IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 March 2024, accepted 2 April 2024, date of publication 15 April 2024, date of current version 23 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3389067

RESEARCH ARTICLE

An Adaptive Policy-Based Anomaly Object Control System for Enhanced Cybersecurity

WON SAKONG AND WOOJU KIM
Department of Industrial Engineering, Yonsei University, Seoul 03722, South Korea
Corresponding author: Wooju Kim (wkim@yonsei.ac.kr)

This work was supported by the Industrial Strategic Technology Development Program-ATC+ (Development of Life Cycle Management Platform by providing Artificial Intelligence-Based Real-Time Model Observability and Explanability) funded by the Ministry of Trade, Industry and Energy (MOTIE), South Korea, under Grant 20023280.

ABSTRACT Anomaly detection research focuses on identifying rare patterns derived from daily occurrences. This study introduces an innovative anomaly–object control system that utilizes adaptive policies through anomaly detection algorithms. Effectively blocking anomalous objects in real–world scenarios poses significant challenges. Therefore, we empirically validate the proposed anomaly object control methodology using the traffic history associated with malicious cyber–attacks in vulnerable network environments. We propose an anomaly object control methodology based on DeepSARSA that utilizes unsupervised anomaly detection deep learning models trained on historical data collected from an environment in which the anomaly object control system operates. Through this approach, we confirmed the adaptive policies for optimal anomaly object control. By employing the out–of–distribution detection and DeepSVDD algorithms as reward functions and comparing the results, we verified the stability of the proposed anomaly object control system. Our experimental results highlight the practical limitations of single–class anomaly detection algorithms and propose new research directions for anomaly detection.

INDEX TERMS Anomaly detection, deepSARSA, deepSVDD, network intrusion detection, network intrusion response, ODIN.

I. INTRODUCTION

Anomaly detection is a critical area of research aimed at discerning abnormal patterns, including faults, fraud, diseases, and intrusions. Despite the infrequent occurrence of abnormal patterns, their impact on regular operation is substantial. Imbalanced datasets, in which most of the collected data are normal, pose a significant challenge for anomaly detection. Various algorithms such as Isolation Forest [1], one–class support vector machine (OCSVM) [2], support vector data description (SVDD) [3], autoencoder [4], [5], and deepSVDD [6] have been employed for anomaly detection. These algorithms learn common patterns from the training data, enabling them to identify abnormal patterns based on deviations from the learned norms.

Research on anomaly detection has primarily focused on enhancing the accuracy of the detection algorithms. This study goes a step further by proposing the practical application of a learned anomaly detection algorithm to trigger alerts and as a criterion for real–time anomaly control. Experiments were conducted in a network communication environment to assess the feasibility of using the anomaly detection algorithm as an objective function for anomaly control systems.

Network communication environments are particularly susceptible to anomalies, with distributed denial of service (DDoS) attacks representing a prominent example. Ongoing research is focused on creating DDoS attack response systems that utilize reinforcement learning. DDoS attacks involve the transmission of an overwhelming volume of abnormal traffic from a botnet to a specific destination, depleting the computing resources necessary for network

The associate editor coordinating the review of this manuscript and approving it for publication was Muammar Muhammad Kabir.

communication at the targeted destination [7], [8]. DDoS attacks exhibit significant diversity, contingent upon the attacker's design, encompassing variations in the consumption of computing resources and attack scales. DDoS attacks manifest in various forms, depending on the type of computing resources they deplete, such as UDP, TCP, and ICMP flooding attacks. However, the scale and methodologies of these attacks vary significantly depending on the attacker's design [9], [10].

In response to DDoS attacks, the objective is to block abnormal traffic as much as possible during anomalous situations while maximizing the allowance of legitimate traffic during typical circumstances. References [11], [12], [13], [14], [15], [16], and [17] have directed efforts towards mitigating DDoS attacks by controlling the volume of traffic packets exceeding the bandwidth entering the network servers. Reference [18] proposed a reward function based on temporal changes in legitimate and abnormal traffic volumes.

To effectively mitigate the anomaly of DDoS attacks, it is imperative to establish patterns that define the normal network communication states. However, even among identical IoT devices, the spectrum of normal network communication patterns can exhibit significant variability, depending on contextual factors. Moreover, the heterogeneity of IoT introduces further complexity, potentially leading to shifts in pre–trained normal patterns. Numerous studies have addressed this heterogeneity in network environments.

Representative methodologies include the development of efficient strategies for controlling malicious traffic, which incorporates feature selection tailored for dynamic environments and formulates mitigation rules derived from selected features [19]. Additionally, adaptive approaches are grounded in multilayer forwarding frameworks, leveraging various classification models such as support vector machines, naive Bayes, random forests, k-nearest neighbors, and logistic regression. Ensemble techniques have also been employed to bolster the accuracy of DDoS attack detection while simultaneously reducing false alarm rates [20]. Furthermore, methods have been developed to detect and respond to DDoS attacks in IoT environments by integrating Long Short–Term Memory (LSTM) networks with adaptive labeling and diagnostic insights [21].

Within the domain of reinforcement–learning–based DDoS attack control, there exist methodologies such as distributed DDoS response systems that leverage adaptive rewards, such as coordinated team learning, to reflect the inherent structural characteristics of networks [22].

We propose that our research address this problem from a new perspective, diverging from existing studies. In practical network environments, the traffic observed on network devices is directed toward different destinations. Consequently, controlling traffic at routers based on the total packet volume directed to a specific server, as employed in traditional DDoS attack–mitigation methodologies, poses significant challenges. In addition, when there are changes in the network structure, retraining the policies of all agents responsible for traffic control becomes necessary.

Traditional anomaly detection research involves training algorithms with patterns from normal data and subsequently detecting anomalies. However, deviations from learned normal patterns often result in low accuracy. The detection of anomalies is aimed at their effective control. We structured the following organization to demonstrate the capability of controlling anomalies in real world environments using algorithms trained for anomaly detection. In Section II, we introduce anomaly detection algorithms and reinforcement learning algorithms for DDoS attack control systems(i.e., anomaly object control systems). Section III presents the methodology for anomaly control using data–based reward functions. Section IV covers the experiments and results, while Sections V and VI summarize the conclusions and limitations, and outline future research directions.

II. RELATED RESEARCH

A. DEEP SVDD

Anomaly detection involves identifying abnormal data by learning the characteristics of the normal data. One classical anomaly detection algorithm is SVDD. In SVDD, normal data in the feature space F_k are trained to lie within a boundary defined by a center $c \in F_k$ and radius $R > 0$ to minimize R, where k is the kernel function. In (1), ν is a regularization parameter and ξ is a slack variable.

$$\min_{R,c,\xi} \frac{1}{\nu n} \sum_i \xi_i$$
$$\text{s.t.} \quad \|\phi_k(x_i) - c\|^2_{F_k} \le R^2 + \xi_i,\ \xi_i \ge 0, \quad \forall i \qquad (1)$$

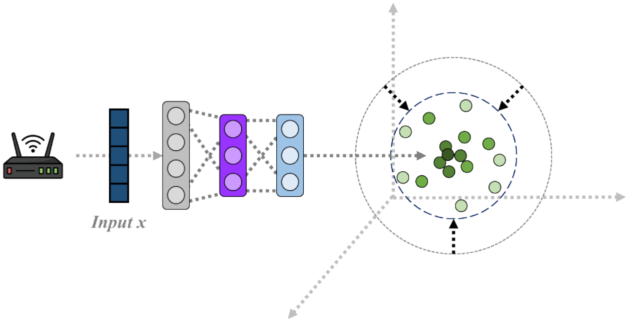

FIGURE 1. Training process of deep SVDD.

