Journal of
Sensor and Actuator Networks

MDPI

Article

Adversarial Attacks on Heterogeneous Multi-Agent Deep Reinforcement Learning System with Time-Delayed Data Transmission

Neshat Elhami Fard and Rastko R. Selmic *

Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada
* Correspondence: rastko.selmic@concordia.ca

Abstract: This paper studies the gradient-based adversarial attacks on cluster-based, heterogeneous, multi-agent, deep reinforcement learning (MADRL) systems with time-delayed data transmission. The structure of the MADRL system consists of various clusters of agents. The deep Q-network (DQN) architecture presents the first cluster's agent structure. The other clusters are considered as the environment of the first cluster's DQN agent. We introduce two novel observations in data transmission, termed on-time and time-delay observations. The proposed observations are considered when the data transmission channel is idle, and the data is transmitted on time or delayed. By considering the distance between the neighboring agents, we present a novel immediate reward function by appending a distance-based reward to the previously utilized reward to improve the MADRL system performance. We consider three types of gradient-based attacks to investigate the robustness of the proposed system data transmission. Two defense methods are proposed to reduce the effects of the discussed malicious attacks. We have rigorously shown the system performance based on the DQN loss and the team reward for the entire team of agents. Moreover, the effects of the various attacks before and after using defense algorithms are demonstrated. The theoretical results are illustrated and verified with simulation examples.

Keywords: multi-agent system; deep Q-network (DQN); data transmission; gradient-based attack; defense

Citation: Elhami Fard, N.; Selmic, R.R. Adversarial Attacks on Heterogeneous Multi-Agent Deep Reinforcement Learning System with Time-Delayed Data Transmission. *J. Sens. Actuator Netw.* **2022**, *11*, 45. https://doi.org/10.3390/jsan11030045

Academic Editor: Marko Beko

Received: 1 July 2022
Accepted: 4 August 2022
Published: 9 August 2022

1. Introduction

The reinforcement learning (RL) algorithm is the process of learning, mapping states to actions, and ultimately maximizing a reward signal through the interaction of an agent with a specific environment [1,2]. Deep reinforcement learning (DRL) is characterized by a combination of RL and deep learning (DL) algorithms, two subdivisions of machine learning (ML) [3–5]. The DRL's advantage is that it addresses the high-dimensional problems that RL algorithms encounter [4,6,7]. Q-learning, as a type of RL algorithm, learns action values in a specific state [1]. Despite Q-learning's technological advances, it has one major flaw—similarly to dynamic programming, Q-learning can only update data within a two-dimensional array {state × action} [8]. The deep Q-network (DQN) algorithm is introduced, which merges Q-learning, and deep neural networks (NN) [7,9]. To cope with the two-dimensional array problem arising from the Q-learning algorithm, the DQN has been used in a wide range of applications [10–12]. There are two main reasons for using the DQN algorithm instead of other DRL approaches in this work: (i) the stability in performing complicated tasks. The discussed stability is the consequence of utilizing randomly sampled experience replay and a target network; (ii) the ability to predict the Q-value function.

1.1. Contributions

We studied the time-delayed data transmission problem between agents in a cluster-based, heterogeneous, multi-agent deep reinforcement learning (MADRL) system under adversarial attacks. The paper contributions are: (i) in addition to the leaderless multi-agent system (MAS), we proposed a leader-follower MAS, such that the preassigned leader in each cluster communicates with the leader of other clusters as well as the agents of its own cluster; (ii) we considered two novel observations in data transmission, called on-time and time-delay observations, and we investigated their effects on the DQN loss and team reward; (iii) we proposed a novel immediate reward function that considers the package length, packet header size, and distance between neighboring agents to improve the MAS performance in terms of approximated cumulative team discounted reward during time-delayed data transmission; (iv) we considered the fast gradient sign method (FGSM), fast gradient method (FGM), and basic iterative method (BIM) adversaries (gradient-based attacks) to attack the DQN algorithm. Then we investigated the effects of such attacks on MAS performance and time-delayed data transmission; (v) we introduced two defense algorithms against the performed adversarial attacks. In the proposed defense methods, the DQN agent's deep NN learns from a state that produces the maximum perturbation value and uses its negative feedback to improve the system performance during an adversarial attack.

1.2. Related Research

DQN algorithm has been used in data transmission between multiple agents. The transfer learning (TL) approach is combined with the DQN algorithm, and a multi-source transfer double DQN (MTDDQN) is introduced in [13]. The MTDDQN is based on actor learning and enables the collection, summarization, and transfer of action knowledge by the RL agent between multiple agents [13]. Compared to [13], the current paper uses one DQN agent in a cluster-based, heterogeneous, MAS for on-time and time-delayed data transmission.

Data transmission in MAS has been investigated in various scenarios and for linear and nonlinear systems [14]. For instance, periodic event-triggered output regulation for linear MAS by considering a leader-follower topology is proposed in [15]. An adaptive event-triggered consensus control of linear MAS with directed leader-follower topology in the presence of a cyber-attack that affects the control input without modification in communication topology is developed in [16].

A dynamic event-triggered asynchronous control integrating fuzzy models with directed topology is presented in [17]. A new adaptive event-triggered leaderless consensus control of nonlinear MAS, including directed topology, can be found in [18]. Moreover, some researches address data exchanges between linear and nonlinear systems as a heterogeneous MAS, e.g., a leaderless and a leader-follower consensus of heterogeneous second-order MAS on time scales using an asynchronous impulsive approach, is presented in [19]. The previous studies have shown that there is little research on data transmission within homogeneous or heterogeneous MADRL systems, and the majority of the research has been focused on linear and nonlinear MAS. A heterogeneous MAS based on carrier-sense multiple access (CSMA) that utilizes DRL algorithm in data transmission, termed carrier-sense deep reinforcement learning multiple access (CS-DLMA), is introduced in [20]. The CS-DLMA uses α-fairness objective to measure system performance. Inspired by [20] and using CS-DLMA, we study time-delayed data transmission between agents of a leaderless MADRL system. The same study is carried out for a leader-follower MADRL system. Note that CSMA is an access control protocol in which an agent in the network checks the state of the data channel for data transmission.

Cyber-attacks can happen to any system, especially those that transmit data. Various adversarial attacks pose a threat to ML algorithms and DL systems [21,22]. The ML algorithms are misled by adversarial attacks that manipulate input data to undermine algorithm performance, access the ML model, and modify model behavior [23,24]. Therefore,

it is important to study the effects of various attacks on ML algorithms [25]. This paper uses three types of gradient-based adversarial attacks, termed FGSM [26], FGM [27,28], and BIM [27,29] to investigate their effects on the DQN algorithm, and consequently, MADRL system performance. Paper [20] is devoid of any information on cyber-attack on the system. The authors of [30] have examined the adverse effects of FGSM attack on DRL-based traffic signal control for a single-intersection and multiple intersection cases; however, its effects are not investigated on sending and receiving data. Hence, the FGSM attack plus FGM and BIM attacks, that try to fool the NN, are considered in this paper to check their impacts on the data transmission robustness. The authors of [31] have used the discussed attacks to target the observation set provided by the RL algorithm environment; as a result, we have applied the three types of adversarial attacks to target the produced environmental state of the DRL algorithm.

There are various defense techniques for ML algorithms, and these adversarial defense methods are used to improve the robustness of a designed model [32]. Among all the presented methods [33–36], the best defense procedure occurs when the adversarial examples are fed to the NN training process [28]. In this regard, we use the worst perturbation as the input of the NN to train the model and reach the robustness against malicious attacks.

This paper scrutinizes adversarial attack issues facing ML algorithms and studies the time-delayed data transmission robustness due to three types of gradient-based malicious attacks– FGSM, FGM, and BIM adversaries– between agents of a cluster-based, heterogeneous MADRL system. This study shows how the leaderless or leader-follower MAS performs due to time-delayed data transmission. After an attack on a leader-follower MADRL system, this paper presents two adversarial attack defense approaches against gradient-based attacks. This paper does not study the detectability of attacks, but how to reverse the attacks.

A potential application of this research is to use the proposed system in smart grids to make the grid more reliable, secure, and efficient. Moreover, by converting static agents to mobile agents and considering relevant contributions to the novel architecture, this system can be used for data transmission between the agents of all types of multi-agent autonomous vehicles, e.g., multi-rescue robots.

After giving a brief introduction on various aspects of this research in Section 1, the background is explained in Section 2. The methodology of the proposed approach is offered in Section 3. Results and discussion on the introduced system and its behavior during adversarial attacks are provided in Section 4. The conclusion and future works are presented in Section 5. A preliminary conference version of this manuscript was submitted and accepted previously [37]. New results on data transmission robustness of the proposed leader-follower MAS due to adversarial attacks are presented for additional comparison and evaluation in this paper.

2. Background

Decision-making is based on the information received from the environment by an RL or DRL agent. It is considered that the finite Markov decision process (MDP) represents the dynamics of the environment for decision-making. The 5-tuple $M = \langle s, a, T, R, \gamma \rangle$ presents an MDP for an RL and DRL system, where s is a finite set of environmental states, and a is a finite action set. Moreover, $T(s_t, a_t, s_{t+1}) \rightarrow [0, 1]$ is the state-transition probability function that agent takes action a_t in the state s_t, and is transferred to the state s_{t+1} to do the next action. Further, $R(s_t, a_t, s_{t+1}) \triangleq \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \rightarrow \mathbb{R}^n$ is a cumulative reward function, where r_{t+k+1} shows the immediate reward, with discount factor γ that is the trade-off between an immediate reward and potential future reward.

In the leaderless MAS scenario, all agents communicate with their cluster-mates, as well as agents of other clusters. In the leader-follower MAS scenario, in each cluster, only the preassigned leader communicates with the other agents in the same cluster as well as leaders of different clusters. Thus, data transmission occurs between the leader and the followers of one cluster as well as leaders of clusters. The leaderless and leader-

follower MAS are considered as the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of all agents, and $\mathcal{E} \subseteq \{(i,j)|i \in \mathcal{V}, j \in \mathcal{V}\}$ is the set of all communication links between agents. The agents i and j communicate if and only if $(i,j) \in \mathcal{E}$ [38].

3. Methodology

In this Section, the leaderless and leader-follower topologies are introduced. Then, the components of the DQN algorithm (observation, action, state, and reward) are explained. Afterward, the DQN loss for on-time and time-delayed data transmission is justified. Three types of adversarial attacks to target the proposed leader-follower MAS (state of the DQN agent) are extended and explained. Finally, two defense methods against performed adversarial attacks are introduced.

3.1. Leaderless and Leader-Follower Topologies

A generic illustration of the MADRL system topology including N static, heterogeneous agents, and P clusters is shown in Figure 1. The leaderless and leader-follower MAS scenarios can be envisioned from the presented topology in Figure 1. The goal of each static agent in this topology is to transfer data with the maximum average reward.

