mathematics

MDPI

Article

Deep Reinforcement Learning-Based Adversarial Attack and Defense in Industrial Control Systems

Mun-Suk Kim

Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea; msk@sejong.ac.kr

Abstract: Adversarial attacks targeting industrial control systems, such as the Maroochy wastewater system attack and the Stuxnet worm attack, have caused significant damage to related facilities. To enhance the security of industrial control systems, recent research has focused on not only improving the accuracy of intrusion detection systems but also developing techniques to generate adversarial attacks for evaluating the performance of these intrusion detection systems. In this paper, we propose a deep reinforcement learning-based adversarial attack framework designed to perform man-in-the-middle attacks on industrial control systems. Unlike existing adversarial attack methods, our proposed adversarial attack scheme learns to evade detection by the intrusion detection system based on both the impact on the target and the detection results from previous attacks. For performance evaluation, we utilized a dataset collected from the secure water treatment (SWaT) testbed. The simulation results demonstrated that our adversarial attack scheme successfully executed man-in-the-middle attacks while evading detection by the rule-based intrusion detection system, which was defined based on the analysis of the SWaT dataset.

Keywords: industrial control systems; adversarial attacks; intrusion detection systems; deep reinforcement learning

MSC: 68T07

Citation: Kim, M.-S. Deep Reinforcement Learning-Based Adversarial Attack and Defense in Industrial Control Systems. *Mathematics* **2024**, *12*, 3900. https://doi.org/10.3390/math12243900

Academic Editors: Jialing He, Zhi Fang and Chunhai Li

Received: 20 November 2024
Revised: 8 December 2024
Accepted: 10 December 2024
Published: 11 December 2024

1. Introduction

Industrial control systems, which are widely used in critical infrastructure, such as electric power and transportation, have become increasingly interconnected with the evolution of the internet [1]. Consequently, concerns about security threats targeting these industrial control systems are also on the rise [2]. Attacks on industrial control systems, such as the Maroochy wastewater system attack, the Stuxnet worm attack, and the German steel plant attack, have inflicted substantial financial damage on related facilities [3–5]. Recently, a growing number of studies have been dedicated to developing intrusion detection systems aimed at improving the security of industrial control systems [6–10]. Intrusion detection systems continuously monitor the status of the system and provide a warning in the event of an attack, enabling engineers to respond promptly. The accuracy of intrusion detection systems for securing industrial control systems has improved through the use of rule-based approaches [2] and deep learning-based techniques [11].

Along with developing algorithms to improve the accuracy of intrusion detection systems, research has also been conducted on methods for generating adversarial attacks to evaluate the performance of intrusion detection systems [12–18]. Among the various methods for adversarial attacks, this paper focuses on man-in-the-middle attacks that manipulate sensor measurements or actuator configurations in industrial control systems. In industrial control systems, a programmable logic controller (PLC) acts as a control device that receives measurements from sensors and determines the configurations for actuators. To carry out a man-in-the-middle attack, an adversarial attacker positions itself between

the PLC and the sensors/actuators, manipulating the measurements sent from the sensors to the PLC, as well as the configurations sent from the PLC to the actuators. However, in industrial control systems, the measurements of each sensor have distinct physical meanings, resulting in different constraints for each sensor's measurements. Moreover, since industrial control systems contain numerous sensors and actuators, simply manipulating several sensor measurements and actuator configurations without fully considering the correlations between them can easily be detected by intrusion detection systems. In industrial control systems, the correlation coefficient between the measurements or configurations of two different sensors or actuators under various normal operating scenarios represents the degree of correlation between their measurements or configurations. A correlation coefficient closer to 1 or -1 indicates a stronger correlation between the two sensors or actuators.

To address the aforementioned issue, in this paper, we propose a deep reinforcement learning-based adversarial attack scheme for conducting man-in-the-middle attacks in industrial control systems. Unlike existing adversarial attack schemes, our proposed adversarial attack scheme enhances the attack to evade intrusion detection by taking into account feedback on both the impact on the target and the detection results from the intrusion detection system. Therefore, our adversarial attack scheme trains the deep reinforcement learning model with each man-in-the-middle attack, refining it to evade detection by the intrusion detection system. We demonstrate our proposed adversarial attack scheme using the scenario of the secure water treatment (SWaT) system, an industrial control system designed for a modern water purification plant. To evaluate the performance of our adversarial attack scheme, we used a dataset collected over several days from the SWaT testbed provided by iTrust Labs [19,20]. The performance evaluation was conducted using a simplified scenario of the SWaT system, and simulation results demonstrate that our adversarial attack scheme successfully carried out man-in-the-middle attacks while evading detection by the rule-based intrusion detection system, defined based on an analysis of the SWaT dataset.

The remainder of this paper is organized as follows. In Section 2, we describe the SWaT testbed and review previous studies on adversarial attacks against industrial control systems. Section 3 describes our proposed deep reinforcement learning approach for adversarial attacks on the SWaT system. In Section 4, we present and discuss the simulation results of our proposed adversarial attack scheme. Finally, Section 5 concludes this paper.

2. Background

In this section, we provide an overview of the SWaT testbed, which is the target of our proposed adversarial attack scheme [21]. Additionally, we review previous studies on adversarial attacks targeting industrial control systems.

2.1. SWaT Testbed

The SWaT testbed, a scaled-down version of a modern water purification plant, was developed to support research on cybersecurity for industrial control systems [19,20]. The SWaT system can produce up to five gallons of safe drinking water per minute through six distinct, interconnected stages that involve chemical processes such as ultrafiltration, de-chlorination, and reverse osmosis. Each stage is managed by a PLC, which interacts with sensors and actuators via a field-bus network. The PLC receives measurements from the sensors and, based on these measurements, determines and sends the appropriate configuration to the actuators. The SWaT system periodically records sensor measurements and actuator configurations to a historian server. The SWaT dataset from iTrust Labs contains sensor measurements and actuator configurations recorded by the historian server over 11 days, including 7 days of normal operation and 4 days of attack scenarios.

The SWaT system is composed of six stages, and to provide a brief overview of its operation, Figure 1 shows the representative sensors and actuators for the first three stages. In the first stage of the SWaT system, a raw water tank (T101) is connected to a motorized

valve (MV101) that controls the inflow of raw water. The level indicator transmitter (LIT101) measures the water level in the raw water tank T101. An electromagnetic flow transmitter (FIT101) measures the flow rate and sends this measurement to the PLC. Then, pump P101 transfers the water from the raw water tank T101 to the chemical dosing process in the second stage. The second stage handles chemical dosing, with the PLC determining the configurations of actuators P201, P202, and P203 to adjust the dosing. An electromagnetic flow transmitter (FIT201) measures the flow rate of water from the first stage, while the motorized valve (MV201) controls the flow into the ultrafiltration (UF) feed water tank (T301) in the third stage. Finally, in the third stage, ultrafiltration is performed, and the level indicator transmitter (LIT301) measures the water level in the UF feed water Tank T301.