In the case of Deep SVDD [23] (see Fig.1), instead of the kernel space in SVDD, a deep learning model was employed to output the representation of the data. Simultaneously, the goal is to determine the smallest enclosing sphere in the representation space. The objective function for the soft–boundary Deep SVDD is as follows:

$$\min_{R,w} R^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \max\{0, \|\phi(x_i; w) - c\|^2 + R^2\} + \frac{\lambda}{2}\sum_{l=1}^{L} \|W_l\|_F^2 \tag{2}$$

When expressing the existing data in a new representation F using the deep learning weights w, the learning process is conducted by connecting w to minimize the center c and radius R while encompassing the normal data. The anomaly score $s(x)$ is defined by. Eq(3).

$$s(x) = \begin{cases} 1 & \text{if } \|\phi(x_i; w^*) - c\|^2 \leq R^2 \\ -1 & \text{if } \|\phi(x_i; w^*) - c\|^2 > R^2 \end{cases} \tag{3}$$

B. ODIN

Out–of–distribution is a subfield of anomaly detection aimed at methodologies for detecting heterogeneous data from various classes of normal distributions. The most popular method for out–of–distribution detection is the out–of–distribution detector for neural networks (ODIN) [24]. ODIN can detect out–of–distribution by adding "temperature scaling" (see Fig.2) and "perturbation addition" (see Fig.3) processes to a pre–trained classification model without additional training. Equation (4) is the formula for the temperature–scaling process, where $S_i(x; T)$ indicates the softmax score of class i for input x. Given x, the classification model outputs the classes with the highest softmax function values. Temperature scaling reflects the scaling parameter T in the softmax function, where $T \in \mathbb{R}^+$, and reduces the overconfidence of the model [25].

$$S_i(x; T) = \frac{e^{\left(\frac{f_i(x)}{T}\right)}}{\sum_{c=1}^{C} e^{\left(\frac{f_c(x)}{T}\right)}} \tag{4}$$

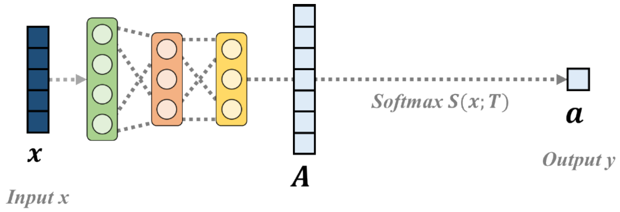

FIGURE 2. Process of temperature scaling.

Equation (5) expresses the perturbation calculation process, where ϵ indicates perturbation magnitude. The addition of perturbation increases the softmax score for a given input by subtracting a small amount of perturbation from the input. This process strengthens the predictions for in–distribution samples and helps them be better separated from out–of–distribution samples [26].

$$\tilde{x} = x - \varepsilon \text{sign}\left(\nabla_x \log S_{\hat{y}}(x; T)\right) \tag{5}$$

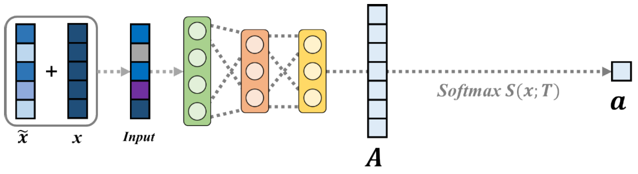

FIGURE 3. Process of perturbation addition in ODIN.

In (6), given parameters δ, T, and ϵ, if the highest probability that input $\tilde{x}$ belongs to class i is higher than δ, it is classified as an in–distribution.

$$g(x; \delta, T, \varepsilon) = \begin{cases} 1 & \text{if } \max_i p(\tilde{x}; T) \leq \delta \\ 0 & \text{if } \max_i p(\tilde{x}; T) > \delta \end{cases} \tag{6}$$

C. REINFORCEMENT LEARNING

Reinforcement learning optimizes an agent's decision–making based on rewards for repeated interactions in a dynamic environment. Reinforcement learning consists of state s, action a, and reward r. First, which refer to the environment observed by the agent; second, the action relates to the agent's movement based on the observed environment; the reward refers to the evaluation result of the action performed. Agents must select the appropriate action to obtain the maximum reward according to the observed environment. This decision–making is called policy π. Note that reinforcement learning aims to determine the optimal policy [27].

If the agent can define all observable environments, model–based reinforcement learning is used; however, because the number of observed environments cannot be described in a real environment, a model–free reinforcement learning method is used to infer the state. The method of improving the policy based on the reward obtained in the final state of the episode reached by the agent is called the Monte Carlo learning method. By contrast, the method of improving the policy with the reward of the new state at each moment of the episode is called the temporal–difference learning method. The on–policy method is used for cases in which the behavior policy used for decision–making and the target policy for improvement during the scenario are the same. By contrast, the off–policy method is employed for cases in which the behavior and target policies differ. When an agent optimizes a policy for selecting actions using a neural network, this is called deep reinforcement learning (see Fig.4).

State–action–reward–state–action (SARSA) and Q–learning are traditional reinforcement learning methods. Both are model–free reinforcement learning methods that are based on the temporal difference learning method. However, unlike Q–learning, SARSA uses on–policy based reinforcement learning to improve the behavior policy before performing actions in each state. The update function for

SARSA is as follows:

$$Q^{new}(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma Q(s',a') - Q(s,a)\right] \quad (7)$$

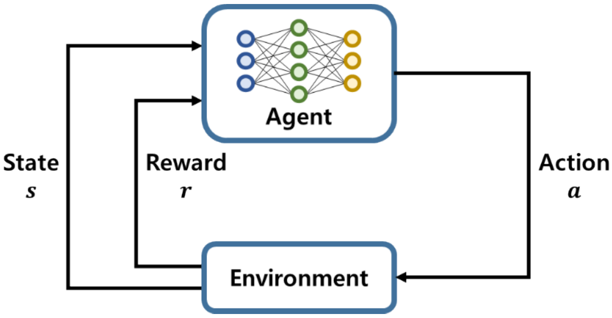

FIGURE 4. **Deep reinforcement learning.**

where $Q(s,a)$ refers to the current Q–value; s, a, and r are the current state, action, and reward, respectively; s' and a' refer to the next state and action, respectively; α is the learning rate ($0 \leq \alpha \leq 1$); and γ is the discount factor ($0 \leq \gamma \leq 1$). Deep SARSA is a deep reinforcement learning method based on SARSA that uses a neural network as a Q–function approximator $Q(s,a;\theta)$ [27], [28], [29], [30]. The input of the neural network is state s, the output is the Q–value of each action, and θ is a parameter of the neural network. The neural network was trained to optimize the loss function L as in (8) for a more accurate Q–value prediction.

$$L_i(\theta_i) = (y_i - Q(s,a;\theta_i))^2 \quad (8)$$

At the i–th iteration, $y_i = r + \gamma Q(s',a';\theta_{i-1})$. Similar to the DQN [31], the agent selects an action using ε–greedy in the current state.

