2022 IEEE International Conference on Assured Autonomy (ICAA) | 978-1-6654-8539-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICAA52185.2022.00014

Resilient Multi-agent Reinforcement Learning Using Medoid and Soft-medoid Based Aggregation

Chandreyee Bhowmick*, Mudassir Shabbir*, Waseem Abbas†, and Xenofon Koutsoukos*
*Institute for Software Integrated Systems, Vanderbilt University, Nashville, USA
Email: {chandreyee.bhowmick, mudassir.shabbir, xenofon.koutsoukos}@vanderbilt.edu
†Department of Systems Engineering, University of Texas at Dallas, Richardson, USA
Email: waseem.abbas@utdallas.edu

***Abstract*—A network of reinforcement learning (RL) agents that cooperate with each other by sharing information can improve learning performance of control and coordination tasks when compared to non-cooperative agents. However, networked Multi-agent Reinforcement Learning (MARL) is vulnerable to adversarial agents that can compromise some agents and send malicious information to the network. In this paper, we consider the problem of resilient MARL in the presence of adversarial agents that aim to compromise the learning algorithm. First, the paper presents an attack model which aims to degrade the performance of a target agent by modifying the parameters shared by an attacked agent. In order to improve the resilience, the paper presents aggregation methods using medoid and soft-medoid. Our analysis shows that the medoid-based MARL algorithms converge to an optimal solution given standard assumptions, and improve the overall learning performance and robustness. Simulation results show the effectiveness of the aggregation methods compared with average and median-based aggregation.**

***Index Terms*—Actor-critic algorithm, adversarial attacks, multi-agent reinforcement learning, neural networks, resilient aggregation.**

I. Introduction

With the recent advancements of deep learning, Reinforcement Learning (RL) has been applied to many real-world problems with excellent learning performance [1], [2]. At the same time, distributed learning has become a prominent research area in recent years, due to the astonishing progress in machine learning, data availability, and increased computational efficiency. Some of its applications are in cellular networks, smart homes, smart wearable [3], etc. Both these areas together have given rise to a very exciting research direction, namely Multi-agent Reinforcement Learning (MARL), and it has started receiving an increased amount of research attention from various communities [4]–[6]. In any distributed learning, each agent shares information with some of the other agents, referred to as its *neighbors*. Based on how the agents interact with the environments, MARL can be of two types. The first type, *MARL in shared MDP*, is where the agents operate in the same environment, and one agent's action influences others [7]. In contrast, we are interested in the *MARL in independent MDP* setup, where the agents operate in similar but independent environments, and one agent's actions do not influence any other agent [8]. The agents aggregate the *useful* information received by communicating with their neighbors. Due to security concerns, it is often not practical to share the observation data between agents [9]; instead, they cooperate by sharing the model parameters. This type of operation is applied in a cellular network, where the service providers attempt to learn the user behavior by interacting with different data, but the underlying models are similar.

The reliability of the system and the learning performance can be improved by using cooperation among the agents. The agents share information with each other and update the parameters accordingly, which causes an overall improved learning performance [8], [10], [11]. This is achieved either by using a centralized network, where all the agents send their data to the same server [12], [13], or in a decentralized way, where each agent makes its own decision while communicating with a subset of other agents [7], [10], [14]. Distributed MARL enjoys some distinct advantages in contrast to the centralized setup. First, parallel computation can be enabled, which significantly improves the utilization of computational resources. Second, it is easier to add new agents and remove malfunctioning ones from a decentralized network. In addition, a fully-decentralized distributed learning effectively addresses the scalability issue and single-point-of-failure problem.

Even though cooperation among agents helps improve learning performance in an ideal scenario, the overall performance of the network may deteriorate in practical cases when some of the agents are influenced by an external adversary [8]. MARL algorithms are subject to potential attacks from various adversarial entities that may target a fraction of agents in the network. When these attacked agents share the corrupted information with their neighbors, propagation of this harmful information may occur potentially deteriorating the performance of the entire network. This makes it highly critical for the healthy agents to have a mechanism to maintain their performance in the presence of a few attacked neighbors whose identity is unknown to them. The research efforts existing in literature show that the presence of adversarial attacks in practical applications of reinforcement learning algorithms may compromise its performance. For example, [15] shows that strategically designed attack can affect the performance of Atari games. RL applied to path-planning application can get affected due to cyber-attacks [16], [17]. Thus, the design of resilient algorithms has become a vital area of research in distributed learning. The main goal is to enable normal agents to mitigate the effect of corrupted parameters shared by adversarial agents. One of the most effective ways of achieving

978-1-6654-8539-5/22/$31.00 ©2022 IEEE
DOI 10.1109/ICAA52185.2022.00014

this goal is by using resilient aggregation, where the healthy agents update their parameters based on the local and the shared information [7], [18], while filtering out the effect of the adversarial information as much as possible.

Similar to the design of resilient learning methods, developing effective attack models is an active area of research. There are multiple ways to design an attack. Here we utilize communication between agents for this purpose. The adversary modifies the shared parameter of the attacked agent to minimize the reward of the targeted agent. Attack design by exploiting the inter-agent communication has been explored in federated learning [19]; however, to the best of our knowledge, it has not been applied in the context of MARL.