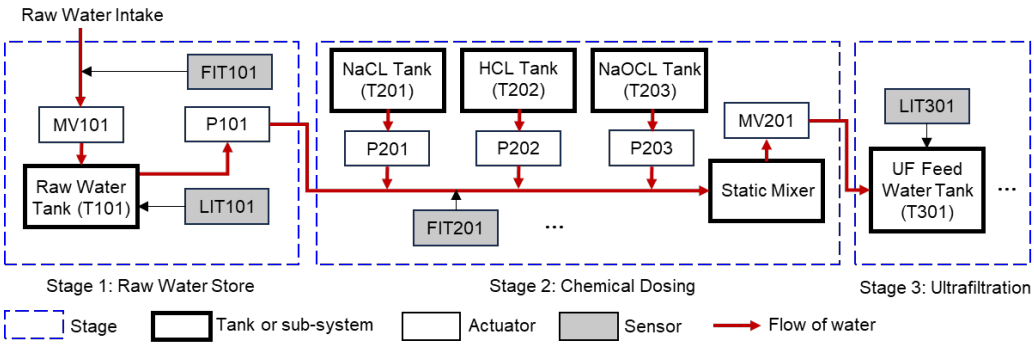

Figure 1. A simplified scenario of the SWaT system.

2.2. Adversarial Attacks on Industrial Control Systems

In industrial control systems, the measurements of each sensor represent different physical meanings, and the normal ranges of these measurements vary. Moreover, the configurations of actuators are determined based on sensor measurements, and these configurations subsequently impact multiple sensors. Therefore, performing adversarial attacks without adequately considering the correlations between sensor measurements and actuator configurations in industrial control systems can lead to easy detection by rule-based intrusion detection systems [12].

The study in [12] proposed a method for conducting real-time manipulation of sensor measurements in industrial control systems by using an autoencoder-based approach to learn the constraints of each sensor's measurements. It also showed that real-time adversarial actions can significantly reduce the accuracy of detection algorithms. The study in [13] proposed a method to conceal adversarial attacks by manipulating the detection outcomes of intrusion detection systems in industrial control systems. Similar to our paper, it also evaluated performance using the SWaT dataset. The study in [14] proposed two strategies—optimal solution attack and generative adversarial network (GAN) attack—aimed at balancing flexibility and data volume through an optimization problem to identify stealthy attacks. The optimal solution attack is more suitable for smaller, flexible samples, while the GAN attack provides a more efficient solution for larger, less flexible samples. The study in [15] examined how adversarial learning attacks can bypass industrial control systems by generating adversarial samples through the Jacobian-based Saliency Map attack. The study in [16] proposed an adversarial attack called the selective and iterative gradient sign method to overcome the limitation that most existing adversarial attacks in industrial scenarios can generate samples misclassified by the intrusion detection system but fail to reach industrial devices. The study in [17] identified and modeled the realistic capabilities and conditions needed for an attacker to carry out feasible and successful adversarial attacks and then applied the model to several common types of adversarial attacks.

3. Proposed Deep Reinforcement Learning for Adversarial Attack

We propose a deep reinforcement learning-based attack scheme for the SWaT system that manipulates sensor measurements before they are transmitted to the PLC, effectively evading detection by intrusion detection systems.

3.1. Overview of Our Proposed Adversarial Attack Scheme

Our proposed adversarial attack scheme is positioned between the PLC and the sensors/actuators within the SWaT system, as shown in Figure 2. The actuator configurations determined by the PLC are manipulated by our proposed adversarial attack scheme before being sent to the actuators. Our proposed adversarial attack scheme also manipulates the sensor measurements before transmitting them to the PLC. The intrusion detection system inspects the sensor measurements received by the PLC to determine whether they are normal or manipulated values.

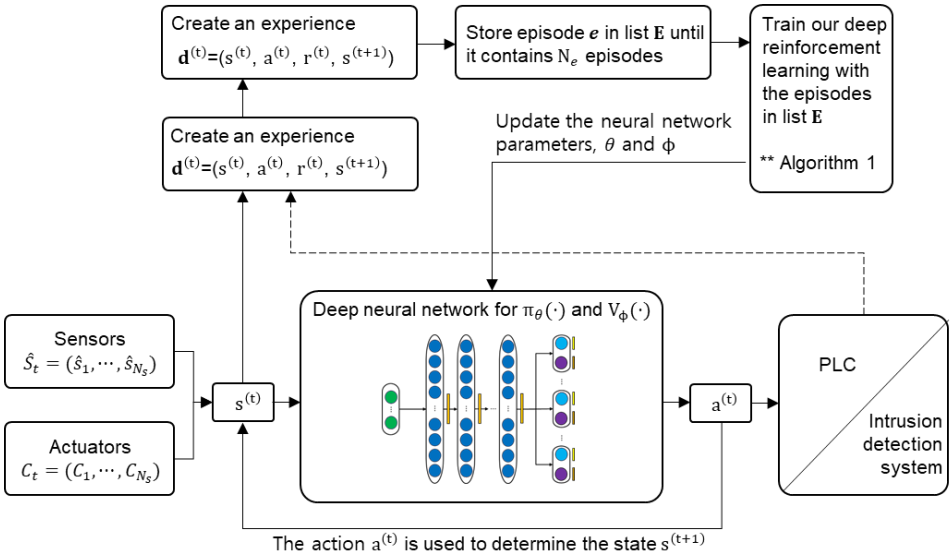

Figure 2. Overview of our deep reinforcement learning process for adversarial attacks.

Our proposed adversarial attack scheme begins by selecting a target to impact and then manipulates the relevant actuator configurations to inflict damage on that target. Subsequently, our proposed adversarial attack scheme manipulates the sensor measurements sent from the sensors to the PLC to prevent the intrusion detection system from detecting the manipulations made to the actuator configurations. To this end, our proposed adversarial attack scheme leverages a deep reinforcement learning model. The state of our deep reinforcement learning model includes the manipulated sensor measurements and actuator configurations determined by the PLC at time step $t-1$, as well as the actual sensor measurements and actuator configurations determined by the PLC at the current time step t. The action of our deep reinforcement learning model comprises manipulated values for the sensor measurements influenced by the manipulated actuator configurations. These manipulated sensor measurements are transmitted to the PLC at the current time t. The reward for the manipulated sensor measurements sent to the PLC at the current time t is set low if the intrusion detection system detects any of the manipulations in the actuator configurations or sensor measurements; otherwise, it is set high.