D. DDoS DETECTION

DDoS attack detection is typically based on anomaly detection, which uses statistical or machine learning techniques. Because the amount of network traffic that constitutes a DDoS attack is large, all traffic accessed simultaneously is formed into one cluster, and the attack is detected based on the cluster pattern. Various patterns can be used to determine whether a cluster is normal or if a DDoS attack has occurred. The simplest method involves checking the total number of packets in the cluster. During a DDoS attack, the total number of packets transmitted to the target server increases compared with normal times. However, additional information is required because increasing the number of packets is not necessarily a pattern that occurs only in DDoS attacks.

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_2 P(x_i) \quad (9)$$

The entropy, expressed in (9), measures the impurity (or randomness) present in a dataset. X is traffic variable and has n values. $P(x_i)$ is the ratio of x_i in the traffic, forming a cluster that satisfies $\sum_{i=1}^{n} P(x_i) = 1$. If the impurity of the dataset also increases, the entropy value increases. Network equipment can monitor information, such as the source IP address, source port number, destination IP address, destination port number, protocol, and packet type of traffic passing through the device. The features of a cluster are the entropy values of piece of information on all traffic constituting the cluster.

During a DDoS attack, the distribution of traffic features changes in various directions [32]. For example, when a DDoS attack occurs, the amount of network traffic connected to the attack target server increases more than usual; thus, the entropy for the traffic source IP address during a specific time period increases. In addition, because DDoS attacks are aimed at a single target, the entropy value for the destination IP address of the traffic observed from network equipment decreases. Therefore, the entropy information of each variable is the most commonly used cluster pattern for identifying DDoS attacks [33], [34], [35].

The entropy–based DDoS attack detection method is simple and advantageous in terms of high sensitivity and low false positive rate. However, detection using single–attribute entropy has the disadvantage of a high false positive rate for forged attacks. Therefore, a DDoS attack detection method was proposed, that utilizes conditional entropy to accurately analyze the N–to–1 relationship for each feature, representing a typical pattern of DDoS attacks [36], [37], [38]. The conditional entropy of variable Y to variable X can be defined as

$$H(Y|X) = -\sum_{i=1}^{n} P(x_i)\sum_{j=1}^{m} P(y_j|x_i)\log_2 P(y_j|x_i) \quad (10)$$

where Y is a variable different from X; y_j is the variable value of Y; and the number of types is m. Fig.5 shows how the entropy calculation is used in DDoS attack detection.

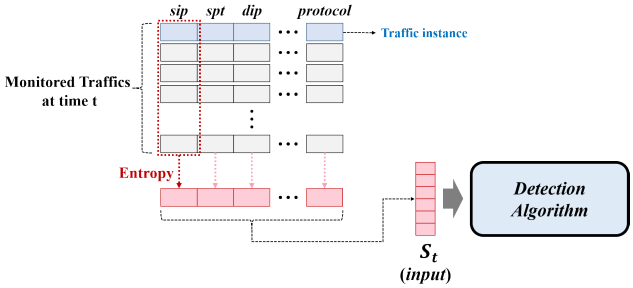

FIGURE 5. **Entropy–based vector processing for DDoS detection.**

III. METHODOLOGY

Fig.6 shows the entire structure of the anomaly object control system. The methodology is designed to monitor the comprehensive state of traffic at each point, assess the anomaly status of the action results using a pre–trained reward algorithm (anomaly detection algorithm), and then select the action that returns the optimal normal state (Anomaly Object Control).

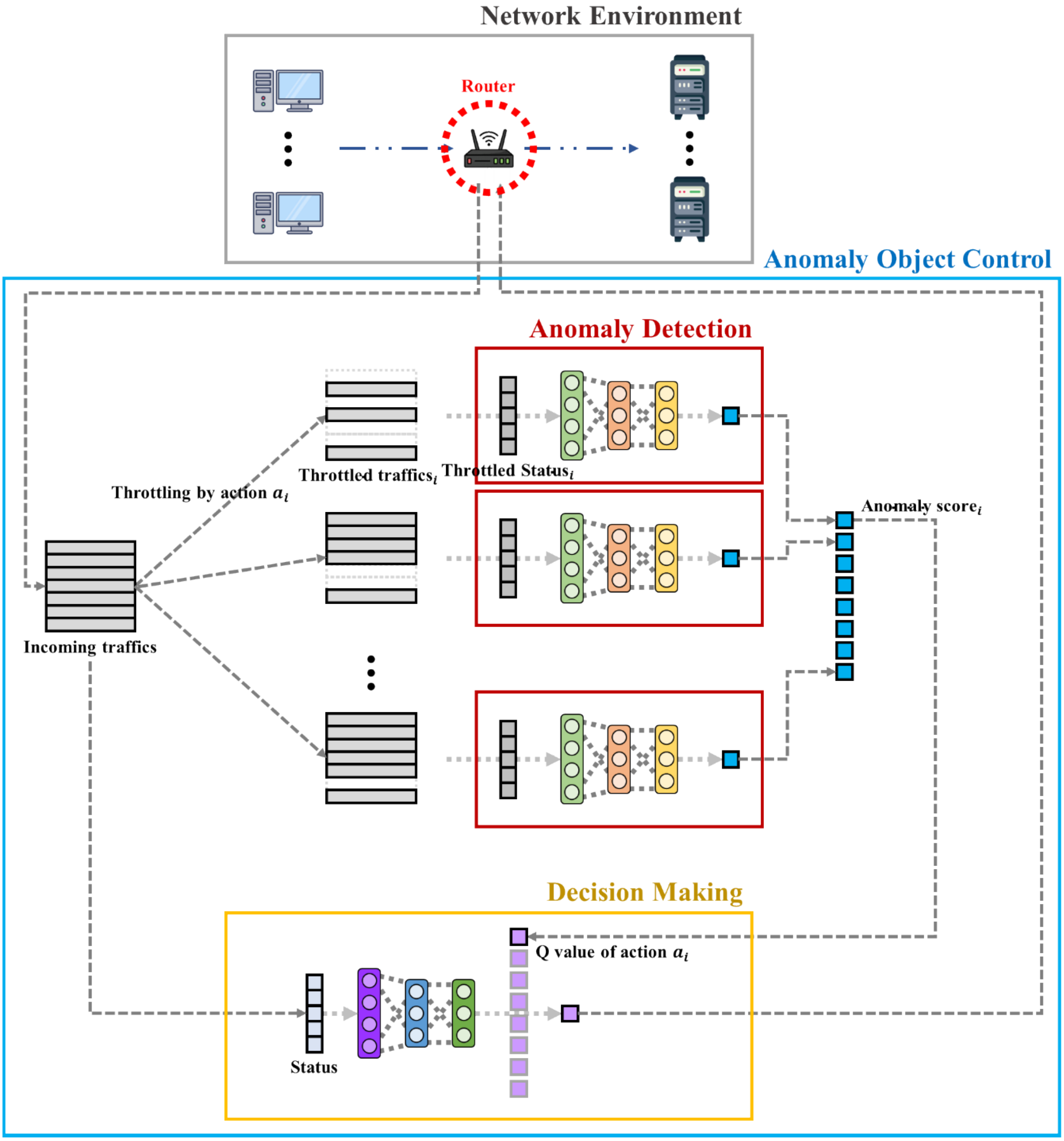

FIGURE 6. Architecture of anomaly object control system.