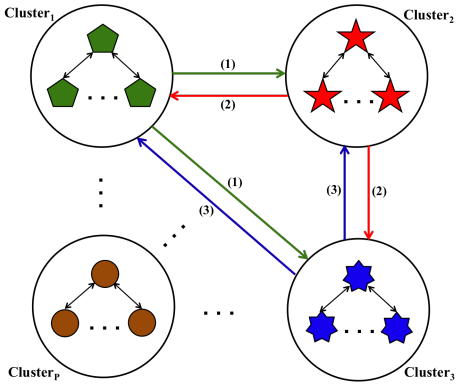


Figure 1. An illustration of the MADRL system topology, including N static, heterogeneous agents, and P clusters.

3.2. Observation

In this paper, observation describes the state of the data transmission channels that are either idle or busy [20]. If the channel between each pair of agents is busy at time step t, no data can be transmitted at time step $t+1$ due to data transition by another agent. However, if this channel is idle, data can be transmitted. The transmitted data either reaches its destination successfully or collides along the way, gets corrupted, and does not reach the goal. Therefore, the defined observation set by [20] is $o_t = \{$busy, idle, successful, collided$\}$.

We propose a modified observation set to use in our MADRL system. We add on-time and time-delayed arrival states to the observation set. As in [20], it is first checked that the data channel between each pair of agents is either idle or busy. If the channel is idle, the data can be transferred successfully on-time, successfully with time-delay, or collided. Therefore, in the introduced scenarios of this paper, the novel observation set $o_t = \{$busy, idle, on-time, time-delay, collided$\}$ is proposed.

The lengths of the transferred packets in the network are different and belong to the set of $R_c \in \{1, 2, \ldots, R_{c\max}\}$. When the observation is on-time, it means that each agent in the network transmits the packet at the next R_c mini-slot, with the action time duration in the set of $T_d(a_t) \in \{1, 2, \ldots, R_{c\max}\}$. With the time-delay observation, each agent in the network transmits the packet at the next R_c mini-slot, with the action time duration in the set of $T'_d(a_t) \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}$. In both on-time and time-delay observations, when an agent transmits a packet in a data transmission channel, no other agent sends the data at that specified channel to avoid a collision. In the following, $T_d(a_t)$ and $T'_d(a_t)$ are abbreviated as T_d and T'_d, respectively. When the observation collides, it means that the

agent transmits the packet at the next R_c mini-slot; however, another agent transmits data in at least one of the R_c mini-slots. Note that each mini-slot is a required time to perform CSMA. In this paper, each mini-slot is considered a time step.

3.3. Action

Action is one of the significant components of RL and DRL algorithms [39]. In general, the agent receives the corresponding state from the environment at time step t and performs the appropriate action accordingly. Due to the quality of the performed action at time step t, the agent receives the reward associated with that action at time step $t+1$. According to the observation set in this paper, the agent first checks whether the data channel is idle or busy. This stage should be done at less than one mini-slot. The performed actions in this paper are based on [20], as follows:

- *No Selection:* If the channel is busy at time step t (checked at less than one mini-slot), the DQN agent does not take any action at the next time step. Hence, $a_{t+1} = 0$.
- *Uniform Selection with Probability ε:* If the channel is idle at time step t, the DQN agent chooses an action (transfer or not to transfer a packet) at time step $t+1$. If the agent at the next R_c mini-slot transmits packets with the length of R_c, then the action at time step $t+1$ is $a_{t+1} = R_c$, where $R_c \in \{0, 1, 2, \ldots, R_{c\,\max}\}$. This action selection method is a uniform random selection with probability ε (exploration) using ε-greedy algorithm.
- *Non-uniform Selection with Probability $1-\varepsilon$:* If the channel is idle at time step t, an action to transfer or not to transfer a packet at time step $t+1$ can be chosen by the DQN agent. According to the conventional ε-greedy DQN algorithm, the action will be the maximum Q-value $\{Q(s, a; \theta) | a \in \mathbb{A}\}$, where Q is a parametric function including state s, action a, and parameter θ as a vector, including the weights in the NN. Moreover, $\mathbb{A}$ is the set of actions. Therefore, the action at time step $t+1$ is

$$a_{t+1} = \underset{a_t \in \{0,1,2,\ldots,R_{c\max}\}}{\arg\max} \sum_{i=1}^{N} Q^i\left(s_{t+1}, a_t; \theta^-\right), \tag{1}$$

  where θ^- denotes the target Q-value weight. This action selection method is a non-uniform selection with probability $1-\varepsilon$ (exploitation) using ε-greedy algorithm.

Note that the ε-greedy algorithm is a widely used policy-based exploration approach in RL and DRL algorithms [1,40].

3.4. State

In this paper, there are two types of states; channel state c^s_{t+1}, and DQN algorithm state s_{t+1} [20]. The DQN algorithm state used in the DRL process is based on the channel state. The channel state at time step $t+1$ is $c^s_{t+1} \triangleq (a_t, o_t)$. Hence, the DQN algorithm state at time step $t+1$ is $s_{t+1} \triangleq [c^s_{t-L+2}, ..., c^s_t, c^s_{t+1}]$, where L is the state history length that describes the number of past time steps to be tracked by the DQN algorithm.

3.5. Reward

Selecting a reward function is usually based on what the RL and DRL systems are supposed to do [41]. In this paper, first, the selection of the reward function depends on the data specifications, including the length of a sent package and the packet header size [20]. The larger the packet header size, the more problems it causes in sending data in the channel (time-delay data transmission or data collision). Therefore, the oversize packet header causes the DQN agent to receive less reward. Afterward, we propose another component to obtain each agent's more precise average reward. Distances between agents may be significant for sending and receiving data and maintaining distances between clusters. Additionally, the distance between two mobile agents is crucial to avoid collisions (the study of mobile agents is beyond the scope of this paper). Hence, we consider the

length of a sent package, the packet's header size, and the distance between a couple of agents in determining the immediate reward function.

In this paper, the utilized rewards are:

- At time step t, if the transferred package does not reach the destination (another agent from another cluster or the agent from the same cluster) successfully, and collides on the way, the immediate reward for i-th agent at time step $t+1$ is $r^i_{t+1} = 0$.
- Using the observation set of [20], if the data packet successfully transferred by each agent in the network, the immediate reward for i-th agent at time step $t+1$ is

$$r^i_{t+1} = R^i_c - H^i_p, \tag{2}$$

where H_p is the packet's header and is a part of each mini-slot.
- By proposing on-time and time-delay observations, we append other components to the immediate reward and present a new immediate reward for each agent. Considering constant $\kappa \in \mathbb{R}^+_*$ and $\kappa \in [1, \infty)$, the new immediate reward for i-th agent is introduced by

$$r^i_{t+1} = R^i_c - (\kappa \cdot H^i_p), \tag{3}$$

where $\kappa = 1$ if the data packet is transferred by each agent in the network successfully and on-time, and $\kappa > 1$ if the data packet is transferred by each agent in the network successfully and with time-delay.
- Considering the distance between agents who transmit data to each other, we propose another type of immediate reward for i-th agent using the combined immediate reward function [41]. If the data packet successfully transferred by i-th agent to j-th agent (on-time), we propose the novel distance-based immediate reward for i-th agent at time step $t+1$ as

$$r^i_{t+1} = R^i_c - H^i_p - \sum_{j=1}^{N-1} Cr^{ij}_{t+1}. \tag{4}$$

If the data packet transferred by i-th agent to j-th agent successfully and time-delayed, the distance-based immediate reward for i-th agent at time step $t+1$ when $\kappa > 1$ is given by

$$r^i_{t+1} = R^i_c - (\kappa \cdot H^i_p) - \sum_{j=1}^{N-1} Cr^{ij}_{t+1}, \tag{5}$$

where $Cr^{ij}_{t+1} = \max(Mr^{ij}_{t+1}, Er^{ij}_{t+1}, \check{C}r^{ij}_{t+1}, Nr^{ij}_{t+1})$ is the combined immediate reward function such that Mr^{ij}_{t+1}, Er^{ij}_{t+1}, $\check{C}r^{ij}_{t+1}$, and Nr^{ij}_{t+1} are Manhattan, Euclidean, Chebyshev, and $n-$norm immediate reward functions, respectively [41]. In this paper, the combined immediate reward function is obtained based on positions (x^i, y^i) and (x^j, y^j) of i-th and j-th agents, respectively. Nevertheless, the original combined immediate reward function, defined by [41], is based on the current position and the desired position of i-th agent in the MARL system.

Typically the formal definition of the DQN target Q-value of a state-action pair (s_t, a_t) is provided by

$$\mathrm{Tar}_Q = R(s_t, a_t, s_{t+1}) + \gamma \max_{a_{t+1}} Q\left(s_{t+1}, a_{t+1}; \theta^-\right), \tag{6}$$

using the cumulative reward function $R(s_t, a_t, s_{t+1})$, discount factor $\gamma \in (0,1)$, and target Q-value weight θ^-. According to [20], the DQN target Q-value of a state-action pair (s_t, a_t) is given by

$$\begin{aligned} \mathrm{Tar}_Q &= \frac{r_{t+1} \cdot (1+\gamma+\ldots+\gamma^{T_d-1})}{T_d} + \gamma^{T_d} \max_{a_{t+1}} Q\big(s_{t+1}, a_{t+1}; \theta^-\big) \\ &= \frac{r_{t+1} \cdot (1+\gamma+\ldots+\gamma^{T_d-1})}{T_d} \cdot \frac{1-\gamma}{1-\gamma} + \gamma^{T_d} \max_{a_{t+1}} Q\big(s_{t+1}, a_{t+1}; \theta^-\big) \\ &= \frac{r_{t+1}}{T_d} \cdot \frac{1-\gamma^{T_d}}{1-\gamma} + \gamma^{T_d} \max_{a_{t+1}} Q\big(s_{t+1}, a_{t+1}; \theta^-\big). \end{aligned} \tag{7}$$

The Q-value function $Q(s_t, a_t; \theta)$ is defined by

$$Q(s_t, a_t; \theta) = \mathbb{E}_{s_{t+1} \sim T(s_t, a_t, s_{t+1})} \Big[R(s_t, a_t, s_{t+1}) + \gamma \max_{a_{t+1}} Q\big(s_{t+1}, a_{t+1}; \theta^-\big) \Big], \tag{8}$$

using the state-transition probability function $T(s_t, a_t, s_{t+1})$ that defines the conditional probabilities between the states. Furthermore, θ is the Q-value weight. According to the gradient method, the parameter θ is updated as

$$\theta_{t+1} \leftarrow \theta_t + \alpha \big(\mathrm{Tar}_Q - Q(s_t, a_t; \theta)\big) \nabla_\theta Q(s_t, a_t; \theta), \tag{9}$$

where α is the learning rate.

Both immediate reward choices (2)–(5) and the final Q-value function, obtained by updating the parameter θ of (9), are connected to the actual data transmission by considering two options: (i) characteristics of transferred packets including the package length and packet header size; (ii) specifications of neighboring agents' distances in such a way that unregulated distance between agents delays data transmission.