3.2. Proposed Actor–Critic Deep Learning Model

We adopt a temporal difference-based advantage actor–critic model for our deep reinforcement learning approach. This model requires a policy function $\pi(\cdot)$, which acts as the actor to select actions, and a value function $V(\cdot)$, which serves as the critic to evaluate those actions. Given the multi-dimensional continuous state space of our

reinforcement learning system, we implement both the policy and value functions using deep neural networks. Here, $\pi_\theta(\cdot)$ and $V_\phi(\cdot)$ denote the neural network-based policy and value functions, respectively, with θ and ϕ representing their respective network parameters. In this subsection, we discuss the states, actions, and reward function used in our advantage actor–critic model, as well as the training methods for the policy function $\pi_\theta(\cdot)$ and value function $V_\phi(\cdot)$.

(1) States. The state $s^{(t)}$ at time step t consists of seven feature groups—S_{t-1}, C_{t-1}, I_s, D_{t-1}, $\hat{S}_t$, C_t, I_r, and $\hat{S}_t^r$—and can be expressed as

$$s^{(t)} = (S_{t-1}, C_{t-1}, I_s, D_{t-1}, \hat{S}_t, C_t, I_r, \hat{S}_t^r) \tag{1}$$

where the first feature group, S_{t-1}, includes all sensor measurements transmitted by our proposed adversarial attack scheme to the PLC at time step $t-1$ and the second feature group, C_{t-1}, includes all actuator configurations applied at time step $t-1$. Let N_s and N_c denote the number of sensors and actuators, respectively, in the SWaT system. The first and second feature groups, S_{t-1} and C_{t-1}, can be represented as $S_{t-1} = (s_1, \cdots, s_i, \cdots, s_{N_s})$ and $C_{t-1} = (c_1, \cdots, c_i, \cdots, c_{N_s})$, respectively, where s_i is the measurement of the ith sensor and c_i is the configuration of the ith actuator.

The third feature group, I_s, includes the indices of the sensors that our proposed adversarial attack scheme attempts to manipulate. The fourth feature group, D_{t-1}, includes bit indicators for each sensor in I_s, indicating whether the measurements manipulated by our adversarial attack scheme at time step $t-1$ were detected by the intrusion detection system. Let N_I denote the number of sensor indices included in the third feature group, I_s. Let $I_s = (l_1, \cdots, l_{N_I})$; then, the fourth feature group can be expressed as $D_s = (b_{l_1}, \cdots, b_{l_{N_I}})$, where $b_{l_{N_I}}$ indicates whether the manipulated measurement of the sensor at index l_{N_I} was detected by the intrusion detection system at time step $t-1$. If $b_{l_{N_I}} = 0$, the manipulation was not detected, and if $b_{l_{N_I}} = 1$, it was detected.

The fifth feature group, $\hat{S}_t$, contains all actual sensor measurements transmitted by our proposed adversarial attack scheme to the PLC at time step t, and the sixth feature group, C_t, includes all actuator configurations applied at time step t. The fifth feature group can be represented as $\hat{S}_t = (\hat{s}_1, \cdots, \hat{s}_i, \cdots, \hat{s}_{N_s})$, where $\hat{s}_i$ is the actual measurement taken by the ith sensor.

The seventh feature group, I_r, includes the indices of the sensors that measure the extent of the impact on the target caused by our proposed adversarial attack scheme. The eighth feature group, $\hat{S}_t^r$, includes the actual measurements taken by each sensor in I_r at time step t, unaltered by our adversarial attack scheme. The sensor measurements in the eighth feature group, $\hat{S}_t^r$, are used to obtain the reward for time step t. Let N_r denote the number of sensor indices included in the seventh feature group, I_r. Let $I_r = (l'_1, \cdots, l'_{N_r})$; then, the eighth feature group can be expressed as $\hat{S}_t^r = (\hat{s}_{l'_1}, \cdots, \hat{s}_{l'_{N_r}})$, where $\hat{s}_{l'_{N_r}}$ represents the actual measurement taken by the $l'_{N_r}{}^{\text{th}}$ sensor.

(2) Actions. The action $a^{(t)}$ at time step t comprises the sensor measurements manipulated by our proposed adversarial attack scheme at time step t. The action at time step t consists of the manipulated measurements of the sensors included in the third feature group I_s of the state $s^{(t)}$, and it can be expressed as $a^{(t)} = (s_{l_1}, \cdots, s_{l_{N_I}})$, where $s_{l_{N_I}}$ represents the manipulated measurement of the l_{N_I}th sensor.

Action $a^{(t)}$ is determined by the policy function $\pi_\theta(\cdot)$. We implement our policy function using a multilayer perceptron neural network with one input layer, one output layer, and multiple hidden layers, as illustrated in Figure 3. We use a state from our proposed deep reinforcement learning model as the input to the neural network and employ ReLU as the activation function for each node in the hidden layers. The output layer of the neural network consists of N_I groups, with each group corresponding to a sensor included in the third feature group I_s of the state $s^{(t)}$. Each output group consists of two nodes: the first node outputs the mean value of a normal distribution representing possible

measurements for the associated sensor, while the second node outputs the standard deviation of the normal distribution. For example, the $N_I{}^{\text{th}}$ output group is associated with the sensor at index $l_{N_I} \in I_s$ and outputs the mean $\mu_{l_{N_I}}$ and standard deviation $\sigma_{l_{N_I}}$ of the normal distribution. In the output layer, the tanh function is used as the activation function for the node representing the mean to define the range for each sensor's measurements, while the softplus function is used for the node representing the standard deviation to ensure the output is positive. Then, the manipulated measurement $s_{l_{N_I}}$ for the l_{N_I}th sensor, included in action $a^{(t)}$, is obtained by sampling from a normal distribution with a mean of $\mu_{l_{N_I}}$ and a standard deviation of $\sigma_{l_{N_I}}$.

(3) Reward Function. The reward $r^{(t)}$ at time step t is negative if the intrusion detection system detects manipulation of sensor measurements at that time step. Otherwise, the reward $r^{(t)}$ is inversely proportional to the damage inflicted on the target by our proposed adversarial attack scheme.