A. ANOMALY OBJECT CONTROL

The anomaly object control was constructed based on a Deep SARSA network. At time t, the state of the observed incoming traffic instances i_t at the router is denoted as s_t, and the state of the allowed traffic instances p_t after throttling to a specific proportion by the selected action a is represented as s_{a_t}. Both s_t and s_{a_t} are vectors composed of entropy–based features for DDoS detection and serve as inputs to the anomaly detection algorithm. The action employed in DDoS control research is predominantly based on utilizing max–min fairness. The anomaly detection model evaluates whether the input s_{a_t} is normal (in distribution). In network traffic environments, the class for normal (in–distribution) is defined as 1, and the class for anomalies is defined as -1. The reward r is formulated by multiplying the class value with $\frac{\sum p_t}{\sum i_t}$ aiming to allow the maximum amount of traffic to pass through in normal scenarios, while minimizing the passage of traffic in abnormal situations. Anomaly detection algorithms for the reward function in anomaly control systems are constructed based on historical data, leveraging pre–trained normal patterns.

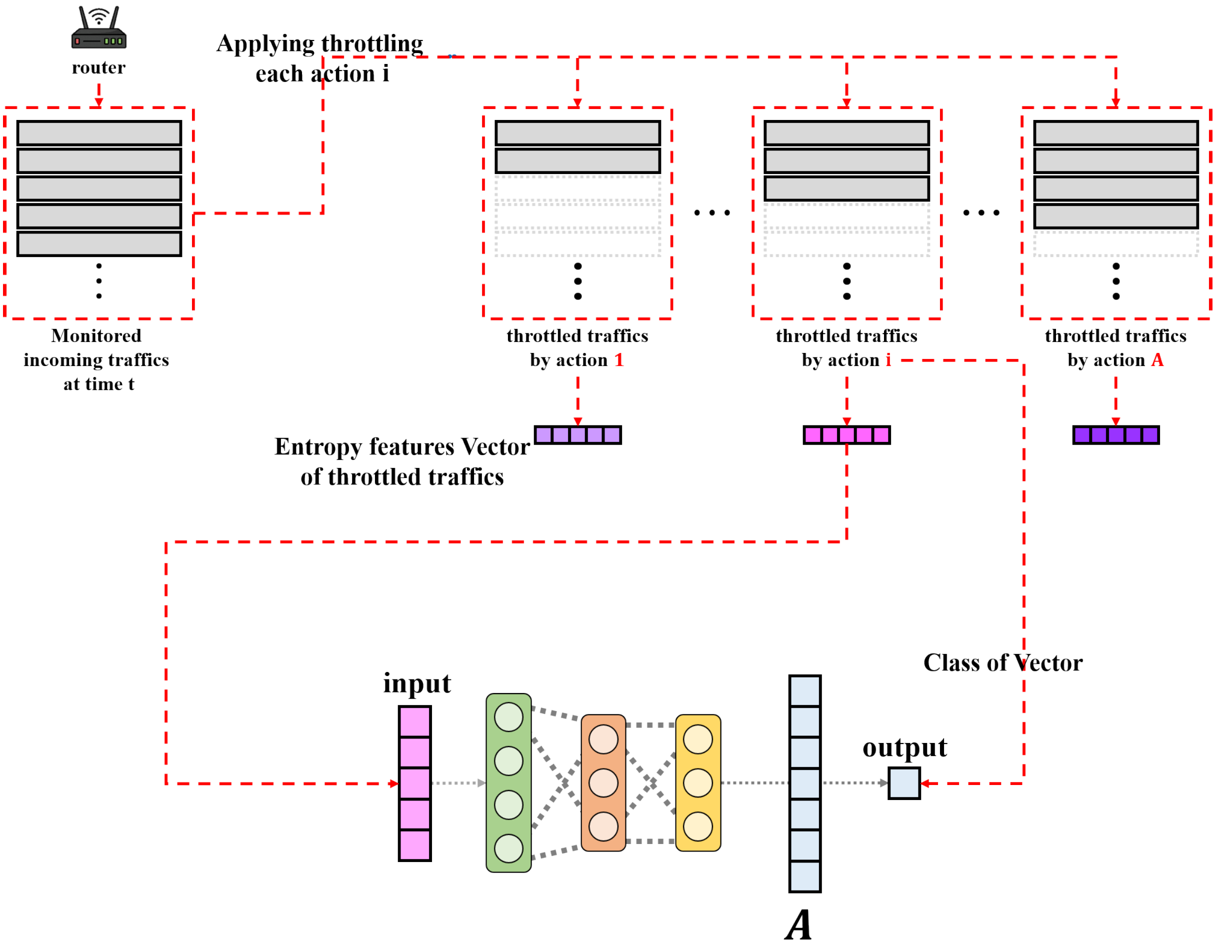

FIGURE 7. **Process of multiclass labeling and training ODIN.**

B. ANOMALY DETECTION

The anomaly detection algorithm considers the patterns in the training data to be normal. However, in real–world scenarios, the observed traffic states at the router can exhibited much greater diversity than that is captured in the training data. Consequently, errors may occur when identifying a class as abnormal, even if it is genuinely normal but differs from learned patterns. Therefore, ODIN was utilized as an anomaly detection algorithm to address this issue, and was designed to classify the possible traffic states in the actual environment into multiple classes. This approach allows the algorithm to learn a wide range of normal patterns, accommodating potential variations in the observed traffic states in real–world scenarios. As network bandwidth ratio values can define all conceivable cases of traffic observed at the router, each momentary s_{a_t} of observed traffic, categorized as normal, is trained as in–distribution by the algorithm (see Fig.7).

IV. EXPERIMENTS

A. DATASETS

The experiment was conducted using three distinct datasets. The first dataset comprises the "CIC–IDS 2018" dataset [39], encompassing network traffic data generated between Feburary 20, 2018, 00:00:00, and Feburary 20, 2018, 12:59:59. This dataset includes 7,372,557 instances of legitimate network traffic originating from 31,281 source IPs and transmitted to 27,076 destination IPs. Additionally, it encompasses a malicious traffic history consisting of 576,191 instances initiated from 10 source IPs and directed toward a single destination IP during the same time frame.