In this paper, we consider a few different learning tasks, which the agents learn using an actor-critic algorithm. Actor-critic learning methods, which are modifications of the standard policy gradient RL method, are effective for various control and learning tasks [7], [12], [20]. Actor-critic approach requires two networks - the critic estimates the state value function, whereas the actor network evaluates the new state of the system. Here, we present two novel targeted attack models - depending on which aggregation method is being used by the RL agents. It is assumed that the adversary has complete knowledge about the agents' algorithm, and the underlying graph of the communication between agents. The adversary uses this inter-agent communication to launch the attack, where the parameters shared by a particular agent (victim node) are modified to inject the attack to the system. The attack design involves solving an optimization problem, and it is shown to be effective through simulation results. Medoid, which is a generalization of median in a higher dimension, and soft-medoid, which is a softmax version of the medoid [21], are introduced as aggregation methods in this paper. The aggregation in MARL deals with higher dimension data, which limits the choice for the aggregation function, as many geometric functions do not translate well in higher dimensions. Both medoid and soft-medoid have the properties of being applicable in higher dimensions, and the aggregated data point always lies in the convex hull of the original data points. In this work, the soft-medoid aggregation achieves the highest collective reward among all aggregation.

The contributions of this paper are the following:

i) We design targeted attacks by solving an optimization problem that uses inter-agent communication. Two attack models are designed that are suitable for a different type of baseline aggregation method.
ii) We use medoid and soft-medoid as aggregation protocols to combine the shared parameters from the neighbors. These methods ensure better learning performance compared to non-cooperative scenario, and are also resilient against the proposed attacks. We analytically prove convergence of the learned parameters and the collective performance improvement in the absence of adversary.
iii) To measure the resilience of these aggregation methods against attacks, we perform the breakdown point analysis of these methods.
iv) Simulation results show the effectiveness of the aggregation methods in comparison with two baseline methods. The proposed attack models are also simulated, and the aggregation methods are tested against them.

II. Related Work

In this section, we discuss a few key works from the literature on relevant areas.

A. Multi-agent Reinforcement Learning

The problem of MARL in the shared MDP scenario has been thoroughly explored in the literature. Based on the assigned tasks and the environment, the agents can be cooperative or competitive [12]. They learn more sophisticated and complicated tasks through communication with each other. Some of these works [12], [22], [23] use the paradigm of centralized training with decentralized execution. There are works in the literature that propose novel methods and explore various application fields [24]–[28]. There are a few efforts that explore the MARL in an independent environment setting. A distributed Q-learning method was designed in [14]. Distributed policy evaluation method with linear function approximation was presented in [10]. Some other important works in this area include [29], [30].

B. Attack Design in MARL

There are various types of attack that can be applied to RL algorithms, depending on factors such as the time of the attack, knowledge available to the adversary, or the attacked signal [31], [32]. However, attack design for the MARL system needs more research attention. Among the few existing efforts, the work in [33] has designed an attack for cooperative MARL in an interactive environment, where an adversarial agent aims to maximize a malicious objective, and disregards other agents' objectives. The effort in [34] was to design an effective attack model for *centralized training distributed execution based MARL* that is capable of reducing the total team reward by attacking a single agent. Byzantine attack has been applied to MARL in both independent [8] and cooperative [18] settings.

C. Resilient Aggregation in Learning

In reinforcement learning, the performance enhancement has been achieved by average-based aggregation [10], or consensus-based method [7]. A coordinate-wise trimmed mean was used in [18] as a method of resilient aggregation. In federated learning, various methods of aggregation functions have been developed and explored, which have shown to be effective against Byzantine attacks, examples include coordinate-wise median [35], coordinate-wise trimmed mean [36], Krun and multi-Krum [37], Bulyan and multi-Bulyan [38].

III. System Model And Problem Statement

Markov decision Process (MDP) is predominantly used in describing the operation in RL [39]. An MDP can be characterized as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R \rangle$ [7], where $\mathcal{S}$ and $\mathcal{A}$ denote the finite state and action spaces, respectively. $P(s'|s,a) : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0,1]$ is the state transition probability from state s to another state s' determined by an action a, and

$R(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function defined as $R(s,a) = \mathbb{E}[r_{t+1}|s_t = s, a_t = a]$, with r_{t+1} being the immediate reward received at time t. An agent's action is defined by a function π. In the case of stochastic policy, it is the probability of choosing action a at state s, given as, $\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$. For deterministic policies, the mapping is defined as, $\pi : \mathcal{S} \to \mathcal{A}$.

The notations used in this paper are fairly standard. For a finite set A, the cardinality is denoted by $|A|$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$ and $(\cdot)^\top$ denotes matrix transposition.

In this work, we consider a network of N agents, represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V}$ represent the agents, and the set of edges $\mathcal{E}$ represents pairwise interactions between them. An edge $(l,k) \in \mathcal{E}$, where $l, k \in \mathcal{V}$, signifies that agents k and l exchange information with each other. The neighborhood of agent k is the set of agents that it interacts with including the agent k itself, and is denoted as $\mathcal{N}_k = \{l \in \mathcal{V} | (l,k) \in \mathcal{E}\} \cup \{k\}$. The cardinality of the set $\mathcal{N}_k$ is denoted as n_k. We consider a stationary graph, meaning that the set of neighbors for each agent remains the same throughout the operation.