Remark 1. *Learning process in DQN algorithm is more stable than Q-learning process since the update rule introduces a delay between the time when Q-value $Q(s_t, a_t; \theta)$ is updated and the time when target network $Q(s_{t+1}, a_{t+1}; \theta^-)$ is updated [42]. Therefore, the target network remains unchanged due to the time-delay.*

Theorem 1. *Suppose that the MAS including N agents is modeled by a graph $\mathcal{G}$, and the learning process is performed by the DQN algorithm. If the i-th agent transfers data to the j-th agent successfully and with time-delay then the average approximated cumulative team discounted reward of a state-action pair (s_t, a_t) satisfies the following*

$$\frac{1}{N} \sum_{i=1}^{N} \underset{\kappa>1}{Q^i(s_t, a_t; \theta)} < \frac{1}{N} \sum_{i=1}^{N} \underset{\kappa=1}{Q^i(s_t, a_t; \theta)}. \tag{10}$$

Proof. To avoid time-delay, we consider the action time duration $T_d \in \{1, 2, \ldots, R_{c\max}\}$. With the time-delay, we assume that the action time duration is unbounded above and $T'_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}$. Therefore, the time-delay occurs when $T'_d \geq R_{c\max}+1$. We set the constant $\kappa \in \mathbb{R}^+_*$ in such a way that $\kappa \in [1, \infty)$. From (3) it follows

$$\sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} \frac{R^i_c - (\kappa \cdot H^i_p)}{T'^i_d} \quad < \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} \frac{R^i_c - H^i_p}{T^i_d}. \tag{11}$$

By considering a specific value of $\gamma \in (0,1)$, we have

$$\sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} \frac{R^i_c - (\kappa \cdot H^i_p)}{T'^i_d} \cdot \frac{1-\gamma^{T'^i_d}}{1-\gamma} < \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} \frac{R^i_c - H^i_p}{T^i_d} \cdot \frac{1-\gamma^{T^i_d}}{1-\gamma}. \tag{12}$$

By considering a specific value of $\gamma \in (0,1)$, for high values of T'_d the following is given

$$\gamma^{T'_d} \underset{T'_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}{\to} 0, \tag{13}$$

while for $T_d \in \{1, 2, \ldots, R_{c\max}\}$ we have

$$0 < \gamma^{T_d} \underset{T_d \in \{1,2,\ldots,R_{c\max}\}}{} < 1. \tag{14}$$

Using (13) and (14), for i-th agent the following is achieved

$$\sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} \gamma^{T'^i_d} < \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} \gamma^{T^i_d}. \tag{15}$$

According to Remark 1, by considering the maximum target network among the possible actions that can be taken from the next state and using (15), the following is valid

$$\sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} \gamma^{T'^i_d} \max_{a_{t+1}} Q^i\left(s_{t+1}, a_{t+1}; \theta^-\right) < \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} \gamma^{T^i_d} \max_{a_{t+1}} Q^i\left(s_{t+1}, a_{t+1}; \theta^-\right). \tag{16}$$

Utilizing (12) and (16) yields

$$\begin{aligned} &\sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} \left(\frac{R^i_c - (\kappa \cdot H^i_p)}{T'^i_d} \cdot \frac{1-\gamma^{T'^i_d}}{1-\gamma} + \gamma^{T'^i_d} \max_{a_{t+1}} Q^i\left(s_{t+1}, a_{t+1}; \theta^-\right) \right) \\ &< \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} \left(\frac{R^i_c - H^i_p}{T^i_d} \cdot \frac{1-\gamma^{T^i_d}}{1-\gamma} + \gamma^{T^i_d} \max_{a_{t+1}} Q^i\left(s_{t+1}, a_{t+1}; \theta^-\right) \right). \end{aligned} \tag{17}$$

Therefore,

$$\sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} \mathrm{Tar}_{Q^i} < \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} \mathrm{Tar}_{Q^i}. \tag{18}$$

To achieve the least amount of training loss $\ell(\theta, s, a)$, the difference between the target Q-value and predicted Q-value should converge to zero. Hence, the below equation can be considered for i-th agent,

$$\lim_{t \to t_0} Q^i(s_t, a_t; \theta) = \mathrm{Tar}_{Q^i}. \tag{19}$$

Substituting (19) in (18) yields

$$\sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} \lim_{t \to t_0} Q^i(s_t, a_t; \theta) < \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} \lim_{t \to t_0} Q^i(s_t, a_t; \theta). \tag{20}$$

According to the monotone convergence condition, the following is given

$$\sum_{i=1}^{N} \lim_{t \to t_0} Q^i(s_t, a_t; \theta) = \lim_{t \to t_0} \sum_{i=1}^{N} Q^i(s_t, a_t; \theta). \tag{21}$$

By considering (21), the inequality (20) is modified as below

$$\lim_{t \to t_0} \sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} Q^i(s_t, a_t; \theta) \quad < \lim_{t \to t_0} \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} Q^i(s_t, a_t; \theta) \tag{22}$$

$$\sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} Q^i(s_{t_0}, a_{t_0}; \theta) \quad < \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} Q^i(s_{t_0}, a_{t_0}; \theta). \tag{23}$$

By averaging each side of inequality (23), and redistribute each side of the inequality to time t, the following is given

$$\frac{1}{N} \sum_{\substack{i=1 \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} Q^i(s_t, a_t; \theta) \quad < \frac{1}{N} \sum_{\substack{i=1 \\ T^i_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} Q^i(s_t, a_t; \theta). \tag{24}$$

$$\begin{cases} T^i_d \in \{1, 2, \ldots, R_{c\max}\} & \text{if} \quad \kappa = 1, \\ T'^i_d \in \{R_{c\max}+1, R_{c\max}+2, \ldots\} & \text{if} \quad \kappa > 1. \end{cases} \tag{25}$$

Therefore, by considering inequality (24) and condition (25) for MAS, including N agents, the inequality (10) is proven as follows

$$\frac{1}{N} \sum_{\substack{i=1 \\ \kappa>1}}^{N} Q^i(s_t, a_t; \theta) < \frac{1}{N} \sum_{\substack{i=1 \\ \kappa=1}}^{N} Q^i(s_t, a_t; \theta). \tag{26}$$

□

Theorem 2. *Suppose that graph $\mathcal{G}$ as a MAS includes N agents. The distance between i-th agent and j-th agent is d^{ij} in such a way that $\xi \leq d^{ij} \leq \lambda$, where ξ and $\lambda \in \mathbb{R}^+_*$ are constant values and $\xi \neq \lambda$. Using the results of [41], it is expected that by considering $\xi \leq d^{ij} \leq \lambda$ the distance-based immediate reward (5) improves the DQN learning process and compensates for the negative effect of the time-delayed data transmission. Therefore, the average approximated cumulative team discounted reward of a state-action pair (s_t, a_t) satisfies the following*

$$\frac{1}{N} \sum_{\substack{i=1 \\ \kappa>1}}^{N} Q^i(s_t, a_t; \theta) \geq \frac{1}{N} \sum_{\substack{i=1 \\ \kappa=1}}^{N} Q^i(s_t, a_t; \theta). \tag{27}$$

Proof. In the case of time-delayed data transmission, distance-based immediate reward, which is calculated based on the distance d^{ij} between i-th and j-th neighboring agents, assists the learning process of the DQN agent. Therefore, this immediate reward compensates for the negative effect of time-delayed data transfer at time step t and causes to take more appropriate action at the next time step $t+1$.

Since the agents are static and the distance d^{ij} between them is constant, the distance-based immediate reward (in combination with the package length and packet header size) helps the DQN agent to adjust the learning process over time in terms of data transmission speed. Hence, the Q-value is improved at each time step by speeding up the data transmission. This trend will continue in which, at higher time steps, the approximated

cumulative team discounted reward of time-delayed data transmission increases more than the on-time data transmission conditions. □

3.6. DQN Loss

The output layer loss function of the DQN algorithm's NN is the mean squared error (MSE) loss function. By decreasing the DQN loss, the DQN reward increases. Therefore, by observing the DQN loss behavior, the DQN reward performance is predicted. In [43], for uniform action selection, the DQN loss function is given by

$$\ell(\theta, s, a) = \frac{1}{B_e \cdot N} \sum_{i=1}^{N} \sum_{e_t} \left(r_{t+1}^{i} + \gamma \max_{a_{t+1}} Q^{i}(s_{t+1}, a_{t+1}; \theta^{-}) - Q^{i}(s_t, a_t; \theta) \right)^2, \tag{28}$$

where B_e is the experience replay mini-batch size. Using the non-uniform action (1), containing the set of $R_c \in \{0, 1, 2, \ldots, R_{c\max}\}$ as possible actions, and applying target Q-value (7) as well as predicted Q-value, the DQN loss function for non-uniform action selection with action time duration $T_d \in \{1, 2, \ldots, R_{c\max}\}$ is defined as

$$\ell(\theta, s, a) = \frac{1}{B_e \cdot N} \sum_{\substack{i=1 \\ T_d \in \{1,2,\ldots,R_{c\max}\}}}^{N} \sum_{e_t} \left(\frac{r_{t+1}^{i}}{T_d} \cdot \frac{1-\gamma^{T_d}}{1-\gamma} + \gamma^{T_d} \max_{a_{t+1}} Q^{i}(s_{t+1}, a_{t+1}; \theta^{-}) - Q^{i}(s_t, a_t; \theta) \right)^2, \tag{29}$$

where $e_t = (s_t, a_t, T_d, r_{t+1}, s_{t+1})$ is the experience at time t that is obtained from

$$e_t = (c_t^s, a_t, T_d, r_{t+1}, c_{t+1}^s).$$

Note that the Equation (28) is derived from Equation (29) if $T_d = 1$. Moreover, the time-delayed DQN loss function for action time duration $T_d' \in \{R_{c\max} + 1, R_{c\max} + 2, \ldots\}$ is given by

$$\ell(\theta, s, a) = \frac{1}{B_e \cdot N} \sum_{\substack{i=1 \\ T_d' \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}}}^{N} \sum_{e_t'} \left(\frac{r_{t+1}^{i}}{T_d'} \cdot \frac{1-\gamma^{T_d'}}{1-\gamma} + \gamma^{T_d'} \max_{a_{t+1}} Q^{i}(s_{t+1}, a_{t+1}; \theta^{-}) - Q^{i}(s_t, a_t; \theta) \right)^2, \tag{30}$$

for experience $e_t' = (s_t, a_t, T_d', r_{t+1}, s_{t+1})$ at time t that is achieved from $e_t' = (c_t^s, a_t, T_d', r_{t+1}, c_{t+1}^s)$. Note that average loss calculations based on experience e_t and experience e_t' are performed from $m = 1$ to B_e as the experience replay mini-batch size.

3.7. Adversarial Attacks

Three types of gradient-based adversarial attacks are considered to benchmark the data transmission robustness of the proposed leader-follower MAS by considering the new observation set and the proposed distance-based immediate reward (Figure 2). The changes made by this type of attack are very subtle, but they can also affect the system's performance. Before occurring an attack, the DQN algorithm aims to reduce the average training loss in a given time step and enhance the average reward.