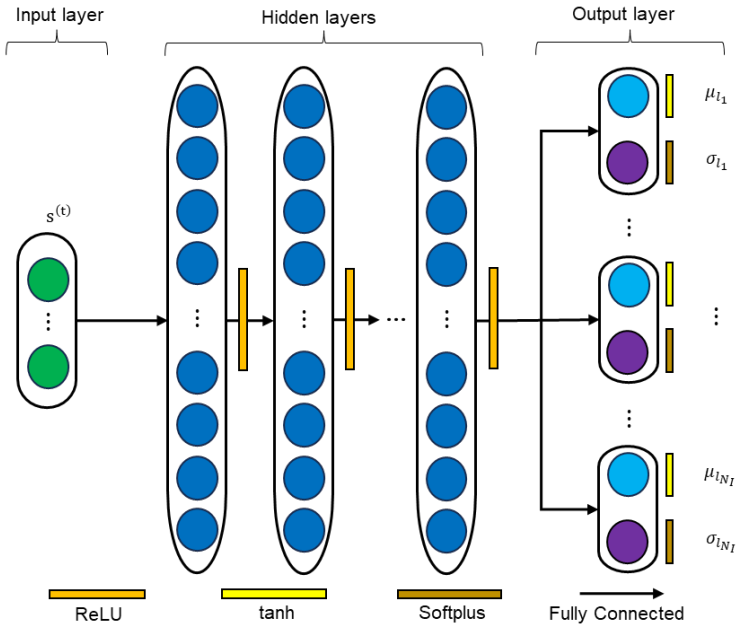

Figure 3. Deep neural network-based policy function.

As mentioned earlier, the seventh feature group, I_r, within state $s^{(t)}$ contains the indices of the sensors that measure the impact of our proposed adversarial attack scheme on the target. Let $d_{l'}$ denote the damage to the target measured by the sensor at index $l' \in I_r$. For example, suppose the target is water tank T101, as shown in Figure 1, and the sensor LIT101 at index $l' \in I_r$ measures the damage from an overflow attack on water tank T101. Let δ_{max} and δ_{min} be the maximum and minimum thresholds, respectively, for the measurements $s_{l'}$ of sensor LIT101, indexed by $l' \in I_r$. Then, the damage $d_{l'}$ can be obtained as follows:

$$d_{l'} = \begin{cases} \log_{10}\left(\frac{\delta_{max}+\delta_{min}}{2} \cdot \frac{1}{\delta_{max}-s_{l'}}\right), & \text{for } s_{l'} < \delta_{max} \\ r_{max}, & \text{for } s_{l'} \leq \delta_{max}, \end{cases} \tag{2}$$

where r_{max} is a constant representing the maximum reward.

In addition, the reward in our proposed adversarial attack scheme is determined based on the detection results of the intrusion detection system. Therefore, for the actual measurements of all sensors in the seventh feature group, I_r, the reward is given as follows:

$$r^{(t)} = \begin{cases} \sum_{l' \in I_r} d_{l'}, & \text{for undetected} \\ r_{min} \cdot N_r, & \text{for detected,} \end{cases} \tag{3}$$

where r_{min} is a negative value representing the minimum reward.

Our proposed adversarial attack scheme is developed to evaluate the performance of various intrusion detection system techniques. Accordingly, it is assumed that our proposed scheme has access to the detection results of the intrusion detection system. As shown in Equation (3), by incorporating the detection results of the intrusion detection system into the reward of our proposed deep reinforcement learning model, our proposed scheme is trained to perform attacks while evading detection by the intrusion detection system.

(4) Training. Our proposed adversarial attack scheme forms an experience at time step t, defined as $\mathbf{d}^{(t)} = (s^{(t)}, a^{(t)}, r^{(t)}, s^{(t+1)})$. The sequence of experiences from time step 1 to t_f is called an episode, assuming that the episode starts at time step 1 and ends at time step t_f. An episode includes the experiences until an attack is detected by the intrusion detection system, and even if the attack is not detected, the number of experiences in the episode is capped at N_d. Then, an episode, denoted by $\mathbf{e}$, can be represented as $\mathbf{e} = (\mathbf{d}^{(1)}, \cdots, \mathbf{d}^{(t)}, \cdots, \mathbf{d}^{(t_f)})$. Similar to the policy function $\pi_\theta(\cdot)$ shown in Figure 3, the value function $V_\phi(\cdot)$ is implemented using a multilayer perceptron neural network consisting of one input layer, one output layer, and multiple hidden layers. The state from our proposed deep reinforcement learning model serves as the input to the neural network for the value function. ReLU is used as the activation function for all nodes in the hidden layers, and the output layer contains a single node that represents the value of the given input state. Whenever N_e episodes are collected, the neural network parameters θ for the policy function $\pi_\theta(\cdot)$ and ϕ for the value function $V_\phi(\cdot)$ are updated based on these N_e episodes.

Algorithm 1 presents the training process of the proposed adversarial attack scheme in pseudocode. The training process of Algorithm 1 is performed every time N_e episodes are collected in the list $\mathbf{E}$. For each experience dataset $\mathbf{d}^{(t)}$ within the collected episodes, we compute the temporal difference target defined as $y_\phi^{(t)}$. Using the temporal difference target $y_\phi^{(t)}$, we then compute the losses L_π and L_V for the policy function $\pi_\theta(\cdot)$ and the value function $V_\phi(\cdot)$, respectively. As mentioned earlier, the action at time step t consists of the manipulated sensor measurements from the third feature group I_s of the state $s^{(t)}$ and can be represented as $a^{(t)} = (s_{l_1}, \cdots, s_{l_{N_I}})$. Let $\pi_\theta(s_{l_{N_I}}|s^{(t)})$ be the probability that $s_{l_{N_I}}$ is included in the action $a^{(t)}$ given that the state is $s^{(t)}$. Then, the probability $\pi_\theta(s_{l_{N_I}}|s^{(t)})$ can be represented as follows:

$$\pi_\theta(s_{l_{N_I}}|s^{(t)}) = \mathrm{N}(s_{l_{N_I}}|\mu_{l_{N_I}}, \sigma_{l_{N_I}}) \tag{4}$$

where $\mathrm{N}(s_{l_{N_I}}|\mu_{l_{N_I}}, \sigma_{l_{N_I}})$ is the probability of obtaining $s_{l_{N_I}}$ by sampling from a normal distribution with mean $\mu_{l_{N_I}}$ and standard deviation $\sigma_{l_{N_I}}$, as determined by the policy function $\pi_\theta(\cdot)$.