The second dataset, denoted as "CIC–DDoS 2019" [40], is a multi–dataset comprising attack histories such as DrDoS DNS, DrDoS LDAP, DrDoS MSSQL, DrDoS NetBIOS, DrDoS NTP, DrDoS SNMP, DrDoS SSDP, and DrDoS UDP. Four specific attack types were employed for the experimentation: DrDoS LDAP, DrDoS UDP, DrDoS MSSQL, and DrDoS NetBIOS. For DrDoS LDAP, the dataset comprises 6,736 instances of legitimate network traffic and 4,288,040 instances of malicious traffic. The DrDoS UDP includes 5,291 instances of legitimate network traffic and 6,913,717 instances of malicious traffic. The DrDoS MSSQL and DrDoS NetBIOS datasets each contain 4,800 instances of legitimate network traffic and 10,295,484 instances of malicious traffic, as well as 3,028 instances of legitimate network traffic and 7,547,857 instances of malicious traffic, respectively. In this study, the traffic history from November 03, 2018, was utilized for training and validating anomaly detection algorithms, while the traffic history from December 11, 2018, was employed to generate reinforcement learning scenarios.

Finally, real–world data provided by a corporate entity are employed in this study to confirm applicability of methodology. The dataset was collected between January 19, 2022, 00:00:00, and 11:00:00, and comprised 7,764,896 instances of legitimate traffic originating from 905,585 source IPs and transmitted to 2,011 destination IPs. Additionally, the dataset included 17,327 instances of malicious traffic initiated from 6,483 source IPs and directed toward a single destination IP during the same time frame.

B. ANOMALY DETECTION

The input vector for the anomaly detection algorithm comprised features aimed at identifying the presence of a DDoS attack. The classes associated with the input vector were configured to be 20 and aligned with the action range of the reinforcement learning model. The input vector is constructed based on temporal variations, encompassing the "entropy of source IP address," "entropy of source port number," "entropy of destination IP address," "entropy of destination port number," "entropy of packets," "conditional entropy of source IP address given destination IP address," "conditional entropy of source IP address given source port number" and "conditional entropy of destination IP address given destination port number." We conducted experiments by dividing each entropy dataset into training, validation, and test sets at a ratio of 6:2:2.

TABLE 1. Performance of multiclass–based anomaly detection.

	ODIN			
	Accuracy	FPR	AUROC	AUPRC
CIC–IDS	0.9549	0.0970	0.9132	0.9818
Real–world dataset	0.8315	0.2937	0.8675	0.9934

The ODIN algorithm was formulated as a fully connected neural network comprising three hidden layers with a softmax function integrated into the output layer. To accommodate the variation in the number of classes, the training of the loss function was executed using weighted cross entropy. The temperature scaling parameter was set to 1,000, perturbation magnitude to 0.001, and threshold for in–distribution to 0.9. In [24], the temperature scaling and perturbation magnitude parameters were selected based on the point where the True Positive Rate (TPR) reached 95%. We initially naively set these parameters for the CIC–IDS dataset during the experimental phase. However, the initial parameter settings yielded a high TPR and low false positive rate (FPR) for the test set, as shown in Table 1, obviating the need for further parameter optimization. Conversely, when dealing with a real–world dataset, we set the temperature scaling parameter to 1,000, and conducted a grid search to determine the perturbation magnitude parameter that resulted in a TPR of 86.7%, which is the highest of our training set. This process guided us to set the perturbation magnitude parameter to 0.0014. The datasets for DrDoS DNS, DrDoS LDAP, DrDoS SNMP, and DrDoS SSDP were not utilized in ODIN training because of the minimal variability in the observed number of legitimate traffic instances across different time points.

TABLE 2. Performance of single–class–based anomaly detection.

		Deep SVDD			
		Accuracy	Precision	Recall	AUROC
CIC–IDS 2018		0.9946	0.9951	0.9993	0.9671
CIC–DDoS 2019	DNS	0.9955	0.9961	0.9992	0.9387
	LDAP	0.9711	0.9754	0.9941	0.9283
	SNMP	0.9100	0.9697	0.9316	0.9298
	SSDP	0.9362	0.9754	0.9560	0.9142
Real–world dataset		0.9151	0.9946	0.9164	0.9479

To assess the performance of the out–of–distribution detection algorithm as a data–driven reward function of reinforcement learning, we conducted tests using the Deep SVDD algorithm, which is a single–class anomaly detection algorithm. In the Deep SVDD algorithm, the input vectors were categorized as normal or abnormal. The anomaly detection performance of the Deep SVDD algorithm is presented in Table 2.

TABLE 3. Average throttling performance of each reward function in normal scenarios.

		Baseline Reward	Deep SVDD	ODIN
CIC–IDS 2018		98.42(%)	40.65(%)	94.41(%)
CIC–DDoS 2019	DNS	95.12(%)	84.96(%)	85.84(%)
	LDAP	90.51(%)	83.03(%)	92.16(%)
	SNMP	88.52(%)	81.18(%)	88.61(%)
	SSDP	88.66(%)	82.87(%)	87.93(%)
Real–world dataset		82.78(%)	42.82(%)	92.16(%)

C. ANOMALY OBJECT CONTROL

To evaluate the performance of the reinforcement learning based anomaly detection algorithm, we conducted comparative experiments using the reward function proposed in [18]. The temporal traffic variation ΔT in (11) is calculated using (12), which represents the change in traffic volume at each time point. Here, N denotes the volume of legitimate traffic each time, and M represents the volume of malicious traffic each time.

$$r = \begin{cases} ((1+\Delta T_{init})^2 - 1)|1+\Delta T_{step}|, & \Delta T_{init} > 0, \\ (-(1-\Delta T_{init})^2 - 1)|1-\Delta T_{step}|, & \Delta T_{init} \leq 0. \end{cases} \tag{11}$$

$$\Delta N_{\mathrm{init}} = \frac{N_i - N_0}{N_0}, \Delta N_{\mathrm{step}} = \frac{N_i - N_{i-1}}{N_{i-1}},$$
$$\Delta M_{\mathrm{init}} = \frac{M_i - M_0}{M_0}, \Delta M_{\mathrm{step}} = \frac{M_i - M_{i-1}}{M_{i-1}}. \tag{12}$$

TABLE 4. Average throttling performance of each reward function in DDoS scenarios.

		BaseLine Reward		Deep SVDD		ODIN	
		Legitimate traffic(%)	Malicious traffic(%)	Letigitimate traffic(%)	Malicious traffic(%)	Legitimate traffic(%)	Malicious traffic(%)
CIC–IDS 2018		61.50	41.79	42.38	37.12	29.36	25.41
CIC–DDoS 2019	DNS	78.62	64.57	61.93	48.44	-	-
	LDAP	72.28	65.24	59.60	42.52	-	-
	SNMP	68.83	69.46	60.32	41.46	-	-
	SSDP	71.45	72.56	58.41	32.28	-	-
Real–world dataset		85.12	62.25	46.64	34.46	49.56	31.08