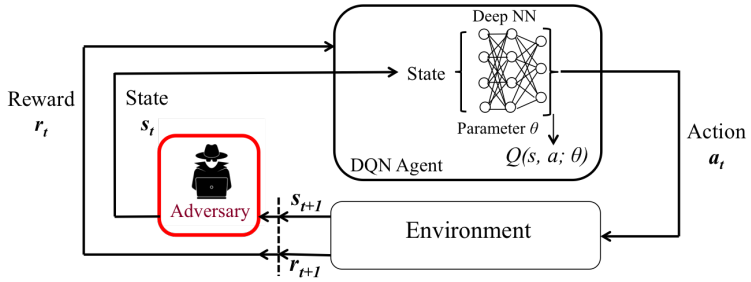


Figure 2. A DQN agent's structure affected by an adversarial attack.

3.7.1. FGSM Adversarial Attack

The FGSM is a type of attack proposed in [26]. Our methodology involves attacking the system's state by causing the FGSM adversary to make very few changes to the state over a period of time to increase the system's average training loss. Using the gradient of the loss function with respect to the state, FGSM maximizes the perturbation and minimizes the difference between the perturbed and original inputs [26,30,44]. In this regard, using (29) and (30), for on-time and time-delayed data transmission, respectively, the FGSM attack signal (perturbation) is obtained by

$$\eta = \epsilon \cdot \text{sign}(\nabla_s \ell(\theta, s, a)), \tag{31}$$

where ϵ is the attack magnitude to ensure the perturbations are small, and sign() is the sign function. Further, ∇_s is the gradient of the loss function related to model state s as well as correct action a, ℓ is the loss function of the DQN agent, and θ is the model parameters. After adding the attack signal to the state s, the adversarial input s_{adv} is calculated as follows

$$\begin{aligned} s_{adv} &= s + \eta \\ &= s + \epsilon \cdot \text{sign}(\nabla_s \ell(\theta, s, a)), \end{aligned} \tag{32}$$

where L_∞-norm bound $\|\eta\|_\infty \leq \epsilon$ for optimal perturbation η. Using (30) and (32), the adversarial input s_{adv} for time-delayed data transmission is given by

$$s_{adv} = s + \epsilon \cdot \text{sign}\left(\nabla_s \frac{1}{B_e \cdot N} \sum_{i=1}^{N} \sum_{e_t^i} \left(\frac{r_{t+1}^i}{T_d'} \cdot \frac{1-\gamma^{T_d'}}{1-\gamma} + \gamma^{T_d'} \max_{a_{t+1}} Q^i(s_{t+1}, a_{t+1}; \theta^-) - Q^i(s_t, a_t; \theta) \right)^2 \right). \tag{33}$$
$$T_d' \in \{R_{c\max}+1, R_{c\max}+2, \ldots\}$$

Once s_{adv} is calculated, it is fed to the NN and replaces the primary input s of the NN. The NN is fooled and trained based on the adversarial input s_{adv}.

3.7.2. FGM Adversarial Attack

The FGM attack signal is a generalization of the FGSM attack signal and is calculated as:

$$\eta = \epsilon \cdot \frac{\nabla_s \ell(\theta, s, a)}{\|\nabla_s \ell(\theta, s, a)\|_2}. \tag{34}$$

Using (34), the adversarial input s_{adv} is calculated by

$$\begin{aligned} s_{adv} &= s + \eta \\ &= s + \epsilon \cdot \frac{\nabla_s \ell(\theta, s, a)}{\|\nabla_s \ell(\theta, s, a)\|_2}, \end{aligned} \tag{35}$$

where L_2-norm bound $\|\eta\|_2 \leq \epsilon$ for optimal perturbation η. By substituting (30) in (35), the adversarial input s_{adv} for time-delayed data transmission is given. The training procedure is performed similarly to the FGSM adversarial attack.

3.7.3. BIM Adversarial Attack

The BIM attack is a simple and straight extension of the FGSM attack proposed by [29]. This method uses a fast gradient multiple times by considering small step size instead of applying the perturbation in a single step. The BIM attack signal and the adversarial input s_{adv} are given by

$$\eta = \beta \cdot \text{sign}\left(\nabla_{s_t^{adv}} \ell(\theta, s_t^{adv}, a)\right), \tag{36}$$

$$\begin{aligned} s_{t+1}^{adv} &= s_t^{adv} + \eta \\ &= s_t^{adv} + \beta \cdot \text{sign}\left(\nabla_{s_t^{adv}} \ell(\theta, s_t^{adv}, a)\right), \end{aligned} \tag{37}$$

where $\beta = \frac{\epsilon}{T}$ is a small step size and T is the number of iterations. Using (30) in (37), the adversarial input s_{t+1}^{adv} for time-delayed data transmission is provided.

3.8. First Adversarial Attack Defense

We provide a simple but effective approach to defend against adversarial attacks and mitigate their destructive effects on the MADRL system performance (Figure 3).

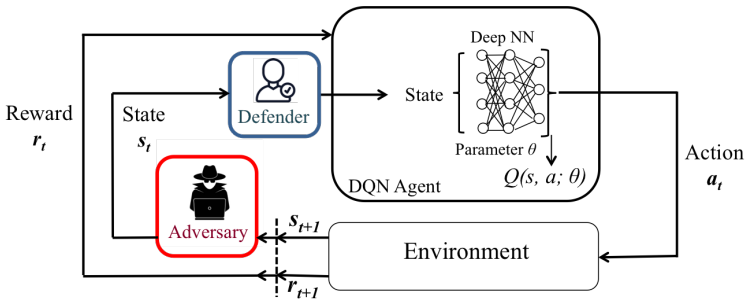


Figure 3. A DQN agent's structure affected by an adversarial attack and defense algorithm.

In the proposed Algorithm 1 that is based on NN behavior, we consider the `argmax` operation on perturbation vector η to find the argument that gives the maximum value from η. In other words, we desire to find a state that provides the maximum perturbation value. We assume that the FGSM, FGM, or BIM adversarial attacks are detectable. Once one of the FGSM, FGM, or BIM adversarial attacks is detected, the state vector s^* is calculated based on the set of states (inputs) and substituted with perturbation vector η as follows

$$\underset{n\times 1}{\vec{s^*}} = \arg\max_{s}(\underset{n\times 1}{\vec{\eta}}), \tag{38}$$

$$\underset{n\times 1}{\vec{\eta}} \longleftarrow \underset{n\times 1}{\vec{s^*}}, \tag{39}$$

where η and s^* are $n \times 1$ vectors. The state vector s^*, which determines the maximum perturbation value, has the worst effect on the MAS performance; however, the deep NN learns from the state vector s^* and uses its negative feedback to improve the system performance during an adversarial attack. For BIM adversarial attack the Equation (38) is presented as $\underset{n\times 1}{\vec{s^*}} = \arg\max_{s_t^{adv}}(\underset{n\times 1}{\vec{\eta}})$.

3.8.1. FGSM Adversarial Attack Defense

Using (31) and (32), the state vector s^* and the adversarial input s_{adv} are calculated to defend against the FGSM adversarial attack as follows

$$s^* = \arg\max_{s}(\epsilon \cdot \text{sign}(\nabla_s \ell(\theta, s, a))), \tag{40}$$

$$\epsilon \cdot \text{sign}(\nabla_s \ell(\theta, s, a)) \longleftarrow s^*, \tag{41}$$

$$\begin{aligned} s_{adv} &= s + \eta \\ &= s + \epsilon \cdot \text{sign}(\nabla_s \ell(\theta, s, a)) \\ &= s + s^*. \end{aligned} \tag{42}$$

Algorithm 1 First Adversarial Attack Defense

Input: s, s_t^{adv}, θ, T, T_{max}, ϵ, β
Output: s_{adv}, s_{t+1}^{adv}
for $T = 0, 1, 2, \ldots, T_{max}$ **do**
  if $attack = FGSM$ **then**
    $\eta = \epsilon \cdot \text{sign}(\nabla_s \ell(\theta, s, a))$;
  else if $attack = FGM$ **then**
    $\eta = \epsilon \cdot \frac{\nabla_s \ell(\theta,s,a)}{\|\nabla_s \ell(\theta,s,a)\|_2}$;
  else if $attack = BIM$ **then**
    $\beta = \frac{\epsilon}{T}$,
    $\eta = \beta \cdot \text{sign}\left(\nabla_{s_t^{adv}} \ell(\theta, s_t^{adv}, a)\right)$;
  end if
  while $attack = True$ **do**
    if $attack = FGSM$ OR $attack = FGM$ **then**
      $s^* = \arg\max_s(\eta)$,
      $\eta \longleftarrow s^*$,
      $s_{adv} = s + \eta$;
    else if $attack = BIM$ **then**
      $s^* = \arg\max_{s_t^{adv}}(\eta)$,
      $\eta \longleftarrow s^*$,
      $s_{t+1}^{adv} = s_t^{adv} + \eta$.
    end if
  end while
end for

3.8.2. FGM Adversarial Attack Defense

Utilizing (34) and (35), the state vector s^* and the adversarial input s_{adv} are computed to defend against the FGM adversarial attack as below

$$s^* = \arg\max_s \left(\epsilon \cdot \frac{\nabla_s \ell(\theta, s, a)}{\|\nabla_s \ell(\theta, s, a)\|_2} \right), \tag{43}$$

$$\epsilon \cdot \frac{\nabla_s \ell(\theta, s, a)}{\|\nabla_s \ell(\theta, s, a)\|_2} \longleftarrow s^*, \tag{44}$$

$$\begin{aligned} s_{adv} &= s + \eta \\ &= s + \epsilon \cdot \frac{\nabla_s \ell(\theta, s, a)}{\|\nabla_s \ell(\theta, s, a)\|_2} \\ &= s + s^*. \end{aligned} \tag{45}$$

3.8.3. BIM Adversarial Attack Defense

Using (36) and (37), the state vector s^* and the adversarial input s_{t+1}^{adv} are calculated to defend against the BIM adversarial attack as follows

$$s^* = \arg\max_{s_t^{adv}} \left(\beta \cdot \text{sign}\left(\nabla_{s_t^{adv}} \ell(\theta, s_t^{adv}, a) \right) \right), \tag{46}$$

$$\beta \cdot \text{sign}\left(\nabla_{s_t^{adv}} \ell(\theta, s_t^{adv}, a) \right) \longleftarrow s^*, \tag{47}$$

$$\begin{aligned} s_{t+1}^{adv} &= s_t^{adv} + \eta \\ &= s_t^{adv} + \beta \cdot \text{sign}\left(\nabla_{s_t^{adv}} \ell(\theta, s_t^{adv}, a) \right) \\ &= s_t^{adv} + s^*. \end{aligned} \tag{48}$$

3.9. Second Adversarial Attack Defense

We provide another effective method to mitigate the gradient-based attacks' destructive effects on the MADRL system performance and defend against the discussed adversarial attacks. We assume that the FGSM, FGM, or BIM adversarial attacks are detectable. In Algorithm 2, that is an extension of Algorithm 1, once one of the FGSM, FGM, or BIM adversarial attacks is detected, a `convert` function changes the sign of the state. Hence, before the attacker can confuse the NN, the state is modified and replaced with the correct state that was fed to the NN. This is done to mislead the attacker so that the attacker generates the attack signal η based on the converted state. Changing the state sign not only fools the attacker and reduces its destructive effects but also causes the generated attack signal by the attacker to be used for appropriate NN training. The remainder of the Algorithm 2 performs similarly to the Algorithm 1.