Let $\pi_\theta(a^{(t)}|s^{(t)})$ denote the log probability of determining $a^{(t)}$ given the state $s^{(t)}$. Then, it can be calculated as follows:

$$\log \pi_\theta(a^{(t)}|s^{(t)}) = \sum_{l' \in I_s} \log \pi_\theta(s_{l'}|s^{(t)}) \tag{5}$$

The neural network parameters θ for $\pi_\theta(\cdot)$ and ϕ for $V_\phi(\cdot)$ are updated to minimize the respective losses L_π and L_V, respectively. After completing the training process on all N_e episodes in the list **E**, the list **E** is cleared.

Algorithm 1 Training of deep reinforcement learning

E: A list containing N_e episodes
θ: Set of neural network parameters for $\pi_\theta(\cdot)$
ϕ: Set of neural network parameters for $V_\phi(\cdot)$
γ: Discount factor
L_π: Loss function for $\pi_\theta(s^{(t)}, a^{(t)})$
L_V: Loss function for $V_\phi(s^{(t)})$
TD: Temporal difference
for each episode **e** $\in$ **E do**
  for each experience data $\mathbf{d}^{(t)} \in$ **e do**
    if time step $t \neq t_f$ **then**
      $y_\phi^{(t)} \leftarrow r^{(t)} + \gamma \times V_\phi(s^{(t+1)})$ ▷ TD target
    else
      $y_\phi^{(t)} \leftarrow r^{(t)}$ ▷ TD target
    end if
    $\Delta_\phi^{(t)} \leftarrow y_\phi^{(t)} - V_\phi(s^{(t)})$ ▷ TD error
    $L_\pi \leftarrow -\log \pi_\theta(s^{(t)}, a^{(t)}) \times \Delta_\phi^{(t)}$
    $L_V \leftarrow \text{smooth\_l1\_loss}(V_\phi(s^{(t)}), y_\phi^{(t)})$
    Update θ to minimize L_π using Adam Optimizer
    Update ϕ to minimize L_V using Adam Optimizer
  end for
end for

(5) Complexity. The computational complexity of executing our proposed adversarial attack scheme is equivalent to that of generating actions using the deep neural network for the policy function. Let L_H^p denote the number of hidden layers in the deep neural network for the policy function. Then, the computational complexity of executing our proposed adversarial attack scheme can be determined as follows:

$$C_{exe}^p = \mathcal{O}(n_{p,in} \cdot n_1^p) + \sum_{h=1}^{L_H^p - 1} \mathcal{O}(n_{p,h} \cdot n_{p,h+1}) + \mathcal{O}(n_{L_H^p}^p \cdot n_{p,out}) \tag{6}$$

where $\mathcal{O}(\cdot)$ is the bit-O notation, $n_{p,in}$ is the number of nodes in the input layer, $n_{p,out}$ is the number of nodes in the output layer, and $n_{p,h}$ is the number of nodes in the hth hidden layer.

The computational complexity of the training process for our proposed adversarial attack scheme is derived based on Algorithm 1. Algorithm 1 includes executing the value function $V_\phi(\cdot)$, calculating the log probability from Equation (5), and updating the parameters of the neural networks for both the policy function and the value function for each experience data. Let C_{exe}^v denote the computational complexity of executing the value function. Since the structure of the neural network for the value function is similar to that of the policy function, C_{exe}^v can be determined in the same manner as described in Equation (6). The computational complexity of calculating the log probability in Equation (5) is $\mathcal{O}(N_I)$. Because the policy function is constructed as a deep neural network, the computational complexity of updating its network parameters is equivalent to the complexity of the backward pass for the neural network and can be represented as $\mathcal{O}(C_{exe}^p)$. Likewise, the computational complexity for updating the network parameters of the value function is $\mathcal{O}(C_{exe}^v)$. Therefore, for each experience data, the computational complexity of the training process for our proposed adversarial attack scheme is given by $\mathcal{O}(C_{exe}^v + N_I + C_{exe}^p)$.

4. Performance Evaluation

We conducted simulations using the SWaT dataset from iTrust Labs to evaluate the performance of our proposed adversarial attack scheme [19,20]. The SWaT dataset contains periodic sensor measurements and actuator configurations collected over 11 days, including 7 days of normal operation and 4 days of attack scenarios, encompassing a total of 51 sensors and actuators.

Figure 1 illustrates the scenario used for the simulation. We considered four sensors—FIT101, LIT101, FIT201, and LIT301—and six actuators—MV101, P101, P201, P202, P203, and MV201—across three stages of the SWaT system. The sensor measurements are fed into our proposed adversarial attack scheme, which then sends manipulated sensor measurements to the PLC. The PLC provides actuator configurations to our proposed adversarial attack scheme, which subsequently sends manipulated configurations to the actuators. The target of our proposed adversarial attack scheme was the raw water tank T101, aiming to cause an overflow in the tank. The impact of our proposed adversarial attack scheme on the target, raw water tank T101, is measured by the sensor LIT101. Consequently, the seventh feature group, I_r, in the state includes the index of sensor LIT101, and the eighth feature group, $\hat{S}_t^r$, includes the actual measurement from sensor LIT101. Our proposed adversarial attack scheme uses a deep reinforcement learning model at each time step to send manipulated sensor measurements to the PLC. The intrusion detection system then detects any manipulation in the current time step by analyzing the sensor measurements sent to the PLC in the previous time step, along with the actuator configurations determined by the PLC. During the attack conducted by our proposed adversarial attack scheme, the PLC sets all actuators to the 'on' state. However, to cause an overflow in the target, raw water tank T101, our adversarial attack scheme manipulates actuator P101 to be 'off'.

The multilayer perceptron neural network for the policy function, $\pi_\theta(\cdot)$, in the proposed deep reinforcement learning model consists of an input layer with 30 nodes, followed by six hidden layers with 512, 2048, 1024, 256, 64, and 16 nodes, respectively, and an output layer with 8 nodes. The neural network for the value function, $V_\phi(\cdot)$, has an identical input layer and hidden layers, but its output layer contains a single node. The discount factor Γ is set to 0.99, and the learning rate is set to 0.000001.

4.1. Rule-Based Detection by Intrusion Detection System

Our proposed deep reinforcement learning model was trained to evade detection by the intrusion detection system, as described in Section 3.2. For the simulation, we defined an intrusion detection system that uses rule-based detection for four sensors: FIT101, LIT101, FIT201, and LIT301.