The agents operate in independent but similar environments, which are modeled as independent MDPs given by $\mathcal{M}_k = \langle \mathcal{S}, \mathcal{A}, P^k, r^k \rangle$ for $k \in [N]$. This representation indicates that the state and action spaces are fixed for all the agents, but the transition probability and the reward functions may be different. As the agents operate in independent MDPs, their actions do not influence each other. The expected time-average return of policy π for agent k is defined as

$$J_k(\pi) = \lim_T \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}(r_{t+1}^k) = \sum_{s \in \mathcal{S}} d_\pi^k(s) \sum_{a \in \mathcal{A}} \pi(s,a) R^k(s,a)$$

where $d_\pi^k = \lim_t P^k(s_t = s|\pi)$ is the stationary distribution of the Markov chain under policy π for agent k. Further, we define the action-value associated with a state-action pair (s,a) under policy π for agent k at instant t as $Q_t^k(s,a) = \sum_t \mathbb{E}\left[r_{t+1}^k - J_k(\pi) | s_0 = s, a_0 = a, \pi\right]$. The objective of the networked agents is to cooperatively learn the optimal policy that maximizes the following optimization function:

$$\max_{\theta_1, \ldots, \theta_N} \left\{ \frac{1}{N} \sum_{k=1}^{N} J_k(\theta_k) \right\}. \tag{1}$$

In RL, the action-value function and the policy function can be parametrized, and the same concept is extended in MARL. We assume that the action-value function Q_t^k of agent k has the parameter w_t^k, and the policy function of the same agent, π_k has parameters θ_t^k.

In this work, we consider that each agent executes an independent actor-critic algorithm which includes a combination step to aggregate the parameters from the neighboring agents. The agents share the actor and critic parameters with their neighbors, and in the combination step that follows the update step, each agent updates the parameters once again using the shared parameters from the neighbors. In a standard actor-critic algorithm [39], the critic update step is given as

$$\begin{aligned} \mu_{t+1}^k &= (1 - \beta_{w,t}^k) \cdot \mu_t^k + \beta_{w,t}^k \cdot r_{t+1}^k, \\ \tilde{w}_t^k &= w_t^k + \beta_{w,t}^k \cdot \delta_t^k \cdot \nabla_w Q_t^k(w_t^k). \end{aligned} \tag{2}$$

where the action-value function is $Q_t^k(w) \triangleq Q^k(s_t^k, a_t^k; w)$, the action-value temporal difference (TD) error is given by $\delta_t^k \triangleq r_{t+1}^k - \mu_t^k + Q_t^k(w_t^k) - Q_t^k(w_t^k)$, and $\beta_{w,t}^k$ is the step-size. The actor-step is given as

$$\tilde{\theta}_t^k = \theta_t^k + \beta_{\theta,t}^k \cdot Q_t^k(w_t^k) \cdot \psi_t^k, \tag{3}$$

where $\psi_t^k \triangleq \nabla_\theta \log \pi_{\theta_t^k}(s_t^k, a_t^k)$ is the gradient of the log of the policy, and $\beta_{\theta_t}^k > 0$ is the step-size.

Here $\tilde{w}_t^k$ and $\tilde{\theta}_t^k$ are the intermediate values of the parameters following the update step. These parameter are then shared with the neighboring agents, and the agents use these information at the combination step, given by

$$w_{t+1}^k = \sum_{l \in \mathcal{N}_k} c_t(k,l) \cdot \tilde{w}_t^l, \tag{4}$$

$$\theta_{t+1}^k = \sum_{l \in \mathcal{N}_k} b_t(k,l) \cdot \tilde{\theta}_t^l, \tag{5}$$

where $c_t(k,l)$ and $b_t(k,l)$ are the aggregation weights assigned by agent k to agent l at time t for the critic and actor parameter aggregation respectively. These weights are normalized, i.e., $\sum_{l \in \mathcal{N}_k} c_t(k,l) = 1$, and $\sum_{l \in \mathcal{N}_k} b_t(k,l) = 1$. Two aggregation matrices C_t and B_t are formed using the element-wise weights, where $C_t(k,l) = c_t(k,l)$ and $B_t(k,l) = b_t(k,l)$. The assigned weights depend on the aggregation algorithm being used.

The objective of this this work are the following:

i) design a novel attack scheme, targeted to a specific node, for respective baseline aggregation methods used by the agents, such that the return of the target agent is minimized;
ii) implement the medoid and soft-medoid functions for aggregation of the parameters;
iii) validate the proposed schemes through analytical and simulation results.

So far we have considered an ideal scenario, where there is no adversary in the system. But there are instances when one or more of these agents are under attack, and normal agents do not know the identity of these agents. In such a scenario, the agents may receive compromised observation data, false rewards, or maliciously modified parameters from the neighbors, which causes their performance to be deteriorated. Here we design an attack on the shared parameters, which is presented next.

IV. Adversary Modeling and Attack Design

In this section, the proposed attack designs are presented. We design a *targeted attack*, which means that the adversary attacks one particular agent, called the *target agent*, denoted as $\tau \in [N]$. The attack is not direct; instead, it is realized by intercepting the information broadcast from another agent in the group, which is called the *victim agent*, denoted by

$\nu \in [N]$. Here, communication and information sharing have been utilized in designing the attack, the attacked signal being the actor and critic model parameter, as shared by the victim node. It is trivial to note that ν needs to be a neighbor of τ. Here we present strong attack models, where the adversary is assumed to have all possible information - the learning parameters used by the agents, the architecture of the actor and critic networks, the aggregation function, and the connectivity of the communication graph. Thus, the proposed model is an example of the *white-box attack*. The objective of the adversary is to make the target agent learn an adversarial policy, given by π^a. The adversary uses a virtual RL agent that learns this policy using the same network architecture and learning parameters as used by the agents. This virtual RL agent is denoted as $\mathcal{A}$. It can be assumed that the state-action value of the virtual agent, $Q_t^{\mathcal{A}}$, can be parametrized with parameter $w_t^{\mathcal{A}}$. In the same way, the policy function of the virtual agent can be parametrized with parameter $\theta_t^{\mathcal{A}}$. So, the simplified objective of the adversary would be to make the parameters of the target agent converge to the parameters of the virtual agent, i.e., $w_t^\tau \to w_t^{\mathcal{A}}$ and $\theta_t^\tau \to \theta_t^{\mathcal{A}}$ for $t \geq T_a$, where T_a denotes the onset of the attack.