Algorithm 2 Second Adversarial Attack Defense

Input: s, s_t^{adv}, θ, T, T_{max}, ϵ, β
Output: s_{adv}, s_{t+1}^{adv}
for $T = 0, 1, 2, \ldots, T_{max}$ **do**
  if $attack = FGSM$ **then**
    $\eta = \epsilon \cdot \text{sign}(\nabla_s \ell(\theta, s, a))$;
  else if $attack = FGM$ **then**
    $\eta = \epsilon \cdot \frac{\nabla_s \ell(\theta, s, a)}{\|\nabla_s \ell(\theta, s, a)\|_2}$;
  else if $attack = BIM$ **then**
    $\beta = \frac{\epsilon}{T}$,
    $\eta = \beta \cdot \text{sign}\Big(\nabla_{s_t^{adv}} \ell(\theta, s_t^{adv}, a)\Big)$;
  end if
  while $attack = True$ **do**
    if $attack = FGSM$ OR $attack = FGM$ **then**
      function convert(s)
        return $-s$
      end function
      $s \longleftarrow -s$,
      attacker generates η,
      $s^* = \arg\max_s(\eta)$,
      $\eta \longleftarrow s^*$,
      $s_{adv} = s + \eta$;
    else if $attack = BIM$ **then**
      function convert(s_t^{adv})
        return $-s_t^{adv}$
      end function
      $s_t^{adv} \longleftarrow -s_t^{adv}$,
      attacker generates η,
      $s^* = \arg\max_{s_t^{adv}}(\eta)$,
      $\eta \longleftarrow s^*$,
      $s_{t+1}^{adv} = s_t^{adv} + \eta$.
    end if
  end while
end for

4. Results and Discussion

We illustrate results for on-time and time-delayed data transmission between agents of heterogeneous MAS with and without a leader, using the DQN algorithm. Additionally, the impacts of FGSM, FGM, and BIM attacks, as well as the consequences of the defense algorithms on the proposed leader-follower system, are illustrated and shown numerically.

Two types of graphs $\mathcal{G}$ are considered: complete (leaderless) and connected (leader-follower) graphs. The leaderless and leader-follower scenarios, including $N = 5$ static, heterogeneous agents, and $P = 3$ clusters, are illustrated in Figure 4.

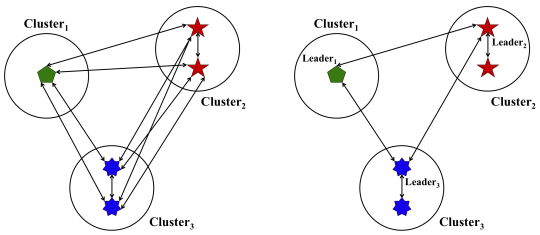


Figure 4. Two heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters. (**a**) Communication topology of a complete graph $\mathcal{G}$ without any leader. (**b**) Communication topology of a connected graph $\mathcal{G}$ with a leader at each cluster.

The adjacency matrices of the leaderless (left) and leader-follower (right) MAS are given by

$$\underset{5\times5}{A_{\mathcal{G}}} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}, \quad \underset{5\times5}{A_{\mathcal{G}}} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

The degree matrices of the leaderless (left) and leader-follower (right) MAS are

$$\underset{5\times5}{D_{\mathcal{G}}} = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix}, \quad \underset{5\times5}{D_{\mathcal{G}}} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

In both types of graphs, the DQN agent's internal structure consists of feed-forward NN architecture for training, including 36 layers with Adam optimizer and MSE loss function. Note that we have used the trial and error method to choose the number of NN hidden layers. To select any number of layers, we performed the learning process five times to reach a definite result about the number of layers. The activation function of all 36 layers is rectified linear unit (ReLU) function. The DQN agent learning rate is $\alpha = 0.01$, the discount factor is $\gamma = 0.999$, the experience replay mini-batch size is $B_e = 64$, and the constant positive real number to calculate immediate reward r_{t+1}^i is $\kappa = 4$ if the data packet is transferred in the network with time-delay. The packet's header for all agents is considered as $H_p = 0.5$. The threshold to determine the on-time or time-delayed data transmission is 11 mini-slots. To compute the attack signal, the attack magnitude is $\epsilon = 0.6$, and the number of iterations is $T_{max} = 30,000$. The five agents' two-dimensional positions are

$$(x^i, y^i) = \{(0.1, 0.22), (0.3, 0.27), (0.21, 0.9), (0.3, 0.23), (0.2, 0.4)\},$$

where $i \in \{1, 2, ..., 5\}$. The positions, which are used to obtain the distance-based immediate rewards of (4) and (5), remain constant during the total time steps due to the static agents. As opposed to this, when agents are mobile, their positions should be updated and added to the list of former positions at any time step, as we will investigate in the subsequent research. Moreover, the returned values of the novel observation set are

$$\begin{cases} [0,0,0,0,1] = busy, \\ [0,0,0,1,0] = idle, \\ [0,0,1,0,0] = on\text{-}time, \\ [0,1,0,0,0] = time\text{-}delay, \\ [1,0,0,0,0] = collided. \end{cases} \tag{49}$$

All scenarios are carried out during the 20,000 timesteps for the data transmission part of the experiment. The experiments are performed during the 30,000 timesteps while investigating the data transmission robustness due to various adversarial attacks. The results are shown after five times training to ensure the reliability of the results.

We have used, modified, and extended a part of the code given in [45] as a part of our implementation. Furthermore, for algorithm's execution, a system with 3.60 GHz Intel Core i7-7700 processor, 16 GB installed RAM, 64-bit operating system, and x64-based processor is used.

4.1. Multi-Agent Performance Analysis

According to Table 1, without considering distance-based reward and time-delay, both leaderless and leader-follower MAS scenarios achieve the superior team reward compared to the case when the packets transfer in the network with time-delay. In this case and for leaderless MAS, by considering time-delay, the team reward has been reduced by -27.04%. In a similar situation and for leader-follower MAS, by considering time-delay, the team reward has been decreased by -9.91%. Figure 5 illustrates the reward convergence of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps for leaderless and leader-follower scenarios by considering on-time and time-delay observations. Based on the results in Table 1 and Figure 5, delay in sending data reduces team rewards for both leaderless and leader-follower scenarios.

Table 1. Comparison of each agent's average reward and DQN loss of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps without considering the distance-based reward.

Rewards and Loss / Various Graphs	Agent 1 Reward	Agent 2 Reward	Agent 3 Reward	Agent 4 Reward	Agent 5 Reward	Team Reward	DQN Loss
Leaderless MAS	0.0047	0.0332	0.0665	0.1330	0.0997	0.3372	3411.58
Leaderless MAS with time-delay	0.0227	0.0427	0.0427	0.0380	0.0997	0.2460	6924.71
Leader-follower MAS	0.0147	0.0475	0.0570	0.0760	0.0902	0.2855	7325.22
Leader-follower MAS with time-delay	0.0340	0.0475	0.0237	0.0617	0.0902	0.2572	8400.12

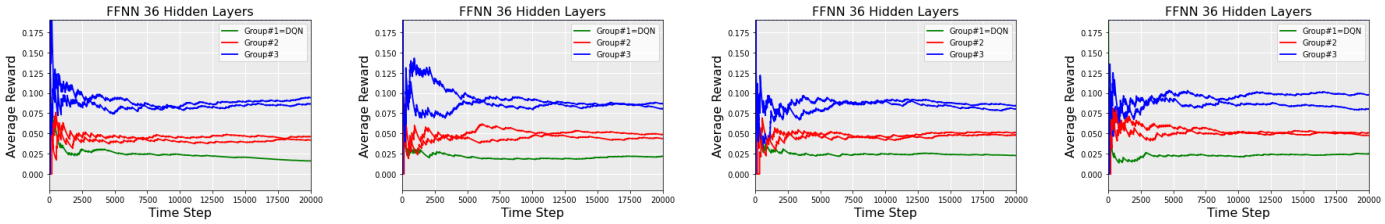


Figure 5. Reward convergence of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps. (**a**) Average reward of a leaderless MAS. (**b**) Average reward of a leaderless MAS by considering time-delay. (**c**) Average reward of a leader-follower MAS. (**d**) Average reward of a leader-follower MAS by considering time-delay.

As can be seen from Tables 1 and 2, the simulation confirms the claim of Theorem 2 in such a way that the distance-based immediate reward has improved the system performance despite the time-delayed data transmission (regardless of whether the system is leaderless or leader-follower). Moreover, according to Table 1, without considering distance-based reward and time delay, the DQN algorithm in both leaderless and leader-follower MAS scenarios achieves less average loss compared to the case when the packets transfer in the network with time-delay. For leaderless MAS, by considering time-delay, the average DQN loss has been increased by +102.97%. For leader-follower MAS, by considering time-delay, the average DQN loss has been enhanced by +14.67%. Figure 6 shows the DQN loss convergence of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 timesteps for leaderless and leader-follower scenarios by considering on-time and time-delay observations. The large fluctuations in the amount of loss after 10,000 timesteps in Figure 6b,d are due to delay in data transmission.

Table 2. Comparison of each agent's average reward and DQN loss of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps by considering the novel distance-based reward.

Rewards and Loss / Various Graphs	Agent 1 Reward	Agent 2 Reward	Agent 3 Reward	Agent 4 Reward	Agent 5 Reward	Team Reward	DQN Loss
Leaderless MAS	0.0202	0.0438	0.0485	0.0930	0.0735	0.2792	6037.81
Leaderless MAS with time-delay	0.0290	0.0438	0.0533	0.1028	0.1275	0.3566	6493.40
Leader-follower MAS	0.0095	0.0674	0.0288	0.0671	0.0768	0.2498	4116.27
Leader-follower MAS with time-delay	0.0322	0.0626	0.0336	0.0959	0.1008	0.3252	7983.38

Regarding Table 2, by considering distance-based reward and time-delay, both leaderless and leader-follower MAS scenarios achieve a higher team reward compared to the criteria when the packets transfer in the network on time. In this case and for leaderless MAS, by considering time-delay, the team reward has been increased by +27.72%. In a comparable status and for leader-follower MAS, by considering time-delay, the team reward has been enhanced by +30.18%. Figure 7 shows the reward convergence of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps for leaderless and leader-follower scenarios by considering on-time and time-delay observations as well as distance-based reward. Based on the results in Table 2 and Figure 7, the proposed distance-based immediate reward, in combination with the previous immediate reward, cover the negative effects of data transmission delays for both leaderless and leader-follower topologies.