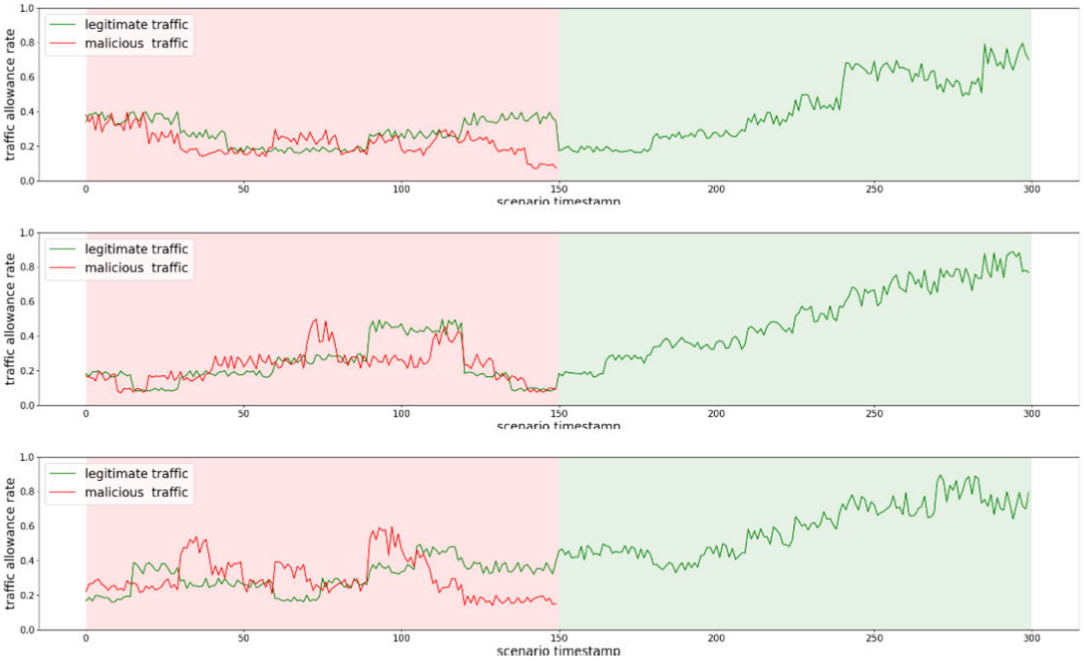

FIGURE 8. Example of control results based on the ODIN reward function in supplementary scenarios.

The reinforcement learning reward function R is defined by (13), where δ serves as a trade–off parameter between maximizing the passage of legitimate traffic and throttling malicious traffic. In this study, δ is set to 0.5.

$$\overline{R} = \delta r_M + (1 - \delta) r_N \tag{13}$$

In the real world, a broader range of patterns is observed than those encompassed by the training dataset. To verify the robustness of an algorithm trained on specific data as a reward function, the scenario data underwent several manipulations. These manipulations include the following:

1) Scenarios were constructed to represent both normal situations and instances of DDoS attacks.
2) The initiation of each scenario occurred at a randomly chosen time point within the entire dataset, and the length of each scenario was set to 300–time steps.
3) Within each scenario, 10 to 20 destination IP addresses were randomly selected from all destination IPs associated with the traffic. Only traffic attempting to connect to these chosen destination IPs was included in the scenario.
4) For each scenario selected above, a random proportion of traffic, ranging from 10% to 20%, is removed from the total traffic to introduce variability in the dataset.
5) The normal and DDoS attack scenarios were constructed in two variations: scenarios where a certain amount of traffic was maintained from the start to the end and scenarios where the traffic gradually increased from the start to the end.

The simulation scenarios were divided into two sets: 100 with only legitimate traffic inflow and 100 with a combination of legitimate and malicious traffic. Training iterations were conducted for 10,000 cycles. Tables 3 and 4 list the traffic control outcomes for each reward function in the legitimate and DDoS attack scenarios, respectively. For the baseline reward function from [18], in the case of all legitimate scenarios, the majority of legitimate traffic was permitted at a 90% rate. In contrast, for Deep SVDD, unlike

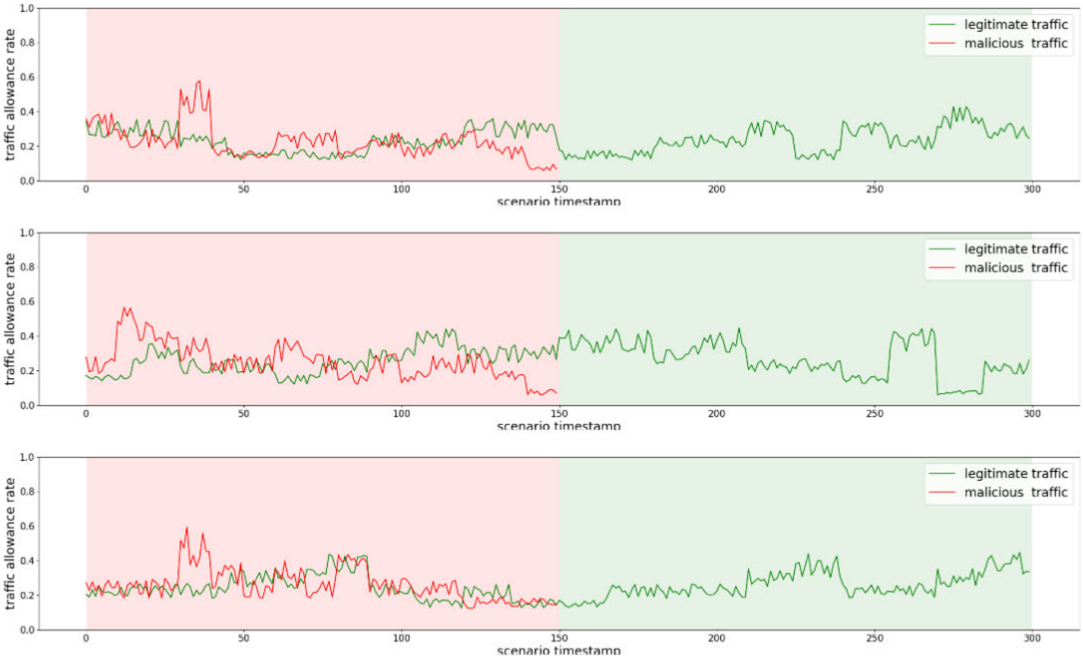

FIGURE 9. Examples of control results based on the DeepSVDD reward function in supplementary scenarios.

scenarios utilizing DNS, LDAP, SNMP, and SSDP, lower traffic acceptance rates were observed in legitimate scenarios using the CIC–IDS2018 and the Real–world datasets. This shows that Deep SVDD, when trained on normal patterns of a single class, has a limited detection performance for different normal patterns learned in real–world environments. For the reward functions utilizing ODIN, the average acceptance rate for various normal patterns across each scenario was approximately 90%, demonstrating a control performance similar to that of related studies. In scenarios involving anomalies, the baseline reward exhibited exceptionally high acceptance rates for legitimate and anomalous traffic. In DDoS scenarios created based on a real–world dataset, the average acceptance rate for legitimate traffic was the highest at 85.12%, whereas in DDoS scenarios based on CIC–IDS 2018, the average acceptance rate for legitimate traffic was the lowest at 61.50%. However, in scenarios based on CIC–DDoS 2019 SNMP and SSDP, the average acceptance rate of malicious traffic was higher than that of legitimate traffic. In all anomalous scenarios, reward functions based on ODIN exhibited the lowest average acceptance rates for legitimate and anomalous traffic.