The parameters shared by ν are modified to achieve the adversarial objective, and this would be different based on the aggregation method used by the agents. We present the attack models for two different cases, where the agents are using *average* and *median* based aggregation. The shared parameters include both the critic and actor parameters. However, due to space constraints, the design here is performed only for the critic parameters. The actor parameter modification follows the same procedure.

The convergence of critic parameters of τ to those of $\mathcal{A}$ can be quantified by the following cost function:

$$J_{\mathcal{A}} = \frac{1}{2}\mathbb{E}\{\left\|w_t^\tau - w_t^{\mathcal{A}}\right\|\}^2 \tag{6}$$

To minimize the cost function (6), the parameters of the target agent should be updated as per the following relation [19]:

$$w_{t+1}^\tau = w_t^\tau - \beta_{w,t}\big(w_t^\tau - w_t^{\mathcal{A}}\big), \tag{7}$$

where $\beta_{w,t} > 0$ is the step size. The adversary cannot regulate the actor-critic update of the target agent, so the adversarial update is modified as follows:

$$w_{t+1}^\tau = \tilde{w}_t^\tau - \beta_{w,t}\big(\tilde{w}_t^\tau - w_t^{\mathcal{A}}\big), \tag{8}$$

The adversary indirectly forces this update using the aggregation step. The parameters shared by the victim node are designed based on that objective.

A. Attack design for average aggregation

In average aggregation, the parameters of the target node are updated using the shared parameters from its neighbor nodes, given by

$$w_{t+1}^\tau = \frac{1}{n_\tau}\sum_{l\in\mathcal{N}_\tau}\tilde{w}_t^l = \frac{1}{n_\tau}\tilde{w}_t^\nu + \frac{1}{n_\tau}\sum_{l\in\mathcal{N}_\tau\setminus\nu}\tilde{w}_t^l \tag{9}$$

It is assumed that the adversary knows the values of $\tilde{w}_t^l$ for $l \in \mathcal{N}_\tau$, and has control over $\tilde{w}_t^\nu$. Comparing (8) and (9), it is implied that to achieve the adversarial objective, the attacked parameters shared by ν should be the following:

$${}_{att}\tilde{w}_t^\nu = n_\tau\left[\tilde{w}_t^\tau - \beta_{w,t}(\tilde{w}_t^\tau - w_t^{\mathcal{A}})\right] - \sum_{l\in\mathcal{N}_\tau\setminus\nu}\tilde{w}_t^l, \tag{10}$$

where ${}_{att}\tilde{w}_t^\nu$ is the shared parameters of the victim agent, under attack.

B. Attack design for median aggregation

The attack design for the median aggregation requires additional constraints to be satisfied, as presented next. The adversary modifies each entry of w_t^ν individually, but it also needs to make sure that the median for that particular coordinate corresponds to agent ν. Therefore, for median aggregation along the i^{th} entry in the critic parameter, the adversary needs

$$\begin{aligned}\sum_{l\in\mathcal{N}_\tau\setminus\nu} c_t^i(\tau,l)\cdot\tilde{w}_t^l(i) + c_t^i(\tau,\nu)\cdot\tilde{w}_t^\nu(i)& \\ = \tilde{w}_t^\tau(i) - \beta_{w,t}^i\big(\tilde{w}_t^\tau(i) - w_t^{\mathcal{A}}(i)\big)& \\ c_t^i(\tau,l) = 0, \quad l\in\mathcal{N}_\tau\setminus\nu& \\ c_t^i(\tau,\nu) = 1,&\end{aligned} \tag{11}$$

where $\tilde{w}_t^l(i)$ is the critic parameter of agent l at the i^{th} coordinate, $c_t(k,l)^i$ is the aggregation weight used for i^{th} coordinate, and $\beta_{w,t}^i$ is the step size of the i^{th} entry. The last two equations in (11) translate to the requirement that

$$\tilde{w}_t^\nu(i) = med\{\tilde{w}_t^l(i), l\in\mathcal{N}_\tau\} \tag{12}$$

The adversary now has the liberty of attacking each entry of the parameter independently. The median attack is given by

$${}_{att}\tilde{w}_t^\nu(i) = \tilde{w}_t^\tau(i) - \beta_{w,t}^i\big(\tilde{w}_t^\tau(i) - w_t^{\mathcal{A}}(i)\big), \tag{13}$$

when (12) is satisfied. Here ${}_{att}\tilde{w}_t^\nu(i)$ is the ith coordinate of modified value of the parameter shared by the victim node. We note that in (13), $\beta_{w,t}^i$ can be regulated by the adversary to satisfy the median requirement. Due to the constraint imposed, it is harder to attack the system when it uses median aggregation. However, the attack is less effective than the one designed for average aggregation. This supports the well-established fact that median is more resilient than average.