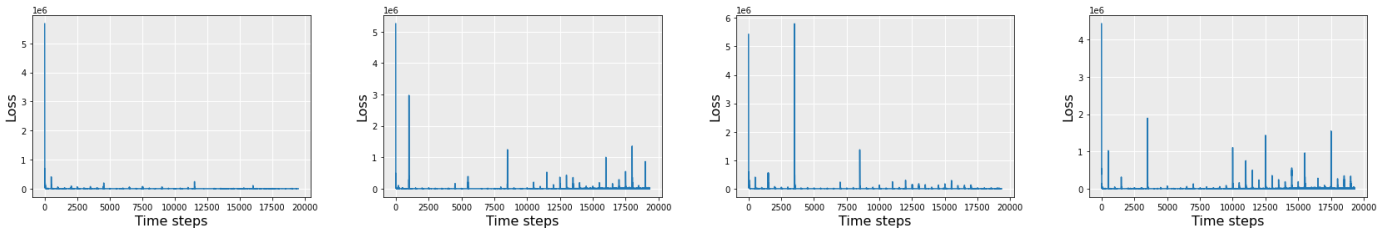


Figure 6. Loss convergence of the DQN algorithm in a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps. (**a**) DQN average loss in a leaderless MAS. (**b**) DQN average loss in a leaderless MAS by considering time-delay. (**c**) DQN average loss in a leader-follower MAS. (**d**) DQN average loss in a leader-follower MAS by considering time-delay.

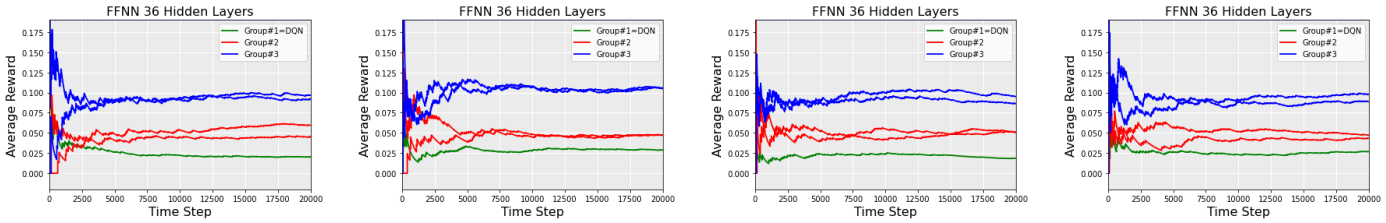


Figure 7. Reward convergence of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps by considering the novel distance-based reward. (**a**) Average reward of a leaderless MAS. (**b**) Average reward of a leaderless MAS by considering time-delay. (**c**) Average reward of a leader-follower MAS. (**d**) Average reward of a leader-follower MAS by considering time-delay.

Based on Table 2, by considering distance-based reward and time-delay, the DQN algorithm in leaderless and leader-follower MAS scenarios achieves a higher loss compared to the case when the packets transfer in the network on-time. For leaderless MAS, by considering time-delay, the average DQN loss has been increased by $+7.54\%$. In similar criteria and for leader-follower MAS, by considering time-delay, the average DQN loss has been enhanced by $+93.94\%$. Figure 8 demonstrates the DQN loss convergence of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps for leaderless and leader-follower scenarios by considering on-time and time-delay observations as well as distance-based reward. The time-delayed data transmission has caused large fluctuations in the amount of loss after 10,000 timesteps in Figure 8b,d. As can be seen from Tables 1 and 2, in scenarios that data is transmitted with time-delay, the average loss of DQN is increased compared to the cases where data is transferred on-time.

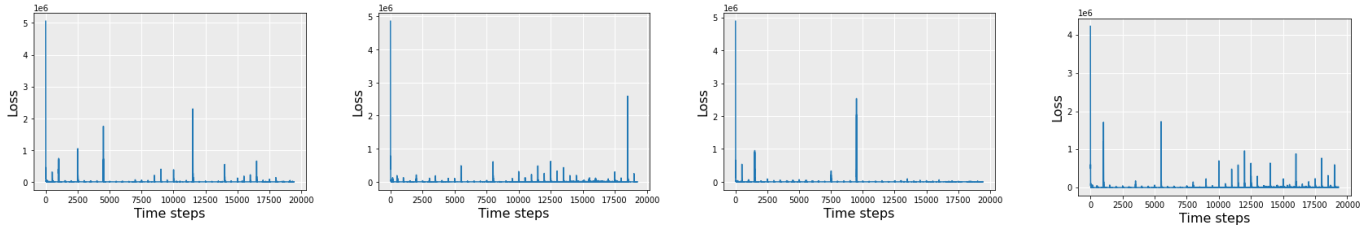


Figure 8. Loss convergence of the DQN algorithm in a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 20,000 time steps by considering the novel distance-based reward. (**a**) DQN average loss in a leaderless MAS. (**b**) DQN average loss in a leaderless MAS by considering time-delay. (**c**) DQN average loss in a leader-follower MAS. (**d**) DQN average loss in a leader-follower MAS by considering time-delay.

Note that the percentage increase of average loss and reward are calculated by

$$\%\ Inc = 100 \times \frac{(\text{With Time-Delay - Without Time-Delay})}{\mid \text{Without Time-Delay} \mid}. \tag{50}$$

Moreover, the percentage decrease of average loss and reward are computed by

$$\%\ Dec = 100 \times \frac{(\text{Without Time-Delay - With Time-Delay})}{\mid \text{Without Time-Delay} \mid}. \tag{51}$$

4.2. Performance Analysis of the Proposed MAS under Adversarial Attacks

According to Table 3, by considering time-delayed data transmission and distance-based reward, the team reward of the leader-follower MADRL system including $N = 5$

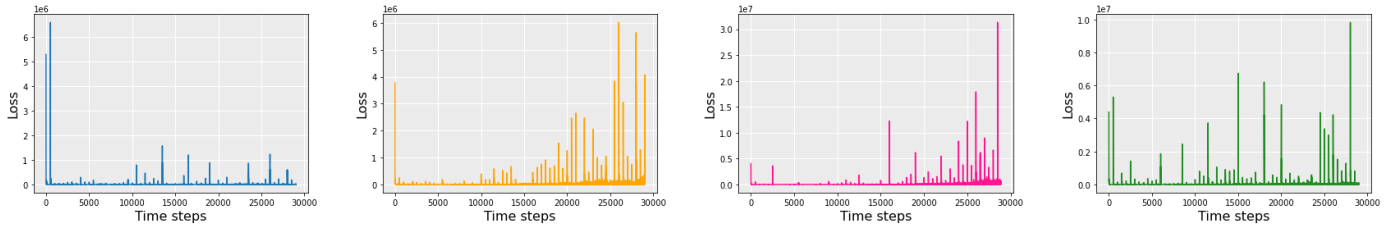
agents in $P = 3$ various clusters without adversarial attack equals to 0.3415 (Figure 9a). Moreover, in similar conditions, the DQN loss of the discussed MADRL system is 6996.28 (Figure 10a). Under FGSM adversarial attack, the team reward of the leader-follower MADRL system is decreased to 0.3236 by −5.24% (Figure 9b), and the DQN loss is increased to 20920.10 by +199.01% (Figure 10b). Furthermore, under FGM adversarial attack, the MADRL system team reward is reduced to 0.3054 by −10.57% (Figure 9c), and the DQN loss is enhanced to 60232.71 by +760.92% (Figure 10c). Under BIM adversarial attack, the team reward of the leader-follower MADRL system is declined to 0.2929 by −14.23% (Figure 9d), and the DQN loss is increased to 27949.57 by +299.49% (Figure 10d). Hence, it is evident that the time-delayed data transmission of the proposed leader-follower MADRL system is not robust under three types of adversarial attacks during 30,000 timesteps, meaning that its team reward is reduced after an attack, and the DQN loss is enhanced.

Table 3. Comparison of each agent's average reward and DQN loss of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 time steps by considering time-delay and distance-based reward under FGSM, FGM, and BIM adversarial attacks.

Rewards and Loss / Various Attacks	Agent 1 Reward	Agent 2 Reward	Agent 3 Reward	Agent 4 Reward	Agent 5 Reward	Team Reward	DQN Loss
Leader-follower MAS without attack	0.0340	0.0674	0.0432	0.0911	0.1056	0.3415	6996.28
Leader-follower MAS with FGSM attack	0.0353	0.0626	0.0432	0.0767	0.1056	0.3236	20,920.10
Leader-follower MAS with FGM attack	0.0315	0.0771	0.0480	0.0719	0.0768	0.3054	60,232.71
Leader-follower MAS with BIM attack	0.0335	0.0385	0.0480	0.0719	0.1008	0.2929	27,949.57

(**a**) (**b**) (**c**) (**d**)

Figure 9. Reward convergence of a heterogeneous leader-follower MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 time steps by considering time-delay and distance-based reward under FGSM, FGM, and BIM adversarial attacks. (**a**) Average reward of a leader-follower MAS without adversarial attack. (**b**) Average reward of a leader-follower MAS under FGSM adversarial attack. (**c**) Average reward of a leader-follower MAS under FGM adversarial attack. (**d**) Average reward of a leader-follower MAS under BIM adversarial attack.

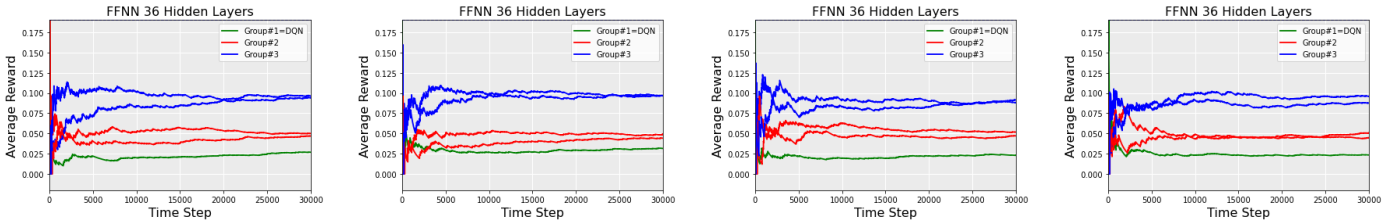


(**a**) (**b**) (**c**) (**d**)

Figure 10. Loss convergence of the DQN algorithm in a heterogeneous leader-follower MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 time steps by considering time-delay and distance-based reward under FGSM, FGM, and BIM adversarial attacks. (**a**) DQN average loss in a leader-follower MAS without adversarial attack. (**b**) DQN average loss in a leader-follower MAS under FGSM adversarial attack. (**c**) DQN average loss in a leader-follower MAS under FGM adversarial attack. (**d**) DQN average loss in a leader-follower MAS under BIM adversarial attack.

