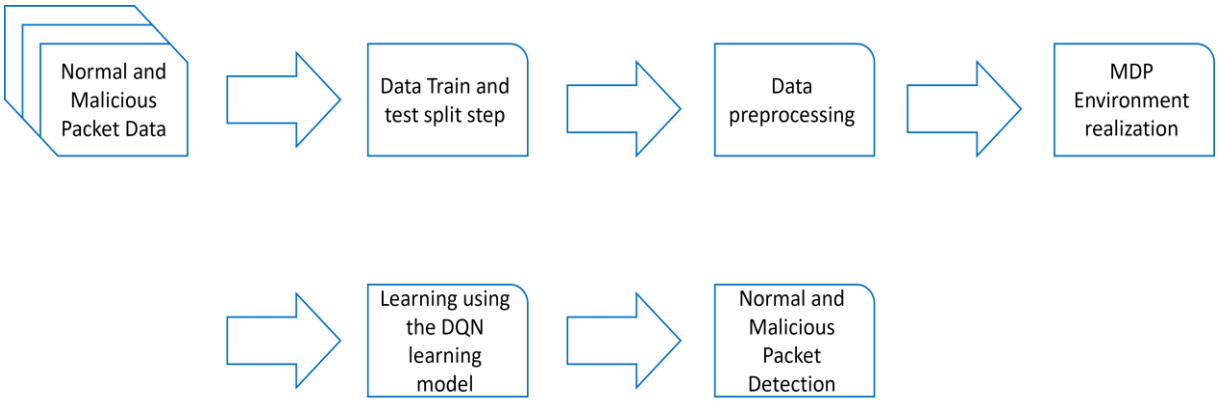

Figure 1: System configuration diagram

As an experimental environment, it was configured as shown in Table 1 to analyze network packet data. To analyze a lot of data, data preprocessing is performed at high speed using NVDIA GPUs, Python 3.9 of the Jupiter Notebook for artificial intelligence models, and Google Korab for data visualization.

Table 1: Analysis Environment

Classification	Subclassification	Content
Analysis Environment	-	raw network Packet information
Data Processing	Pre-processing	NVDIA GPU High-Speed Pre-processing
Data Analysis	Analysis Tools	Jupyter Notebook
Data Analysis	Analysis Tools	python 3.9
Visualization	-	Google Korab Graph

3.2 Train and Test Split

The learning data was divided into X lists with the features of the dataset as columns, and Label divided into Y lists. After that, the machine learning model called Overfitting is trained only with Train data, and the existing data set is divided into a certain ratio (7:3) to prevent performance than expected when the model is applied to test data (Figure 2), and then the intermediate validation data is used to verify the learning process.

Figure 2: Train and Test split results

If the prediction rate or error rate decreases during the verification process, this means that the model has been overfitted, so the learning is terminated immediately. After that, learning is carried out in a way that is not overfit.

3.3 Data preprocessing

In order to convert the previous data set into a form that the model can understand and improve the quality of the data, the data preprocessing process, which is the process of removing data realization, incomplete data, and data noise, and solving data imbalance (overcapture, overcapture). In addition, the preprocessed data was prevented from misaligning in advance using Pandas' Data Frame and made it easier for machine learning models to access data values.

3.4 Environment - Markov Decision Process (MDP)

As shown in (Figure 3), in the Markov decision process, the agent (malicious packet detection system) and the environment interact.

First, the malicious packet detection system receives a state from the environment and selects an action based on this state and experience. Based on the behavior chosen by the agent in this way, the environment gives a new state and also rewards based on this behavior. In this way, an agent going from the current state to the next state is called a state transition [7].