V. Medoid and Soft-medoid Based Parameter Aggregation

A medoid is a generalization of the notion of scalar median in the higher dimensions. Given a set of n data points $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$, where $x_i \in \mathbb{R}^m$, the medoid is given as,

$$Med(\mathbf{X}) = arg\min_{y\in\mathbf{X}}\sum_{j=1}^{n}\left\|x_j - y\right\|, \tag{14}$$

i.e., mediod is a point in the given data that minimized the sum of distances to the rest of the points. Generally, mediod is a very good approximation for the intuitive center of point cloud. However, when all the points in the data are farther away (for example when all points lie on the boundary of the convex hull), the mediod, being constrained to be one of the data points, will also be on extreme away from intuitive

center. To remedy this, a differentiable version of the medoid was proposed in [21], called *soft medoid*, that uses a softmax function and forms a weighted average of all the data points. This is defined as follows

$$SM(\mathbf{X}) = \sum_{i=1}^{n} s_i x_i, \tag{15}$$

where $0 \leq s_i \leq 1$ are the normalized weights, i.e., $\sum_i s_i = 1$, which are calculated using the distance between the data points:

$$s_i = \frac{\exp\Big(-\frac{1}{T} \sum_{j=1}^{n} \|x_j - x_i\| \Big)}{\sum_{q=1}^{n} \exp\Big(-\frac{1}{T} \sum_{j=1}^{n} \|x_j - x_i\| \Big)}, \tag{16}$$

where $T \in \mathbb{R}^+$ is the temperature parameter. For $T \to 0$, soft-medoid becomes the exact medoid, and $T \to \infty$ tends to calculate the average. Due to their known robustness properties, mediod and soft-mediod are ideal candidates for aggregation of parameters in MARL. The data points are the critic parameters w and the policy parameters θ. In the combination step, each agent can independently calculate the medoid or soft-medoid of the communicated parameters from the neighbors.

A. Medoid Based Parameter Aggregation

As discussed in Section III, agents share the actor and critic parameters with their respective neighbors. The combination step for the critic and actor parameters using medoid based aggregation can be described as,

$$w_{t+1}^k = arg \min_{\tilde{w}_t^j \in \mathcal{N}_k} \sum_{p \in \mathcal{N}_k} \left\| \tilde{w}_t^p - \tilde{w}_t^j \right\|, \tag{17}$$

and

$$\theta_{t+1}^k = arg \min_{\tilde{\theta}_t^j \in \mathcal{N}_k} \sum_{p \in \mathcal{N}_k} \left\| \tilde{\theta}_t^p - \tilde{\theta}_t^j \right\|, \tag{18}$$

respectively. This can also be expressed as

$$\begin{aligned} w_{t+1}^k &= \tilde{w}_t^{l_1} \\ \text{and} \quad \theta_{t+1}^k &= \tilde{\theta}_t^{l_2}, \end{aligned} \tag{19}$$

where $l_1, l_2 \in \mathcal{N}_k$ have the minimum loss in terms of critic and actor parameters, respectively. The losses are calculated as the sum of the difference of norms with parameters of every agent in the neighborhood of k. Thus, for the critic aggregation, $c_t(k, l_1) = 1$, and $c_t(k, p) = 0$ for $p \in \mathcal{N}_k$, $p \neq l_1$. Similarly, $b_t(k, l_2) = 1$, and $b_t(k, p) = 0$ for $p \in \mathcal{N}_k$, $p \neq l_2$. Thus, each row of the aggregation matrices C_t and B_t have only one element as 1, and the rest are all 0. Note that during a combination step using medoid, it is not necessary that the same neighboring agent needs to be chosen for both critic and actor aggregation. This flexibility essentially provides better resilience against attacks. Consider a scenario where two different agents are under attack, one sends malicious critic parameters, and the other shares malicious actor parameters. Due to the independence of the critic and actor aggregations, effectively only one agent can be considered to be adversarial.

B. Soft-medoid Based Parameter Aggregation

In this case, each agent combines the parameters received from the neighbors by a weighted average, in the forms of (4) and (5), where the weights are calculated using the same idea as in (16). The critic aggregation can thus be described as,

$$w_{t+1}^k = \sum_{l \in \mathcal{N}_k} \left(\frac{\exp\Big(-\frac{1}{T_c} \sum_{j \in \mathcal{N}_k} \left\| \tilde{w}_t^j - \tilde{w}_t^l \right\| \Big)}{\sum_{q \in \mathcal{N}_k} \exp\Big(-\frac{1}{T_c} \sum_{j \in \mathcal{N}_k} \left\| \tilde{w}_t^j - \tilde{w}_t^q \right\| \Big)} \right) \cdot \tilde{w}_t^l, \tag{20}$$

where T_c is the temperature parameter associated with the critic aggregation. In a similar way, the actor aggregation weights are defined as

$$\theta_{t+1}^k = \sum_{l \in \mathcal{N}_k} \left(\frac{\exp\Big(-\frac{1}{T_a} \sum_{j \in \mathcal{N}_k} \left\| \tilde{\theta}_t^j - \tilde{\theta}_t^l \right\| \Big)}{\sum_{q \in \mathcal{N}_k} \exp\Big(-\frac{1}{T_a} \sum_{j \in \mathcal{N}_k} \left\| \tilde{\theta}_t^j - \tilde{\theta}_t^q \right\| \Big)} \right) \cdot \tilde{\theta}_t^l, \tag{21}$$

where T_a is the temperature parameter associated with actor aggregation. Therefore, the elements of the aggregation matrices can be calculated as

$$\begin{aligned} c_t(k,l) &= \frac{\exp\Big(-\frac{1}{T_c} \sum_{j \in \mathcal{N}_k} \left\| \tilde{w}_t^j - \tilde{w}_t^l \right\| \Big)}{\sum_{q \in \mathcal{N}_k} \exp\Big(-\frac{1}{T_c} \sum_{j \in \mathcal{N}_k} \left\| \tilde{w}_t^j - \tilde{w}_t^q \right\| \Big)}, \\ b_t(k,l) &= \frac{\exp\Big(-\frac{1}{T_a} \sum_{j \in \mathcal{N}_k} \left\| \tilde{\theta}_t^j - \tilde{\theta}_t^l \right\| \Big)}{\sum_{q \in \mathcal{N}_k} \exp\Big(-\frac{1}{T_a} \sum_{j \in \mathcal{N}_k} \left\| \tilde{\theta}_t^j - \tilde{\theta}_t^q \right\| \Big)}. \end{aligned} \tag{22}$$