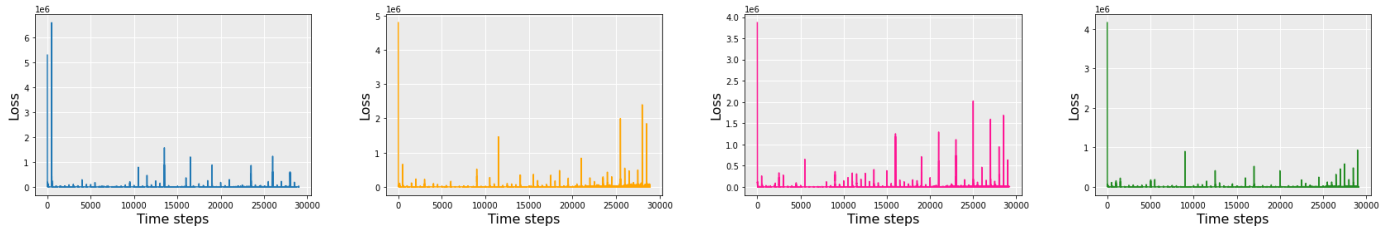
4.3. Performance Analysis of the Proposed MAS after Applying First Adversarial Attack Defense

Regarding Tables 3 and 4, after using the proposed adversarial attack defense Algorithm 1, the destructive effects of the FGSM, FGM, and BIM malicious attacks are mitigated during 30,000 time steps. In this regard, the team reward of the leader-follower MADRL system reached 0.3433 from 0.3236 by +6.08% after applying the adversarial attack defense method against the FGSM attack (Figure 11b). Moreover, the DQN loss is decreased from 20,920.10 to 8081.50 by −61.36% (Figure 12b). For FGM adversarial attack and after using the introduced defense procedure, the team reward of the MADRL system is enhanced from 0.3054 to 0.3342 by +9.43% (Figure 11c). The DQN loss is reduced from 60,232.71 to 6966.24 by −88.43% (Figure 12c). Furthermore, the team reward of the MADRL system under BIM attack is enhanced from 0.2929 to 0.3336 by +13.89%, and the DQN loss is decreased from 27,949.57 to 3705.51 by −86.74% after utilizing the suggested attack defense technique (Figures 11d and 12d).



(**a**) (**b**) (**c**) (**d**)

Figure 11. Reward convergence of a heterogeneous leader-follower MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 time steps by considering time-delay and distance-based reward after adversarial attack defense Algorithm 1. (**a**) Average reward of a leader-follower MAS without adversarial attack. (**b**) Average reward of a leader-follower MAS after defending against FGSM adversarial attack. (**c**) Average reward of a leader-follower MAS after defending against FGM adversarial attack. (**d**) Average reward of a leader-follower MAS after defending against BIM adversarial attack.



(**a**) (**b**) (**c**) (**d**)

Figure 12. Loss convergence of the DQN algorithm in a heterogeneous leader-follower MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 timesteps by considering time-delay and distance-based reward after adversarial attack defense Algorithm 1. (**a**) DQN average loss in a leader-follower MAS without adversarial attack. (**b**) DQN average loss in a leader-follower MAS after defending against FGSM adversarial attack. (**c**) DQN average loss in a leader-follower MAS after defending against FGM adversarial attack. (**d**) DQN average loss in a leader-follower MAS after defending against BIM adversarial attack.

Table 4. Comparison of each agent's average reward and DQN loss of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 time steps by considering time-delay and distance-based reward after adversarial attack defense Algorithm 1.

Various Attacks \ Rewards and Loss	Agent 1 Reward	Agent 2 Reward	Agent 3 Reward	Agent 4 Reward	Agent 5 Reward	Team Reward	DQN Loss
Leader-follower MAS without attack	0.0340	0.0674	0.0432	0.0911	0.1056	0.3415	6996.28
Leader-follower MAS with FGSM attack	0.0359	0.0626	0.0480	0.1150	0.0815	0.3433	8081.50
Leader-follower MAS with FGM attack	0.0172	0.0433	0.0720	0.1294	0.0720	0.3342	6966.24
Leader-follower MAS with BIM attack	0.0213	0.0771	0.0384	0.0959	0.1008	0.3336	3705.51

4.4. Performance Analysis of the Proposed MAS after Applying Second Adversarial Attack Defense

Regarding Tables 3 and 5, after using the proposed adversarial attack defense Algorithm 2, the destructive effects of the FGSM, FGM, and BIM malicious attacks are mitigated during 30,000 time steps. The team reward of the leader-follower MADRL system reached 0.3563 from 0.3236 by +10.10% after applying the adversarial attack defense method against the FGSM attack (Figure 13b). Moreover, the DQN loss is decreased from 20,920.10 to 2905.81 by −86.11% (Figure 14b). For FGM adversarial attack and after using the introduced defense procedure, the team reward of the MADRL system is enhanced from 0.3054 to 0.3187 by +4.35% (Figure 13c). The DQN loss is reduced from 60,232.71 to 4100.37 by −93.19% (Figure 14c). Furthermore, the team reward of the MADRL system under BIM attack is enhanced from 0.2929 to 0.3292 by +12.39%, and the DQN loss is decreased from 27,949.57 to 4526.58 by −83.80% after utilizing the suggested attack defense technique (Figures 13d and 14d).

Figure 15a,b shows the team reward and DQN loss before and after defense Algorithm 1 against various adversarial attacks, respectively. Figure 16a,b shows the team reward and DQN loss before and after defense Algorithm 2 against various adversarial attacks, respectively.

Table 5. Comparison of each agent's average reward and DQN loss of a heterogeneous MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 time steps by considering time-delay and distance-based reward after adversarial attack defense Algorithm 2.

Various Attacks \ Rewards and Loss	Agent 1 Reward	Agent 2 Reward	Agent 3 Reward	Agent 4 Reward	Agent 5 Reward	Team Reward	DQN Loss
Leader-follower MAS without attack	0.0340	0.0674	0.0432	0.0911	0.1056	0.3415	6996.28
Leader-follower MAS with FGSM attack	0.0105	0.0626	0.0480	0.1246	0.1103	0.3563	2905.81
Leader-follower MAS with FGM attack	0.0306	0.0578	0.0288	0.1006	0.1007	0.3187	4100.37
Leader-follower MAS with BIM attack	0.0314	0.0530	0.0528	0.1006	0.0911	0.3292	4526.58

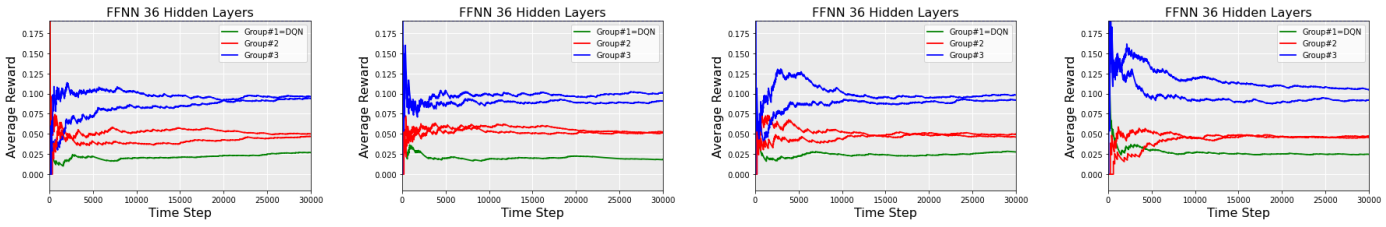


Figure 13. Reward convergence of a heterogeneous leader-follower MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 time steps by considering time-delay and distance-based reward after adversarial attack defense Algorithm 2. (**a**) Average reward of a leader-follower MAS without adversarial attack. (**b**) Average reward of a leader-follower MAS after defending against FGSM adversarial attack. (**c**) Average reward of a leader-follower MAS after defending against FGM adversarial attack. (**d**) Average reward of a leader-follower MAS after defending against BIM adversarial attack.

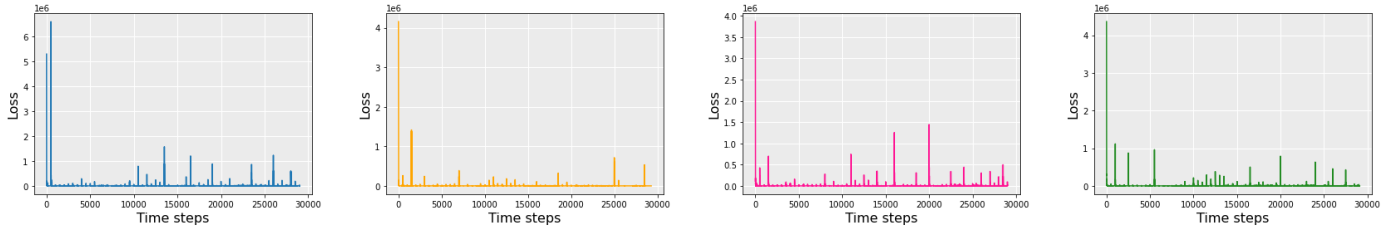


Figure 14. Loss convergence of the DQN algorithm in a heterogeneous leader-follower MAS, including $N = 5$ agents in $P = 3$ different clusters, during 30,000 time steps by considering time-delay and distance-based reward after adversarial attack defense Algorithm 2. (**a**) DQN average loss in a leader-follower MAS without adversarial attack. (**b**) DQN average loss in a leader-follower MAS after defending against FGSM adversarial attack. (**c**) DQN average loss in a leader-follower MAS after defending against FGM adversarial attack. (**d**) DQN average loss in a leader-follower MAS after defending against BIM adversarial attack.

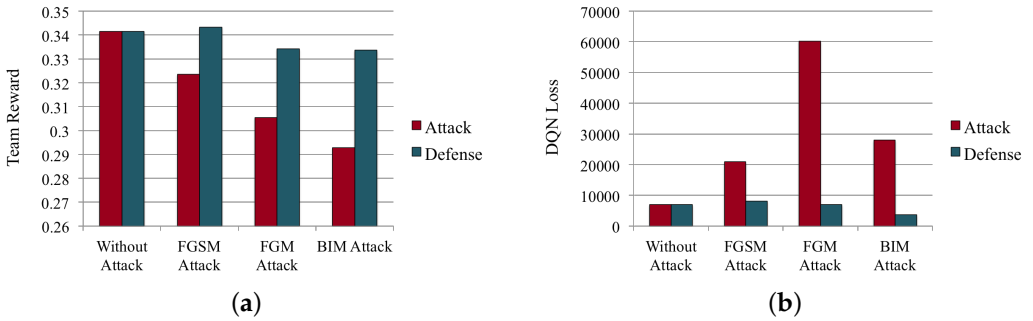


Figure 15. Team reward and DQN loss before and after using defense Algorithm 1 against various adversarial attacks, during 30,000 time steps. (**a**) Team reward before and after defense Algorithm 1. (**b**) DQN loss before and after defense Algorithm 1.

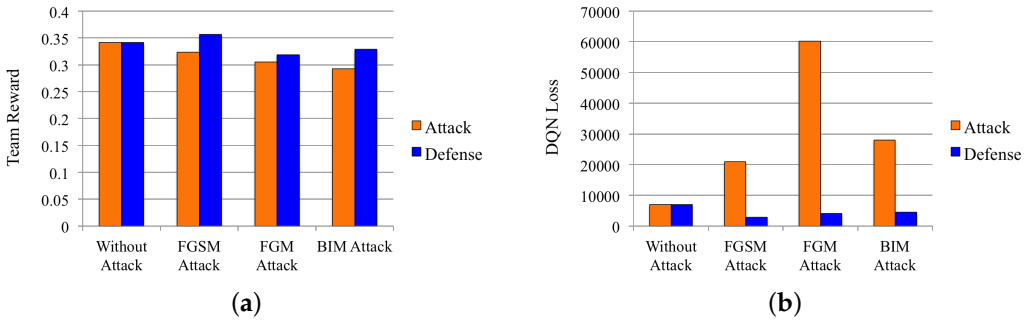


Figure 16. Team reward and DQN loss before and after using defense Algorithm 2 against various adversarial attacks, during 30,000 time steps. (**a**) Team reward before and after defense Algorithm 2. (**b**) DQN loss before and after defense Algorithm 2.