D. SUPPLEMENTARY PERFORMANCES

To verify the stability of the anomaly detection algorithm as a reward function for reinforcement learning, we conducted additional experiments based on new scenarios using the CIC–IDS 2018. The scenarios were designed to gradually decrease the volume of malicious traffic, eventually resulting in legitimate traffic observations. In this scenario, we examined changes in the traffic allowance capacity of the algorithm.

During DDoS attacks, ODIN and Deep SVDD–based reward functions maintain meager acceptance rates for legitimate and malicious traffic. However, in legitimate traffic intervals, the ODIN–based reward function exhibited an increasing trend in the traffic acceptance rates (see Fig.8). Conversely, the Deep SVDD–based reward function tended to inaccurately assess the traffic situation as normal, even after the conclusion of the DDoS attack period (see Fig.9).

V. CONCLUSION

This study aims to evaluate the effectiveness of an anomaly detection algorithm in real–time changing environments using both legitimate and malicious network traffic histories. Anomaly detection algorithms learn normal patterns as a single class and identify divergent patterns as anomalies. However, the performance of these algorithms tends to degrade significantly in real–world settings, where it is difficult to anticipate every normal situation. This is because divergent patterns may include actual anomalous patterns and new normal patterns that are not included in the training of the algorithm.

The Deep SVDD–based algorithm exhibited high accuracy in anomaly detection during the training process. However, in dynamic environments, its performance is significantly diminished. The acceptance rate for normal traffic was high in scenarios in which the traffic pattern closely resembled the learning pattern of the Deep SVDD algorithm. However, the acceptance rate declined in scenarios with gradually increasing traffic volumes.

This study enhanced the ODIN by using it as a data–driven reward function in reinforcement learning to address this issue. The algorithm was trained with the control outcomes for each action as distinct normal classes, increasing its utility as a reward function. The acceptance rate for legitimate traffic was consistently lower than that of the baseline reward function in all tested anomaly scenarios. However, the acceptance rate for malicious traffic is also low. Furthermore, in additional experiments, it was observed that the traffic acceptance rate gradually increased in normal segments, which had previously remained low during DDoS attack segments.

In our methodology, calculating the observed traffic into state vector S involves exploring the data for each k variable to understand the distribution of variables and counting each variable's unique values. Therefore, the computational complexity is $O(n \log n)$, and the time complexity is represented as $O(nk)$, iterating through each variable's value to measure entropy for observed traffic counts k. Moreover, both the time and computational complexities of max–min fairness scheduling can be expressed as $O(n \log n)$.

For the ODIN algorithm, which receives state vectors throttled by actions as input, we designed it with a three–hidden layer fully connected structure, with the activation function being "ReLU." With a total of A classes in the distribution and an input vector length of l, the computational and time complexities are both represented as $O(l \times h_1 + h_1 \times h_2 + h_2 \times h_3 + h_3 \times A)$, where h_i represents the number of nodes in each hidden layer.

Finally, for DeepSARSA designed with a fully connected structure with three hidden layers, the computational complexity is $O(l \times h_1 + h_1 \times h_2 + h_2 \times h_3 + h_3 \times A)$, and the time complexity, with a batch size of b, is $O(l \times h_1 + h_1 \times h_2 + h_2 \times h_3 + h_3 \times A + b)$.

VI. LIMITATION AND FUTURE WORKS

We introduced a reinforcement learning–based anomaly detection control system that utilizes ODIN as the reward function to improve the stability of the control system. However, some areas require further refinement.

First, the action process used in this study employed max–min fairness for comparison with existing reinforcement learning–based DDoS attack control research. However, max–min fairness allocates bandwidth based on packet size per traffic, presents practical limitations in blocking malicious traffic in scenarios where DDoS attacks occur because malicious traffic tends to be smaller than normal traffic. Additionally, there is the issue of increasing computational and time complexity when blocking traffic based on traffic content because of the large amount of traffic monitored at each time point, making it challenging to control large traffic volumes promptly.

Second, in network environments, the total traffic sum defining the reinforcement learning state does not exceed the bandwidth, allowing for the definition of normal classes based on the ratio of the traffic control bandwidth. However, unforeseen situations may arise in real–world environments that require anomalous control. Therefore, there is a need to enhance pre–trained anomaly–detection algorithms. During the execution of the proposed anomaly control system, anomalies identified by the anomaly detection algorithm may include the possibility of new types of normal that were not pre–trained. In such situations, human administrators' interventions can only determine the algorithm updates. We anticipate that the practicality of data–driven anomaly control methodologies will be enhanced by incorporating research into continual learning or incremental learning to continuously update anomaly detection algorithms in future research.

REFERENCES

[1] G. Xing, J. Chen, R. Hou, L. Zhou, M. Dong, D. Zeng, J. Luo, and M. Ma, "Isolation forest-based mechanism to defend against interest flooding attacks in named data networking," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 98–103, Mar. 2021, doi: 10.1109/MCOM.001.2000368.

[2] L. Mhamdi, D. McLernon, F. El-moussa, S. A. R. Zaidi, M. Ghogho, and T. Tang, "A deep learning approach combining autoencoder with one-class SVM for DDoS attack detection in SDNs," in *Proc. IEEE 8th Int. Conf. Commun. Netw.*, Oct. 2020, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9306073

[3] T. Kenaza, K. Bennaceur, and A. Labed, "An efficient hybrid SVDD/clustering approach for anomaly-based intrusion detection," in *Proc. 33rd Annu. ACM Symp. Appl. Comput.*, Apr. 2018, doi: 10.1145/3167132.3167180.

[4] K. Yang, J. Zhang, Y. Xu, and J. Chao, "DDoS attacks detection with autoencoder," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, Apr. 2020, pp. 1–9, doi: 10.1109/NOMS47738.2020.9110372.

[5] H. Choi, M. Kim, G. Lee, and W. Kim, "Unsupervised learning approach for network intrusion detection system using autoencoders," *J. Supercomput.*, vol. 75, no. 9, pp. 5597–5621, Mar. 2019, doi: 10.1007/s11227-019-02805-w.

[6] X. Chen, C. Cao, and J. Mai, "Network anomaly detection based on deep support vector data description," in *Proc. 5th IEEE Int. Conf. Big Data Anal. (ICBDA)*, May 2020, pp. 251–255, doi: 10.1109/ICBDA49040.2020.9101325.

[7] M. J. Awan, U. Farooq, H. M. A. Babar, A. Yasin, H. Nobanee, M. Hussain, O. Hakeem, and A. M. Zain, "Real-time DDoS attack detection system using big data approach," *Sustainability*, vol. 13, no. 19, p. 10743, Sep. 2021, doi: 10.3390/su131910743.

[8] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, "A new framework for DDoS attack detection and defense in SDN environment," *IEEE Access*, vol. 8, pp. 161908–161919, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9186014

[9] 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, 2017, doi: 10.1109/MC.2017.201. https://doi.org/10.1109/mc.2017.201

[10] M. Gniewkowski, "An overview of DoS and DDoS attack detection techniques," in *Theory and Applications of Dependable Computer Systems*, 2020, pp. 233–241, doi: 10.1007/978-3-030-48256-5_23.