VI. Theoretical Analysis

This section presents the analytical results, including the convergence of the parameters and the overall performance improvement of the networked agents. Following that, we analyze the breakdown point of these aggregation methods, which provides a theoretical bound on the fraction of attacked agents for the healthy agents' parameters to stay bounded. As mentioned earlier, the action-value functions and the policy functions of each agent are parametrized, and both linear and nonlinear functions can be used to approximate these functions. It has been argued in the literature that the convergence of parameters with nonlinear function approximation using neural networks cannot be proven explicitly [39]. Here, we prove convergence with linear function approximations. In the next section, we validate our proposed method for nonlinear function approximations through simulation results. Stating the following assumptions is the key step to prove convergence.

Assumption 1. *For each agent k, the action-value function is parameterized as $Q_t^k(w) = w^\top \phi_w^k(s_t, a_t)$, where $\phi_w^k(s,a) = \left[\phi_{w,1}^k(s,a), \ldots, \phi_{w,d_w}^k(s,a)\right]^\top \in \mathbb{R}^{d_w}$ is the uniformly bounded feature vector for any $s \in \mathcal{S}, a \in \mathcal{A}$. In addition, the feature matrix $\Phi_w^k \in \mathbb{R}^{|\mathcal{S}|\cdot|\mathcal{A}| \times d_w}$ has full column rank, where the j-th column of Φ_w^k is $[\phi_{w,j}^k(s,a), s \in \mathcal{S}, a \in \mathcal{A}]^\top$, for $j \in [d_w]$. Also, for any $u \in \mathbb{R}^{d_w}$, $\Phi_w^k u \neq \mathbb{1}$.*

Assumption 2. *For each agent k, the policy function is modeled by a Gaussian function $\pi_{\theta_k}(s,a) = \frac{1}{\sigma_k\sqrt{2\pi}}\exp(-\frac{(a-\mu_k(s))^2}{2\sigma_k^2})$ with $\mu_k(s) = \theta_k^\top \phi_\pi^k(s)$ as the mean, $\sigma_k > 0$ the constant variance, and $\phi_\pi^k(s) \in \mathbb{R}^{d_\theta}$ the uniformly bounded feature vector.*

The standard practice of proving convergence of parameters in an actor-critic algorithm is by using two-time-scale stochastic method [7], where the convergence of the critic step is first analyzed on a faster time scale, assuming that the policy is fixed. Following that, the convergence of the policy function is analyzed, while assuming that the critic parameters have already converged. The convergence analysis uses some standard assumptions from the literature, which are listed below [7].

Assumption 3. *The transition matrix of the Markov chain $\{s_t\}_{t\geq 0}$ induced by policy π_{θ_k}, i.e., $P^{\theta_k}(s'|s) = \sum_{a\in\mathcal{A}}\pi_{\theta_k}(s,a)\cdot P^k(s'|s,a), \forall s,s' \in \mathcal{S}$, is irreducible and aperiodic under any π_{θ_k} for $k \in [N]$. Further, $\pi_{\theta_k}(s,a) > 0$ for any $s \in \mathcal{S}$, $a \in \mathcal{A}$, θ_k, and $\pi_{\theta_k}(s,a)$ is continuously differentiable with respect to θ_k.*

Assumption 4. *The instantaneous reward r_t^k is uniformly bounded for any $k \in [N]$ and $t \geq 0$.*

Assumption 5. *For every normal agent $k \in [N]$, the step-sizes $\beta_{w,t}^k$ and $\beta_{\theta,t}^k$ satisfy*

$$\sum_t \beta_{w,t}^k = \sum_t \beta_{\theta,t}^k = \infty, \quad \sum_t {\beta_{w,t}^k}^2 + {\beta_{\theta,t}^k}^2 < \infty,$$
$$\frac{\beta_{\theta,t}^k}{\beta_{w,t}^k} \to 0, \quad \lim_{t\to\infty} \beta_{w,t+1}^k \cdot {\beta_{w,t}^k}^{-1} = 1.$$

Assumption 6. *Through the entire history of the algorithm, θ_t^k belongs to a compact set for all k and t. And this compact set contains at least one local optimum of the problem* (1).