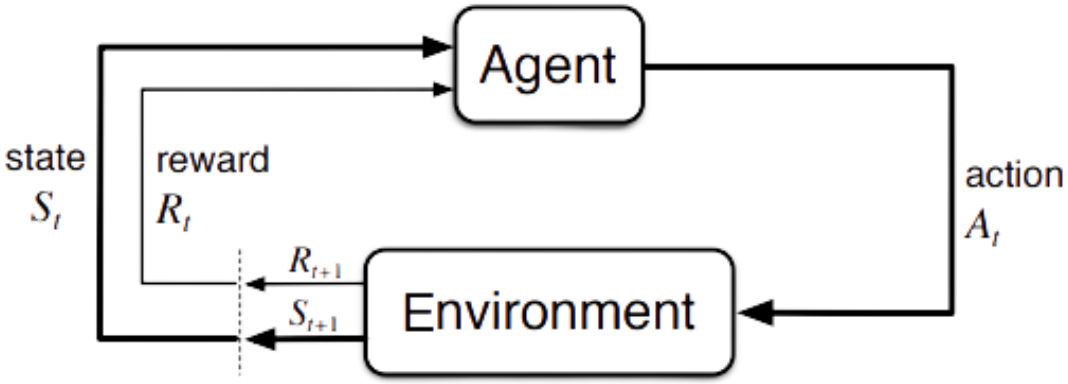

Figure 3: Markov Decision Process

By repeating the same process as (Figure 3), agents and environments create tuples of states and behaviors, assuming that some states contain information about all previous states, which is called the Markov attribute. By this Markov attribute, all states can be predicted in the MDP.

3.5 Deep Q-Network (DQN) learning model

Deep Q-Network (DQN) is a deep queue network that combines multi-layer neural networks and queue learning. The DQN is divided into an input layer that handles information extraction as input data, a hidden layer that handles data classification, and an output layer that outputs data. CNN (Convolution Neural Network) was used to classify malicious packets in Q-network.

CNNs are constructed as shown in (Figure 4) and undergo a convolution process between the input layer and the hidden layer to extract features with eight data obtained through preprocessing with CNN's input [8]. A convolution extracts a feature using a composite product of values. When features are extracted, they pass through the activation function ReLU (Rectified Linear Unit) and generate 1 if there is a feature of malicious packets through backpropagation that activates neurons and adjusts the weight to the hidden layer, and 0 if not. Since not all features are needed in the feature data, it goes through a polling process that reduces the size to make it into an appropriate amount of data. It detects malicious packets with softmax functions in the hidden layer with values 0 and 1 received from the ReLU function. The output layer is a probability value derived from the softmax function and is reinforced with a malicious packet detection rate (compensation).

In addition, when reinforcement learning from Q-network to DQN, optimizers should be made to derive accurate values as the layer becomes deeper and deeper. We use a loss function that evaluates whether it matches the train data separated by optimization [9]. We use the Mean Square Error (MSE) formula to reduce the error between the predictive probability of malicious packet detection and the probability of the actual result.

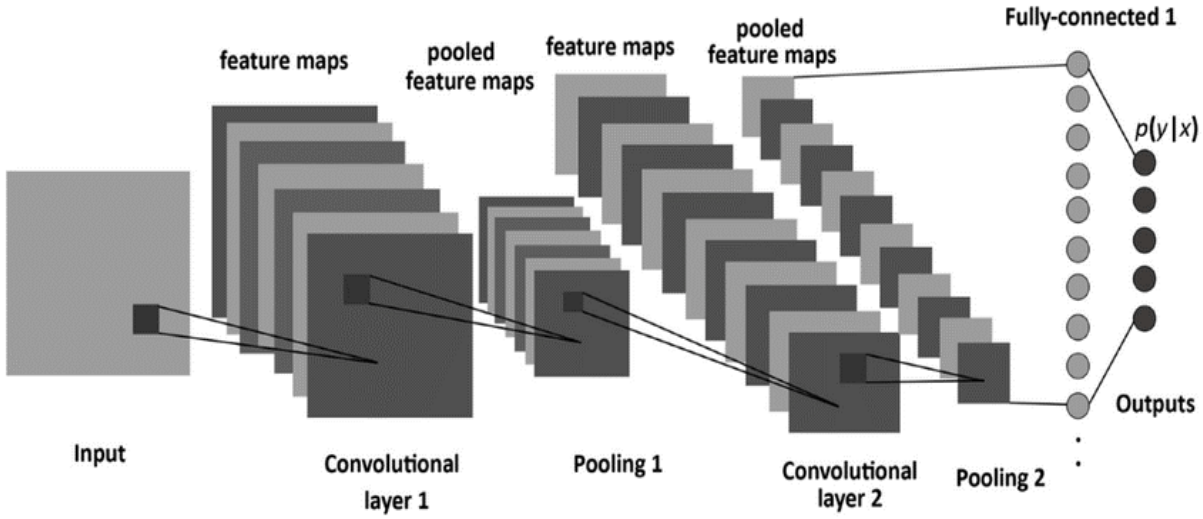

Figure 4: CNN Configuration

3.6 Q-learning learning model

Reinforcement learning is repeatedly trained with Deep Q-Learning to proceed with malicious packet detection and finish when the final step of detecting malicious packets is reached. In order to eliminate the temporal correlation, reinforcement learning is conducted with a mini batch process (Mini Batch) in which replay memory is generated as a recording variable for update and data is randomly extracted and learned from replay memory. Prior to the update, values such as reinforced learned malicious packet detection rate and behavior for reinforcement learning obtained by rewarding and interacting with the Markov decision process environment and malicious packet detection rate are quantified and added to memory.