By analyzing the SWaT dataset, we determined the normal measurement ranges as follows: FIT101 is 0 to 4.5 m^3/h, LIT101 is 490 to 826 mm, FIT201 is 0 to 2.4 m^3/h, and LIT301 is 730 to 1020 mm. In addition to the normal measurement ranges for the four sensors, the intrusion detection system is defined to detect any violations of the following normal operational rules for each sensor:

- When actuator MV101 is in the 'close' state, the measurement of sensor FIT101 is less than 1 m^3/h.
- When actuator MV101 is in the 'open' state, the measurement of sensor FIT101 is greater than 3 m^3/h.
- When actuator MV101 is 'open' and actuator P101 is 'off', the measurement of sensor LIT101 at the current time step t increases by no more than 1 mm compared to the previous time step $t-1$.
- When actuator MV101 is 'close' and actuator P101 is 'on', the measurement of sensor LIT101 at the current time step t decreases by no more than 0.8 mm compared to the previous time step $t-1$.

- When actuator MV101 is 'open' and actuator P101 is 'on', the difference between the current measurement of sensor LIT101 at time step t and the previous time step $t-1$ is within 0.2 mm.
- When actuator P101 is 'on' and actuator MV201 is 'open', the measurement of sensor LIT301 at the current time step t increases by no more than 0.4 mm compared to the previous time step $t-1$.

Our proposed deep reinforcement learning model performed the training process described in Algorithm 1 every time 100 episodes were collected in list **E**. After the initial training on the 100 lists of **E**, the detection rate of the rule-based intrusion detection system against the adversarial attacks generated by our proposed scheme was 100%. This detection rate decreased to 50.7% with 200 lists, 44.2% with 300 lists, 11.3% with 400 lists, 5% with 500 lists, and finally 0% with 600 lists.

4.2. Discussion of the Simulation Results

In the simulation, the PLC set the configuration of actuators MV101 and MV201 to 'open', and all other actuators (P101, P201, P202, and P203) were set to 'on'. Our proposed adversarial attack scheme manipulated the measurements of sensors FIT101, LIT101, FIT201, and LIT301 over 100 time steps and transmitted them to the PLC. The intrusion detection system then evaluated these manipulated sensor measurements based on the rules outlined in Section 4.1. The attack conducted by our proposed adversarial scheme was not detected by the intrusion detection system. Figures 4–7 illustrate the actual measurements from sensors FIT101, LIT101, FIT201, and LIT301 over 100 time steps, alongside the manipulated measurements generated by our proposed adversarial scheme.

Figure 4 presents the measurements from sensor LIT101, which indicate the impact of the overflow attack on the target raw water tank T101 performed by our proposed adversarial attack scheme. Based on the SWaT dataset, when the configuration of actuator MV101 was 'open', the measurement from sensor LIT101 increased by approximately 0.5 mm at each time step. In contrast, when the configuration of actuator P101 was 'on', the measurement from sensor LIT101 decreased by approximately 0.4 mm at each time step. Under the attack by our proposed adversarial scheme, actuator MV101 was set to 'open' and actuator P101 to 'off'. Consequently, as shown in Figure 4, the actual measurement from sensor LIT101 increased by approximately 0.5 mm per time step. However, since the PLC set actuator MV101 to 'open' and actuator P101 to 'on', our proposed adversarial scheme manipulated the measurements from sensor LIT101, which were sent to the PLC, to increase by approximately 0.1 mm per time step, as shown in Figure 4.

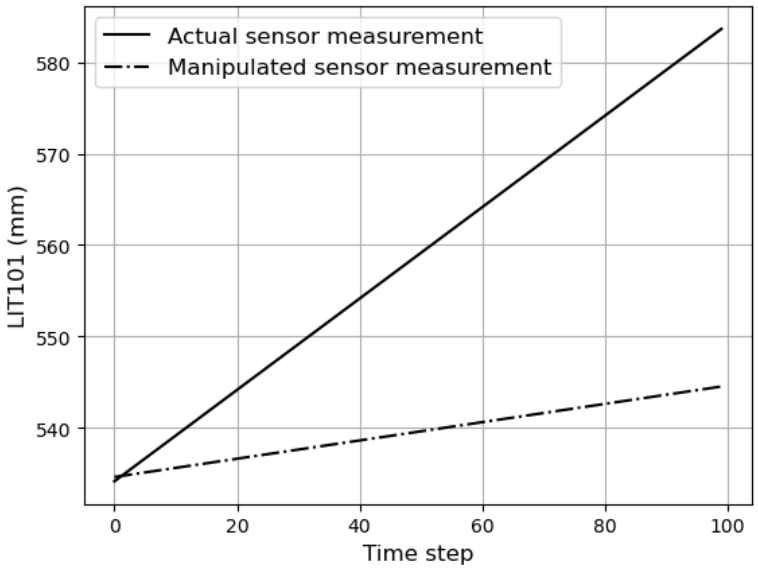

Figure 4. The actual and manipulated measurements of sensor LIT101 over 100 time steps.

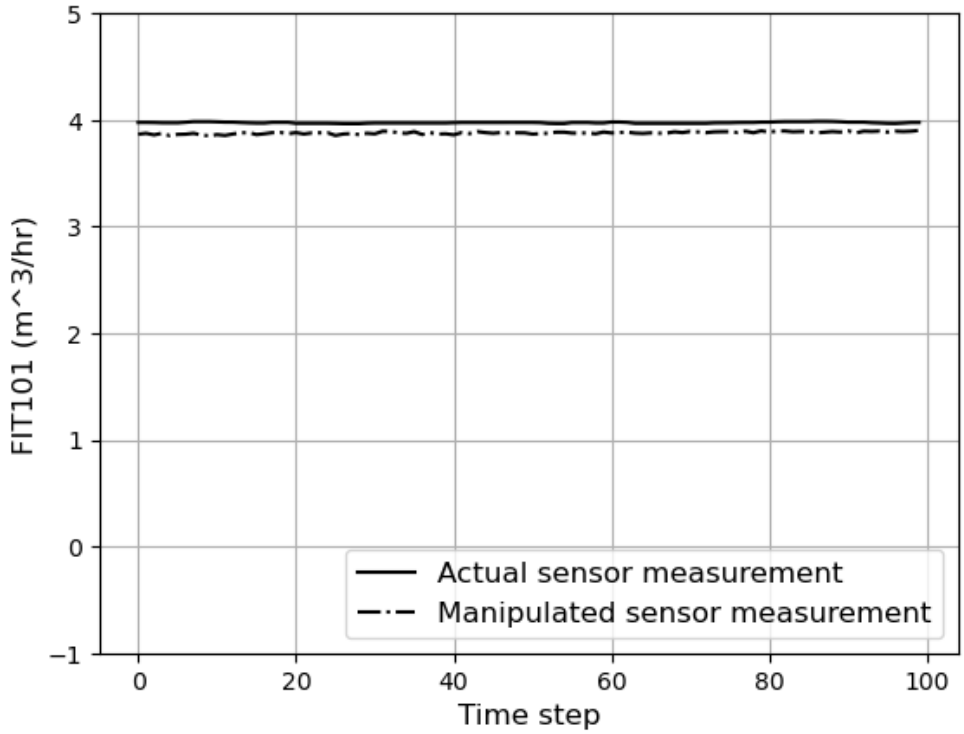

Figure 5. The actual and manipulated measurements of sensor FIT101 over 100 time steps.