[11] D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yam, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 29–42, Feb. 2005, doi: 10.1109/TNET.2004.842221.

[12] K. Malialis and D. Kudenko, "Distributed response to network intrusions using multiagent reinforcement learning," *Eng. Appl. Artif. Intell.*, vol. 41, pp. 270–284, May 2015, doi: 10.1016/j.engappai.2015.01.013.

[13] Y. Liu, M. Dong, K. Ota, J. Li, and J. Wu, "Deep reinforcement learning based smart mitigation of DDoS flooding in software-defined networks," in *Proc. IEEE 23rd Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Sep. 2018, doi: 10.1109/CAMAD.2018.8514971.

[14] S.-M. Xia, S.-Z. Guo, W. Bai, J.-Y. Qiu, H. Wei, and Z.-S. Pan, "A new smart router-throttling method to mitigate DDoS attacks," *IEEE Access*, vol. 7, pp. 107952–107963, 2019, doi: 10.1109/ACCESS.2019.2930803.

[15] S.-M. Xia, L. Zhang, W. Bai, X.-Y. Zhou, and Z.-S. Pan, "DDoS traffic control using transfer learning DQN with structure information," *IEEE Access*, vol. 7, pp. 81481–81493, 2019, doi: 10.1109/ACCESS.2019.2923993.

[16] K. A. Simpson, S. Rogers, and D. P. Pezaros, "Per-host DDoS mitigation by direct-control reinforcement learning," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 103–117, Mar. 2020, doi: 10.1109/TNSM.2019.2960202.

[17] O. O. Olakanmi and K. O. Odeyemi, "Throttle: An efficient approach to mitigate distributed denial of service attacks on software-defined networks," *Secur. Privacy*, vol. 4, no. 4, Apr. 2021, doi: 10.1002/spy2.158.

[18] S. Chen, C. Shen, C. Wu, and Y. Shen, "DeepThrottle: Deep reinforcement learning for router throttling to defend against DDoS attack in SDN," in *Proc. IEEE Int. Perform., Comput., Commun. Conf. (IPCCC)*, Nov. 2022, pp. 416–417, doi: 10.1109/IPCCC55026.2022.9894298.

[19] T. Cai, T. Jia, S. Adepu, Y. Li, and Z. Yang, "ADAM: An adaptive DDoS attack mitigation scheme in software-defined cyber-physical system," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7802–7813, Jun. 2023, doi: 10.1109/TII.2023.3240586.

[20] M. Aslam, D. Ye, A. Tariq, M. Asad, M. Hanif, D. Ndzi, S. A. Chelloug, M. A. Elaziz, M. A. A. Al-Qaness, and S. F. Jilani, "Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT," *Sensors*, vol. 22, no. 7, p. 2697, Mar. 2022, doi: 10.3390/s22072697.

[21] S. Mergendahl and J. Li, "Rapid: Robust and adaptive detection of distributed denial-of-service traffic from the Internet of Things," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Jun. 2020, pp. 1–9, doi: 10.1109/CNS48642.2020.9162278.

[22] K. Malialis, S. Devlin, and D. Kudenko, "Distributed reinforcement learning for adaptive and robust network intrusion response," *Connection Sci.*, vol. 27, no. 3, pp. 234–252, Apr. 2015, doi: 10.1080/09540091.2015.1031082.

[23] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402. [Online]. Available: https://proceedings.mlr.press/v80/ruff18a/ruff18a.pdf

[24] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," 2017, *arXiv:1706.02690*.

[25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[27] M. S. Frikha, S. M. Gammar, A. Lahmadi, and L. Andrey, "Reinforcement and deep reinforcement learning for wireless Internet of Things: A survey," *Comput. Commun.*, vol. 178, pp. 98–113, Oct. 2021, doi: 10.1016/j.comcom.2021.07.014.

[28] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," *Int. J. Inf. Secur.*, vol. 22, no. 1, pp. 235–247, Nov. 2022, doi: 10.1007/s10207-022-00634-2.

[29] M. S. Rais, R. Boudour, K. Zouaidia, and L. Bougueroua, "Decision making for autonomous vehicles in highway scenarios using harmonic SK deep SARSA," *Int. J. Speech Technol.*, vol. 53, no. 3, pp. 2488–2505, May 2022, doi: 10.1007/s10489-022-03357-y.

[30] D. Zhao, H. Wang, K. Shao, and Y. Zhu, "Deep reinforcement learning with experience replay based on SARSA," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–6, doi: 10.1109/SSCI.2016.7849837.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[32] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. Conf. Appl., Technol., Architectures, Protocols Comput. Commun.*, Aug. 2005, doi: 10.1145/1080091.1080118.

[33] N. Hoque, H. Kashyap, and D. K. Bhattacharyya, "Real-time DDoS attack detection using FPGA," *Comput. Commun.*, vol. 110, pp. 48–58, Sep. 2017, doi: 10.1016/j.comcom.2017.05.015.

[34] S. Behal and K. Kumar, "Detection of DDoS attacks and flash events using novel information theory metrics," *Comput. Netw.*, vol. 116, pp. 96–110, Apr. 2017, doi: 10.1016/j.comnet.2017.02.015.

[35] M. Sachdeva, K. Kumar, and G. Singh, "A comprehensive approach to discriminate DDoS attacks from flash events," *J. Inf. Secur. Appl.*, vol. 26, pp. 8–22, Feb. 2016, doi: 10.1016/j.jisa.2015.11.001.

[36] Y. Liu, J. Yin, J. Cheng, and B. Zhang, "Detecting DDoS attacks using conditional entropy," in *Proc. Int. Conf. Comput. Appl. Syst. Model. (ICCASM)*, vol. 13, Oct. 2010, pp. V13-278–V13-282, doi: 10.1109/ICCASM.2010.5622759.

[37] Y. Gu, K. Li, Z. Guo, and Y. Wang, "Semi-supervised K-means DDoS detection method using hybrid feature selection algorithm," *IEEE Access*, vol. 7, pp. 64351–64365, 2019, doi: 10.1109/ACCESS.2019.2917532.

[38] M. Marvi and A. Arfeen, "Development of an approach for the detection of DDoS attacks based on hybrid feature selection method and clustering algorithm," *SSRN Electron. J.*, Jan. 2020, doi: 10.2139/ssrn.3734895.

[39] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, Jan. 2018, doi: 10.5220/0006639801080116.

[40] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Security Technol. (ICCST)*, Oct. 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8888419

WON SAKONG received the master's degree in business administration from Kyungpook National University, in 2016. He is currently pursuing the Ph.D. degree in industrial engineering with Yonsei University. His primary research interests include anomaly detection, out-of-distribution detection, anomaly object control, machine learning, and reinforcement learning.

WOOJU KIM received the Ph.D. degree in operations research from KAIST, South Korea, in 1994. He is currently a Professor with the School of Industrial Engineering, Yonsei University. His main research interests include natural language processing, reliable knowledge discovery, big data intelligence, machine learning, and artificial intelligence.