A. Convergence Analysis

One way of proving convergence is by analyzing the gradients of critic and policy functions. Define $G_{k,t}^w(w) \triangleq \delta_t^k \cdot \nabla_w Q(s_t,a_t;w)$ and $G_{k,t}^\theta(\theta) \triangleq Q_t^k(w_t^k)\cdot\nabla_\theta \log\pi_\theta(s_t,a_t)$ as the gradient of the action-value and policy functions for agent k at time t. Under Assumptions 1-2, by simple mathematical manipulation, the gradients can be parametrized as

$$G_{k,t}^w(w) = \gamma_t^k + w^\top \phi_t^k, \quad G_{k,t}^\theta(\theta) = \xi_t^k - \theta^\top \zeta_t^k. \tag{23}$$

where $\gamma_t^k \triangleq \left(r_{t+1}^k - \mu_t^k\right)\cdot\phi_w^k(s_t,a_t)$,
$\phi_t^k \triangleq \left(\phi_w^k(s_{t+1},a_{t+1}) - \phi_w^k(s_t,a_t)\right)\cdot\phi_w^k(s_t,a_t)$,
$\xi_t^k \triangleq a_t\phi_\pi^k(s_t)\cdot\frac{Q_t^k(w_t^k)}{\sigma_k^2}$, $\zeta_t^k \triangleq \phi_\pi^k(s_t)\phi_\pi^k(s_t)\cdot\frac{Q_t^k(w_t^k)}{\sigma_k^2}$.

Theorem 1. *Under Assumptions 1, 3-5, for any policy π_θ, with $\{w_t^k\}$ generated from* (2) *and aggregated using the medoid rule* (17) *or soft-medoid rule* (20)*, we have $\lim_{t\to\infty} w_t^k = \chi_k(\theta)$ almost surely for all $k \in [N]$.*

Proof. Given the critic aggregation step from (4), the gradient of critic function can be written as

$$G_{k,t}^w(w_{t+1}^k) = \gamma_t^k + \sum_{l\in\mathcal{N}_k} c_t(k,l)\cdot(\tilde{w}_t^l)^\top\phi_t^k.$$

In the case of medoid aggregation, for an agent k, the coefficient equals 1 for only one neighboring agent, and 0 for the rest. In the case of soft-medoid, we have $\sum_{l\in\mathcal{N}_k} c_t(k,l) = 1$. Therefore, we can rewrite the gradient as

$$\begin{aligned} G_{k,t}^w(w_{t+1}^k) &= \sum_{l\in\mathcal{N}_k} c_t(k,l)\cdot\left(\gamma_t^k + (\tilde{w}_t^l)^\top\phi_t^k\right) \\ &= \sum_{l\in\mathcal{N}_k} c_t(k,l) G_{k,t}^w(\tilde{w}_t^l) \end{aligned}$$

Computing norms on both sides, we get

$$\left\|G_{k,t}^w(w_{t+1}^k)\right\| \leq \sum_{l\in\mathcal{N}_k} c_t(k,l)\left\|G_{k,t}^w(\tilde{w}_t^l)\right\|.$$

It has been shown in the Theorem IV.12 of [7] that with the standard actor-critic algorithm as given in (2) and (3), the critic parameters converge almost surely, for all $k \in [N]$, i.e., $\lim_{t\to\infty}\tilde{w}_t^k = \chi_k(\theta)$ a.s., without cooperation. This essentially means that the gradient norm corresponding to the critic parameters converge almost surely for all agents, i.e., $\lim_{t\to\infty}\|G_{k,t}(\tilde{w}_k^t)\| = 0$ a.s. As the aggregation weights $c_t(k,l)$ are finite for all $l \in \mathcal{N}_k$, and $\sum_{l\in\mathcal{N}_k} c_t(k,l) = 1$, in the case of both medoid and soft-medoid based aggregations, we conclude that $\lim_{t\to\infty}\|G_{k,t}(w_k^t)\| = 0$ a.s., and thus $\lim_{t\to\infty} w_t^k = \chi_k(\theta)$ a.s. □

Theorem 2. *Under Assumptions 2, 3-6, with the policy parameters $\{\theta_t^k\}$ generated from* (3) *and aggregation using the medoid rule in* (18) *or the soft-medoid rule in* (21)*, θ_t^k converges almost surely to a point in the set Λ_k for any $k \in [N]$.*

Proof. The proof of this theorem follows similar steps as that of Theorem 1. Given the actor aggregation step (5), the actor gradient norm can be expressed as

$$\|G_{k,t}^\theta(\theta_{t+1}^k)\| \leq \sum_{l\in\mathcal{N}_k} b_t(k,l)\cdot\|G_{k,t}^\theta(\tilde{\theta}_t^k)\|.$$

Using the same arguments as in Theorem 1, we may write

$$\left\|G_{k,t}^\theta(\theta_{t+1}^k)\right\| \leq \sum_{l\in\mathcal{N}_k} b_t(k,l)\left\|G_{k,t}^\theta(\tilde{\theta}_t^l)\right\|$$

Literature [7] shows that without cooperation, the actor parameters converge to a point in the set Λ_k almost surely for all $k \in [N]$. As a result, $\lim_{t\to\infty}\|G_{k,t}^\theta(\tilde{\theta}_t^k)\| = 0$ a.s. Using similar arguments about the aggregation weights $b_t(k,l)$, we get $\lim_{t\to\infty}\|G_{\theta,t}^{k,k}\| = 0$ a.s. Thus, θ_t^k converges a.s. to a point in the set Λ_k. □