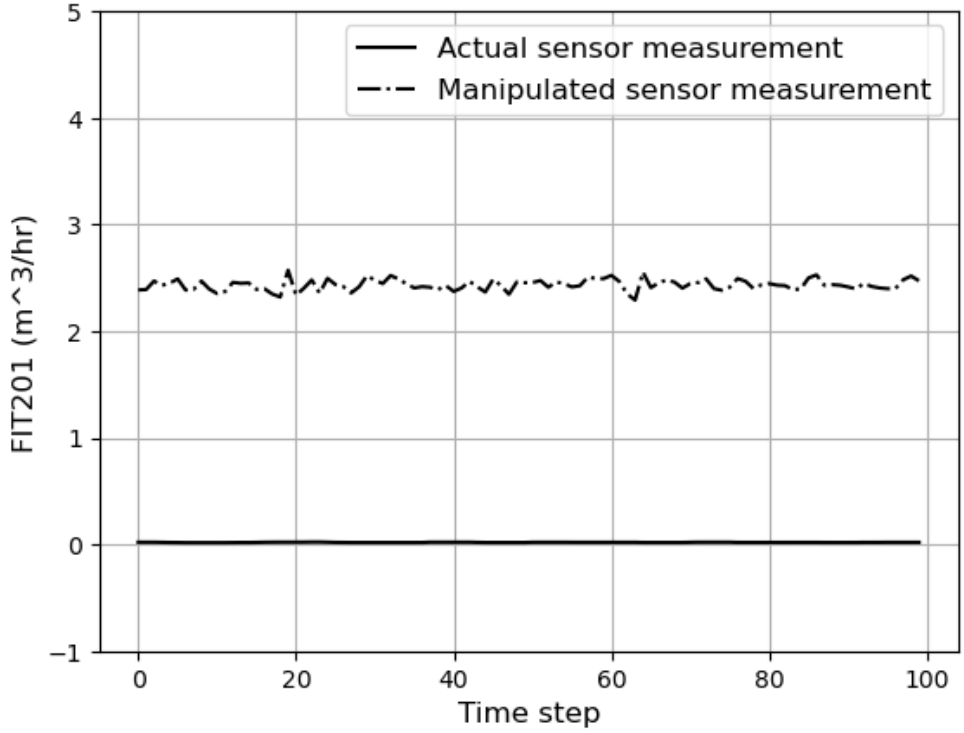

Figure 6. The actual and manipulated measurements of sensor FIT201 over 100 time steps.

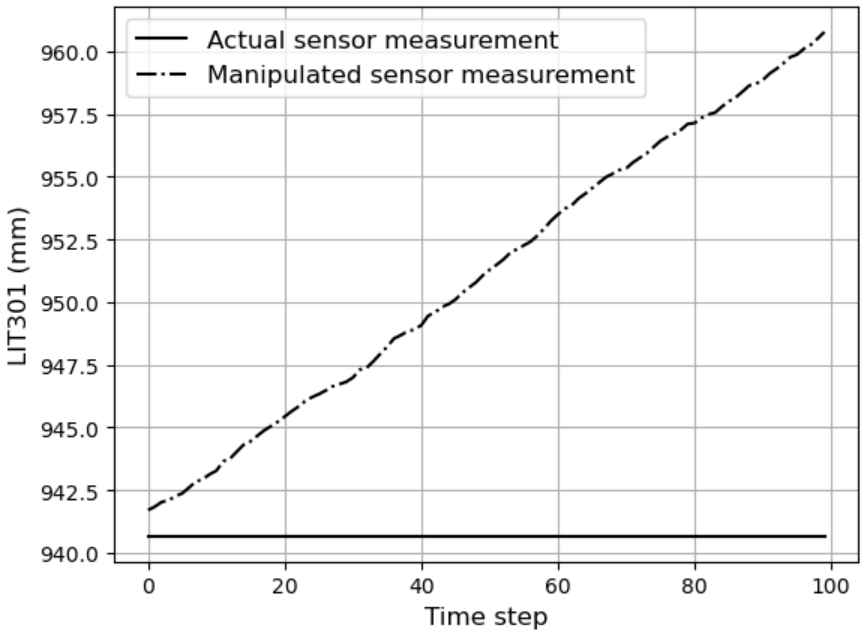

Figure 7. The actual and manipulated measurements of sensor LIT301 over 100 time steps.

The PLC set actuator MV101 to 'open', and our proposed adversarial scheme kept it in the 'open' state without making any changes. As a result, as shown in Figure 5, both the actual and manipulated measurements of sensor FIT101 were approximately 4 m^3/h. Although the PLC set actuator P101 to 'on', our proposed adversarial scheme manipulated it to 'off'. Consequently, sensor FIT201 could not measure the water flow, and as shown in Figure 6, the actual measurements from sensor FIT201 were close to zero. Our proposed adversarial scheme manipulated the measurements of sensor FIT201 to approximately 2.4 to indicate the presence of water flow, as shown in Figure 6.

Lastly, our proposed adversarial attack scheme turned actuator P101 off, preventing sensor FIT201 from measuring water flow. As a result, even though actuator MV201 was configured to open, no water entered the UF feed water tank T301. Consequently, as shown in Figure 7, the actual measurement of sensor LIT301 remained constant for 100 time steps. However, since the PLC determined the configuration of actuator P101 to be on and perceived water as flowing into the UF feed water tank T301 based on the manipulated measurements of sensor FIT201, our proposed adversarial attack scheme manipulated the measurements of sensor LIT301 to increase by approximately 0.2 at each time step, as shown in Figure 7.