In addition, since samples are accumulated in memory by repeating train and update, memory is mixed to break the correlation between memories (shuffle) and DQN is improved by utilizing memory in the mini-batch process. The Ipsilon Greed policy initially makes a lot of exploration, but it becomes more and more used over time.

While updating, we found the best way to act, but the rewards we receive now (malicious packet detection rate) when we receive malicious packet detection rate as a reward, the rewards we receive after one year, and the rewards we receive after five years are different. As compensation changes over time and is bound to be less than the current compensation, the Q value should address future compensation.

$$\widehat{\boldsymbol{Q}}(\boldsymbol{s},\boldsymbol{a}) \leftarrow \boldsymbol{r} + \boldsymbol{r}\boldsymbol{max}\,\widehat{\boldsymbol{Q}}(\acute{\boldsymbol{s}},\acute{\boldsymbol{a}})$$

Equation 1: Bellman equation

At this time, we use the Bellman equation of (Equation 1). We derive the Q behavior value function with the current reward (r), the r(gamma) used when the future reward is low, and the MaxQ with the highest Q value when optimal behavior is performed. The Bellman equation represents the relationship between the value function and the behavior function. Therefore, the Bellman equation used a Q behavior value function that, combined with the reward (R) for state, calculates behavior based on optimal policies and states that probabilistic predict the value of state as a value function to derive optimal behavior.

4 Experiments and evaluations

4.1 Experimental results

In order to compare the performance of the malicious packet detection model using reinforcement learning and the existing malicious packet detection model, three methods of performance evaluation were conducted using five algorithms: D-Tree classification, Logistic Regression, artificial neural network, SVM, and Random Forest. The first method was to measure the detection rate of existing network malicious packets by dividing a dataset of similar packet types into learning data and performance evaluation data at a ratio of 7 to 3. The results of the experiment follow Table 2.

Table 2: Experiment 1 Results

	Category	Accuracy	Performance Ranking	Learning Time
Experiment 1	D-Tree Classifier	98%	2	1 min
	Logistic Regression	84%	6	1 min
	Neural Network	95%	4	1 min
	SVM	95%	4	1 min
	Random Forest	99%	1	1 min
	Reinforcement Learning	97%	3	10 min

In the second way, in order to evaluate how many malicious packets are detected by learning from a small amount of data, 1,000 packet data were learned and 19,000 packet data were used as performance evaluation data to measure the detection rate of malicious packets. The results of the experiment follow Table 3.

Table 3: Experiment 2 Results

	Category	Accuracy	Performance Ranking	Learning Time
Experiment 2	D-Tree Classifier	70%	6	1 min
	Logistic Regression	77%	4	1 min
	Neural Network	78%	2	1 min
	SVM	78%	2	1 min
	Random Forest	86%	5	1 min
	Reinforcement Learning	93%	1	1 hour 30 min

As a third method, in order to measure the detection rate of new types of network malicious packets, learning was conducted with existing datasets and performance evaluation was conducted with datasets, which are network packets in new types. The results of the experiment follow Table 4.

Table 4: Experiment 3 Results

	Category	Accuracy	Performance Ranking	Learning Time
Experiment 3	D-Tree Classifier	34%	3	1 min
	Logistic Regression	67%	2	1 min
	Neural Network	24%	5	1 min
	SVM	24%	5	1 min
	Random Forest	34%	3	1 min
	Reinforcement Learning	80%	1	1 hour 30 min

4.2 Analysis of Experimental Results

Experiment 1 to detect existing packet types showed high detection rates in both supervised and reinforced learning methods. However, in terms of learning time, the existing malicious packet model completed learning faster than the malicious packet model using reinforcement learning. In Experiment 2 learned with small amounts of data and Experiment 3 measuring detection rates for new types of network malicious packets, the malicious packet detection model using reinforcement learning took a lot of learning time, but showed a higher detection rate than the existing malicious packet detection model.