4.5. Variety of Agents

According to [20], in the proposed topology, there are three types of agents assigned to $P = 3$ different clusters. The first cluster's agents use a DQN architecture. The agents in the second cluster follow the ALOHA protocol [46–48]. Moreover, the time division multiple access (TDMA) protocol makes up the agents' architecture of the third cluster [49]. The concentration of this paper is on DQN agent behaviors (first cluster) and their effects on the other clusters of agents' performance of the MADRL system in different situations.

5. Conclusions

We studied the on-time and time-delayed data transmission of a leaderless (complete graph), heterogeneous, MADRL system using the DQN algorithm. Moreover, we investigated the on-time and time-delayed data transmission of a leader-follower (connected graph), heterogeneous, MADRL system using the DQN algorithm as well. We studied the MADRL system's performance under various conditions. We did the data transmission investigation on a cluster-based MAS. We proposed a novel immediate reward, including a new version of distance-based reward. We used three types of adversarial attacks to check the data transmission robustness of the MADRL system. We introduced two approaches to defend against malicious attacks and mitigate the destructive effects of adversarial attacks. The results of various scenarios were demonstrated and compared with each other numerically.

Future work will contain agents including different NN architectures in the MADRL system to reach the position consensus. Further, adversarial attack detection will be considered. Moreover, another adversarial attack defense approach will be introduced. Furthermore, the proposed model will examine obstacle and collision avoidance.

Author Contributions: Conceptualization, N.E.F.; methodology, N.E.F.; software, N.E.F.; formal analysis, N.E.F.; investigation, N.E.F.; writing—original draft preparation, N.E.F.; review and editing, R.R.S.; supervision, R.R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BIM	Basic Iterative Method
CS-DLMA	Carrier-sense Deep Reinforcement Learning Multiple Access
CSMA	Carrier-sense Multiple Access
DQN	Deep Q-network
DL	Deep Learning
DRL	Deep Reinforcement Learning
FGM	Fast Gradient Method
FGSM	Fast Gradient Sign Method
ML	Machine Learning
MSE	Mean Squared Error
MADRL	Multi-agent Deep Reinforcement Learning
MAS	Multi-agent System
MTDDQN	Multi-source Transfer Double DQN
NN	Neural Network
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
TDMA	Time Division Multiple Access
TL	Transfer Learning

References

1. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA , 2018.
2. Canese, L.; Cardarilli, G.C.; Di Nunzio, L.; Fazzolari, R.; Giardino, D.; Re, M.; Spanò, S. Multi-agent reinforcement learning: A review of challenges and applications. *Appl. Sci.* **2021**, *11*, 4948. [CrossRef]

3. Mousavi, S.S.; Schukat, M.; Howley, E. Deep reinforcement learning: An overview. In Proceedings of the SAI Intelligent Systems Conference, London, UK, 21–22 September 2016 ; Springer: Berlin/Heidelberg, Germany, 2016; pp. 426–440.
4. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
5. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An introduction to deep reinforcement learning. *arXiv* **2018**, arXiv:1811.12560.
6. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
7. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
8. Chen, Z.; Zhang, S.; Doan, T.T.; Maguluri, S.T.; Clarke, J.P. Performance of Q-learning with linear function approximation: Stability and finite-time analysis. *arXiv* **2019**, arXiv:1905.11425.
9. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
10. Gao, Z.; Gao, Y.; Hu, Y.; Jiang, Z.; Su, J. Application of deep q-network in portfolio management. In Proceedings of the Proceedings of the IEEE International Conference on Big Data Analytics (ICBDA), Xiamen, China, 8–11 May 2020; pp. 268–275.
11. Carta, S.; Ferreira, A.; Podda, A.S.; Recupero, D.R.; Sanna, A. Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting. *Expert Syst. Appl.* **2021**, *164*, 113820. [CrossRef]
12. Jang, B.; Kim, M.; Harerimana, G.; Kim, J.W. Q-learning algorithms: A comprehensive classification and applications. *IEEE Access* **2019**, *7*, 133653–133667. [CrossRef]
13. Pan, J.; Wang, X.; Cheng, Y.; Yu, Q. Multisource transfer double DQN based on actor learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2227–2238. [CrossRef]
14. Zhao, C.; Liu, X.; Zhong, S.; Shi, K.; Liao, D.; Zhong, Q. Secure consensus of multi-agent systems with redundant signal and communication interference via distributed dynamic event-triggered control. *ISA Trans.* **2021**, *112*, 89–98. [CrossRef]
15. Zheng, S.; Shi, P.; Agarwal, R.K.; Lim, C.P. Periodic event-triggered output regulation for linear multi-agent systems. *Automatica* **2020**, *122*, 109223. [CrossRef]
16. Yuan, S.; Yu, C.; Sun, J. Adaptive event-triggered consensus control of linear multi-agent systems with cyber attacks. *Neurocomputing* **2021**, *442*, 1–9. [CrossRef]
17. Chen, M.; Yan, H.; Zhang, H.; Chi, M.; Li, Z. Dynamic event-triggered asynchronous control for nonlinear multi-agent systems based on TS fuzzy models. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 2580–2592 . [CrossRef]
18. Rehan, M.; Tufail, M.; Ahmed, S. Leaderless Consensus Control of Nonlinear Multi-agent Systems under Directed Topologies subject to Input Saturation using Adaptive Event-Triggered Mechanism. *J. Frankl. Inst.* **2021**, *358*, 6217–6239 . [CrossRef]
19. Zhou, B.; Yang, Y.; Li, L.; Hao, R. Leaderless and leader-following consensus of heterogeneous second-order multi-agent systems on time scales: An asynchronous impulsive approach. *Int. J. Control* **2021**, 1–11 . [CrossRef]
20. Yu, Y.; Liew, S.C.; Wang, T. Non-Uniform Time-Step Deep Q-Network for Carrier-Sense Multiple Access in Heterogeneous Wireless Networks. *IEEE Trans. Mob. Comput.* **2021**, *20*, 2848–2861. [CrossRef]
21. Chakraborty, A.; Alam, M.; Dey, V.; Chattopadhyay, A.; Mukhopadhyay, D. Adversarial attacks and defences: A survey. *arXiv* **2018**, arXiv:1810.00069.
22. Akhtar, N.; Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **2018**, *6*, 14410–14430. [CrossRef]
23. Goodfellow, I.; McDaniel, P.; Papernot, N. Making machine learning robust against adversarial inputs. *Commun. ACM* **2018**, *61*, 56–66. [CrossRef]
24. Wang, X.; Li, J.; Kuang, X.; Tan, Y.a.; Li, J. The security of machine learning in an adversarial setting: A survey. *J. Parallel Distrib. Comput.* **2019**, *130*, 12–23. [CrossRef]
25. Pitropakis, N.; Panaousis, E.; Giannetsos, T.; Anastasiadis, E.; Loukas, G. A taxonomy and survey of attacks against machine learning. *Comput. Sci. Rev.* **2019**, *34*, 100199. [CrossRef]
26. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
27. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial machine learning at scale. *arXiv* **2016**, arXiv:1611.01236.
28. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 9185–9193.
29. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2016**, arXiv:1607.02533.
30. Haydari, A.; Zhang, M.; Chuah, C.N. Adversarial Attacks and Defense in Deep Reinforcement Learning (DRL)-Based Traffic Signal Controllers. *IEEE Open J. Intell. Transp. Syst.* **2021**, *2*, 402–416 . [CrossRef]
31. Hussenot, L.; Geist, M.; Pietquin, O. Manipulating Neural Policies with Adversarial Observations. In Proceedings of the Real-World Sequential Decision Making Workshop, ICML, Long Beach, CA, USA, 14 June 2019.
32. Yuan, Z.; Zhang, J.; Jia, Y.; Tan, C.; Xue, T.; Shan, S. Meta gradient adversarial attack. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 7748–7757.
33. Metzen, J.H.; Genewein, T.; Fischer, V.; Bischoff, B. On detecting adversarial perturbations. *arXiv* **2017**, arXiv:1702.04267.

34. Dong, Y.; Su, H.; Zhu, J.; Bao, F. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv* **2017**, arXiv:1708.05493.
35. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv* **2017**, arXiv:1705.07204.
36. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 582–597.
37. Elhami Fard, N.; Selmic, R.R. Time-delayed Data Transmission in Heterogeneous Multi-agent Deep Reinforcement Learning System. In Proceedings of the 2022 30th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 28 June–1 July 2022; pp. 636–642.
38. Mesbahi, M.; Egerstedt, M. *Graph Theoretic Methods in Multiagent Networks*; Princeton University Press: Princeton, NJ, USA , 2010; Volume 33.
39. Wang, S.; Diao, R.; Lan, T.; Wang, Z.; Shi, D.; Li, H.; Lu, X. A DRL-aided multi-layer stability model calibration platform considering multiple events. In Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM), Virtual, 3–6 August 2020; pp. 1–5.
40. Liu, Y.; Cao, B.; Li, H. Improving ant colony optimization algorithm with epsilon greedy and Levy flight. *Complex Intell. Syst.* **2021**, *7*, 1711–1722. [CrossRef]
41. Elhami Fard, N.; Selmic, R.R. Consensus of Multi-agent Reinforcement Learning Systems: The Effect of Immediate Rewards. *J. Robot. Control (JRC)* **2022**, *3*, 115–127. [CrossRef]
42. Ohnishi, S.; Uchibe, E.; Yamaguchi, Y.; Nakanishi, K.; Yasui, Y.; Ishii, S. Constrained deep q-learning gradually approaching ordinary q-learning. *Front. Neurorobot.* **2019**, *13*, 103. [CrossRef] [PubMed]
43. Yu, Y.; Wang, T.; Liew, S.C. Deep-reinforcement learning multiple access for heterogeneous wireless networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1277–1290. [CrossRef]
44. Ammouri, K. Deep Reinforcement Learning for Temperature Control in Buildings and Adversarial Attacks. 2021. Available online: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1590898&dswid=-7284 (accessed on 6 September 2021).
45. Yu, Y. CS-DLMA. 2019. Available online: https://github.com/YidingYu/CS-DLMA (accessed on 26 August 2019).
46. Abramson, N. The ALOHA system: Another alternative for computer communications. In Proceedings of the Fall Joint Computer Conference, Houston, TX, USA, 17–19 November 1970; pp. 281–285.
47. Kuo, F.F. The ALOHA system. *ACM SIGCOMM Comput. Commun. Rev.* **1995**, *25*, 41–44. [CrossRef]
48. Kuo, F.F. *Computer Networks–The ALOHA System*; Technical report; Hawaii University at Manoa Honolulu Department of Electrical Engineering: Honolulu, HI, USA , 1981.
49. Jung, P. Time Division Multiple Access (TDMA). In *Wiley Encyclopedia of Telecommunications*; John Wiley & Sons: Hoboken, NJ, USA, 2003.