5. Conclusions

We proposed an adversarial attack scheme to perform man-in-the-middle attacks, aimed at supporting the performance evaluation of intrusion detection systems for industrial control systems. In industrial control systems, each sensor measurement carries a unique physical meaning, leading to different constraints for each sensor. Additionally, industrial control systems involve numerous sensors and actuators, making it easy for intrusion detection systems to detect any manipulation of sensor measurements or actuator configurations that fail to account for their interdependencies. Our proposed adversarial attack scheme utilizes a deep reinforcement learning approach to learn the correlations between sensor measurements and actuator configurations, allowing it to effectively evade detection by the intrusion detection system. Using simulations based on the SWaT dataset, we showed that our proposed adversarial attack scheme successfully performed man-in-the-middle attacks while evading detection by the rule-based intrusion detection system. In our future work, we plan to expand by constructing datasets for industrial control systems, including steam-turbine power generation, pumped-storage

hydropower generation, and nuclear instrumentation and control systems, in addition to the SWaT system. Furthermore, we intend to implement our proposed scheme using advanced reinforcement learning models, such as Asynchronous Advantage Actor–Critic (A3C) and Soft Actor–Critic (SAC), alongside the Advantage Actor–Critic (A2C) model utilized in this study, to perform comprehensive performance comparisons. To evaluate the effectiveness of our proposed adversarial attack scheme, we plan to employ intrusion detection systems based on machine learning and deep learning technologies.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2021-0-01816) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and this work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2022-00165231).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Notation	Description
$s^{(t)}$	The state at time step t in our proposed reinforcement learning model
$a^{(t)}$	The action at time step t in our proposed reinforcement learning model
$r^{(t)}$	The reward at time step t in our proposed reinforcement learning model
I_s	The third feature group in the state, which includes the indices of the sensors that our proposed adversarial attack scheme aims to manipulate
N_I	The number of sensor indices included in the third feature group, I_s
I_r	The seventh feature group in the state includes the indices of sensors that measure the impact of our adversarial attack scheme on the target
N_r	The number of sensor indices included in the seventh feature group, I_r
$\mathbf{d}^{(t)}$	The experience at time step t, defined as $\mathbf{d}^{(t)} = (s^{(t)}, a^{(t)}, r^{(t)}, s^{(t+1)})$
$\mathbf{e}$	An episode representing the set of experiences determined from time step 1 to t_f
$\mathbf{E}$	A list for storing the N_e episodes used to train our proposed reinforcement learning model
$\pi_\theta(\cdot)$	The neural network-based policy function with θ representing its network parameters
$V_\phi(\cdot)$	The neural network-based value function with ϕ representing its network parameters

References

1. Zeng, G.Q.; Shao, J.M.; Lu, K.D.; Geng, G.G.; Weng, J. Automated federated learning-based adversarial attack and defence in industrial control systems. *IET Cyber Syst. Robot.* **2024**, *6*, 1–19. [CrossRef]
2. Urbina, D.; Giraldo, J.; Tippenhauer, N.O.; Cardenas, A. Attacking fieldbus communications in ICS: Applications to the SWaT testbed. In Proceedings of the Singapore Cyber-Security Conference (SG-CRC), Singapore, 14–15 January 2016; pp. 75–89.
3. Slay, J.; Miller, M. Lessons learned from the maroochy water breach. In Proceedings of the Critical Infrastructure Protection, Hanover, NH, USA, 19–21 March 2007; pp. 73–82.
4. Langner, R. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Secur. Priv.* **2011**, *9*, 49–51. [CrossRef]
5. Lee, R.M.; Assante, M.J.; Conway, T. German steel mill cyber attack. *SANS Indust. Cont. Syst.* **2014**, *30*, 1–15.
6. Schuster, F.; Paul, A.; Rietz, R.; Koenig, H. Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks. In Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7–10 December 2015; pp. 83–90.
7. Liu, W.; Qin, J.; Qu, H. Intrusion detection algorithm of industrial control network based on improved one-class support vector machine. *J. Comput. Appl.* **2018**, *38*, 1360–1365. [CrossRef]
8. Fang, Y.; Li, M.; Wang, P.; Jiang, X.; Zhang, X. Intrusion detection model based on hybrid convolutional neural network and recurrent neural network. *J. Comput. Appl.* **2018**, *38*, 2903–2907.
9. Chu, A.; Lai, Y.; Liu, J. Industrial control intrusion detection approach based on multiclassification GoogLeNet-LSTM model. *Secur. Commun Netw.* **2019**, *1*, 1–11. [CrossRef]

10. Terai, A.; Abe, S.; Kojima, S.; Takano, Y.; Koshijima, I. Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Paris, France, 26–28 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp 132–138.
11. Khan, I.A.; Pi, D.; Abbas, M.Z.; Zia, U.; Hussain, Y.; Soliman, H. Federated-SRUs: A federated-simple-recurrent-units-based IDS for accurate detection of cyber attacks against IoT-augmented industrial control systems. *IEEE Int. Things J.* **2022**, *10*, 8467–8476. [CrossRef]
12. Erba, A. Real-time evasion attacks with physical constraints on deep learning-based anomaly detectors in industrial control systems. *arXiv* **2019**, arXiv:1907.07487.
13. Zizzo, G.; Hankin, C.; Maffeis, S.; Jones, K. Adversarial attacks on time-series intrusion detection for industrial control systems. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020; pp. 899–910.
14. Chen, J.; Gao, X.; Deng, R.; He, Y.; Fang, C.; Cheng, P. Generating adversarial examples against machine learning-based intrusion detector in industrial control systems. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 1810–1825. [CrossRef]
15. Anthi, E.; Williams, L.; Rhode, M.; Burnap, P.; Wedgbury, A. Adversarial attacks on machine learning cybersecurity defences in industrial control systems. *J. Inf. Secur. Appl.* **2021**, *58*, 1–9. [CrossRef]
16. Gomez, A.L.P.; Maimo, L.F.; Celdran, A.H.; Clemente, F.J.G.; Cleary, F. A Crafting adversarial samples for anomaly detectors in industrial control systems. *Procdia Comput. Sci.* **2021**, *184*, 573–580. [CrossRef]
17. Apruzzese, G.; Andreolini, M.; Ferretti, L.; Marchetti, M.; Colajanni, M. Modeling realistic adversarial attacks against network intrusion detection systems. *Dig. Threat. Res. Pract.* **2022**, *3*, 1–19. [CrossRef]
18. Anton, S.D.; Kanoor, S.; Fraunholz, D.; Schotten, H.D. Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set. In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES'18), New York, NY, USA, 27–30 August 2018; pp. 1–9.
19. Mathur, A.P.; Tippenhauer, N.O. SWaT: A water treatment testbed for research and training on ICS security. In Proceedings of the 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater), Vienna, Austria, 11 April 2016; pp. 31–36.
20. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A dataset to support research in the design of secure water treatment systems. In Proceedings of the Critical Information Infrastructures Security, Paris, France, 10–12 October 2016; pp. 88–99.
21. Yoong, C.H.; Palleti, V.R.; Maiti, R.R.; Silva, A.; Poskitt, M.C. Deriving invariant checkers for critical infrastructure using axiomatic design principles. *Cybersecurity* **2021**, *4*, 1–24. [CrossRef]