5 Conclusion

In this paper, a method of efficiently detecting malicious packets using reinforcement learning was proposed. The implications for the research results are as follows.

First, the time to create training data can be shortened by compensating for the disadvantage of supervised learning that the model cannot be sufficiently trained if the number of data is too small. Second, by overcoming the problems and limitations of the original malicious packet detection system using supervised learning, the performance has been verified with a high detection rate even when new malicious packets or variant malicious packets appear, and thus reinforcement learning is expected to be efficiently used in the security field in the future. As a further study in the future, reinforcement learning using the DQN method uses the max operator in the formula of updating the Q-learning value, so Q-value is evaluated higher than it actually is, and as a result, learning tends to slow down. To compensate for this, research is needed to speed up the learning model using an algorithm called Double DQN, an evolved artificial neural network.

References

[1] Daesung Lee. “Next-generation cybersecurity trends.” Journal of the Korean Society of Information and Communication 23.11 (2019): 1478-1481.
[2] CISCO, Cisco Annual Internet Report (2018–2023) White Paper, March 2020.
[3] Reporter Park Sung-kyu, Cyber Terror Grown on Digitization and Corona, Global Damage in 2025, https://www.sedaily.com/NewsView/22L80BKPS0
[4] GugGyong-Wan, ByungCheol-Gong, "Trends in Security Technology Development Using Artificial Machine Learning and Deep Learning to Detect Malicious Packets 115 Intelligence".
[5] Reporter Won Byung-chul, DDoS in the financial sector? Just a new 'attack preview' by Armada Collective! https://www.boannews.com/media/view.asp?idx=55458,2017 .
[6] Sonicwall, 2021 Sonicwall Cyber Threat Report Mid-Year Update, July 2021.
[7] Xi-Lang Huang, Seon Han Choi, “A Simulation Sample Accumulation Method for Efficient Simulation-based Policy Improvement in Markov Decision Proces” Journal of Korea Multimedia Society Vol. 23, No. 7, July 2020(pp. 830-839).
[8] Jihyeon Park, Taeok Kim, Yulim Shin, Jiyeon Kim, Eunjung Choi, “Design and Implementation of a Pre-processing Method for Image-based Deep Learning of Malware” Journal of Korea Multimedia Society Vol. 23, No. 5, May 2020 (pp. 650-657).
[9] ZHIYANG FANG, JUNFENG WANG, JIAXUAN GENG and XUAN KAN “Feature Selection for Malware Detection based on Reinforcement Learning” IEEE Access, vol. 7, 2019, pp.176177-176187.
[10] Yilun He, “Convolution neural network(CNN) based image processing system”, 2017.
[11] Richard Sutton and Andrew Barto. Reinforcement Learning: An Introduction. MIT Press, 1998.
[12] Christopher JCH Watkins and Peter Dayan. Q-learning. Machine learning, 8(3-4):279–292, 1992.
[13] Tong He, Zhi Zhang, “Bag of Tricks for Image Classification with Convolutional Neural Networks”, 2018
[14] Hyeon-Ho Lee, CNN Generalization Error Evaluation Method, 2020
[15] Martin Riedmiller. Neural fitted q iteration–first experiences with a data efficient neural reinforcement

learning method. In Machine Learning: ECML 2005, pages 317–328. Springer, 2005.

Author Biography

Byung-Wook An is currently working as a security manager at NCSoft. His research interests include machine learning, deep learning, security, and artificial intelligence.

Joong-Chan Lee is currently pursuing a master's degree in the Department of Artificial Intelligence Convergence at Hanyang University. His research interests include machine learning, deep learning, speech recognition, image processing, security, and artificial intelligence.

Jai-Sung Choi is currently performing information security consulting work (technology and management) at the security group Moby Dick. His research interests include machine learning, deep learning, and security.

Park Won-hyung is a professor in the Department of Convergence Security at Sung shin Women's University. His received a bachelor's degree in industrial information systems from Seoul National University of Science and Technology in 2002 and a master's degree in information industry engineering from Seoul National University of Science and Technology in 2005. In addition, he received a bachelor's degree in information protection from Kyunggi University in 2009 and a doctorate in computer education from Sungkyunkwan University in 2015. He was an adjunct professor/director of cybersecurity at Far Eastern University from 2012-2020 and an adjunct professor of information security engineering at Sangmyung University from 2021-2022.