B. Learning Performance

It is often argued that cooperative MARL improves learning performance. We analytically show that the overall performance of the group of agents improves when they cooperate. In the context of this work, we show the performance improvement for medoid and soft-medoid based aggregation. The performance improvement is evaluated based on how close the estimated parameters of the critic and actor networks are to their respective optimal values. For the critic parameter estimation

of agent k, the estimation performance is measured by the variable $\Delta^Q(s_t^k, a_t^k; w_t^k)$, which is defined as

$$\Delta^Q(s_t^k, a_t^k; w_t^k) = |Q(s_t^k, a_t^k; w_t^k) - Q(s_t^k, a_t^k; w^*)|,$$

where w^* is the optimal critic parameter values. Here $\Delta^Q(s_t^k, a_t^k; w_t^k) \in \mathbb{R}^+$ measures how close the estimate of the Q-value is to its optimal value, when the actual critic parameter estimate is used for calculation. To evaluate the performance of the network before and after the aggregation step, we define the following variables:

$$(\Delta_{t+1}^Q)^{all} = \left[\Delta^Q(s_t^1, a_t^1; w_{t+1}^1) \cdots \Delta^Q(s_t^N, a_t^N; w_{t+1}^N)\right] \in \mathbb{R}^N$$
$$(\Delta_t^Q)^{all} = \left[\Delta^Q(s_t^1, a_t^1; \tilde{w}_t^1) \cdots \Delta^Q(s_t^N, a_t^N; \tilde{w}_t^N)\right] \in \mathbb{R}^N$$

where $(\Delta_{t+1}^Q)^{all}$ and $(\Delta_t^Q)^{all}$ are measures of the estimation errors of the critic parameters after and before the aggregation step, respectively.

Theorem 3. *Consider the distributed actor-critic algorithm of N cooperating agents with critic parameter update given by* (2)*. The agents use model aggregation of critic parameters, as given by* (17) *(for medoid) or* (20) *(for soft-medoid), depending on the method of aggregation. In both of these cases, the aggregation step results in the critic parameters move closer to their optimal values, i.e.,*

$$\left\|(\Delta_{t+1}^Q)^{all}\right\| \leq \left\|(\Delta_t^Q)^{all}\right\|. \tag{24}$$

Proof. From the definition of $\Delta^Q(s_t^k, a_t^k; w_t^k)$ and the critic aggregation rule (4), we may write

$$\begin{aligned}\Delta^Q(s_t^k, a_t^k; w_{t+1}^k) &= \left|Q(s_t^k, a_t^k; w_{t+1}^k) - Q(s_t^k, a_t^k; w^*)\right| \\ &= \left|\left(w_{t+1}^k - w^*\right)^T \phi_w^k(s_t^k, a_t^k)\right| \\ &= \left|\left(\sum_{l\in\mathcal{N}_k} c_t(k,l)\cdot \tilde{w}_t^l - w^*\right)^T \phi_w^k(s_t^k, a_t^k)\right| \\ &\leq \sum_{l\in\mathcal{N}_k} c_t(k,l)\cdot \Delta^Q(s_t^k, a_t^k; \tilde{w}_t^l).\end{aligned} \tag{25}$$

Using this inequality, we may write

$$(\Delta_{t+1}^Q)^{all} \leq C_t(\Delta_t^Q)^{all}, \tag{26}$$

where $C_t \in \mathbb{R}^{N\times N}$ is the critic aggregation matrix of the network of agents. using norm operator in (26), we get

$$\left\|(\Delta_{t+1}^Q)^{all}\right\| \leq \rho(C_t)\left\|(\Delta_t^Q)^{all}\right\|, \tag{27}$$

where $\rho(C_t) = \lambda_{max}(C_t)$ is the spectral radius of the critic aggregation matrix, given by the largest absolute value of its eigenvalues. As shown earlier, in the case of both medoid and soft-medoid based aggregation, the critic aggregation matrix C_t is a square matrix with non-negative real values, whose each row sums up to 1. Thus, the matrix C_t is a row-stochastic matrix for all $t \geq t_0$, in the case of medoid and soft-medoid based aggregations. Using Perron-Frobenius theorem, it can be concluded that the spectral radius of C_t is bounded by 1, which implies that $\left\|(\Delta_{t+1}^Q)^{all}\right\| \leq \left\|(\Delta_t^Q)^{all}\right\|$. □

From Assumption 2, we use the quantity $\mu_k(s)$ to evaluate the estimate of the policy parameter θ. The proximity of the estimated policy parameter of agent k is expressed by the variable $\Delta^\Theta(s_t^k; \theta_t^k)$, which is defined as $\Delta^\Theta(s_t^k; \theta_t^k) = \left|\mu(s_t^k; \theta_t^k) - \mu(s_t^k; \theta^*)\right|$, where θ^* is the optimal actor parameter. Here $\Delta^\Theta(s_t^k; \theta_t^k)$ measures how close the current policy chosen by agent k is to the optimal policy. To evaluate the performance of the actor network of all the agents in the network, we define the following two variables

$$(\Delta_{t+1}^\Theta)^{all} = \left[\Delta^\Theta(s_t^1; \theta_{t+1}^1) \cdots \Delta^\Theta(s_t^N; \theta_{t+1}^N)\right] \in \mathbb{R}^N$$
$$(\Delta_t^\Theta)^{all} = \left[\Delta^Q(s_t^1; \tilde{\theta}_t^1) \cdots \Delta^\Theta(s_t^N; \tilde{\theta}_t^N)\right] \in \mathbb{R}^N$$

where $(\Delta_{t+1}^\Theta)^{all}$ and $(\Delta_t^\Theta)^{all}$ are measures of the estimation errors of the actor parameters after and before the aggregation step, respectively.

Theorem 4. *Consider the distributed actor-critic algorithm of N cooperating agents with actor parameter update* (3)*. The agents use model aggregation of actor parameters, which is given by* (18) *or* (21)*, for medoid an d soft-medoid, respectively. For both of these methods, the aggregation step results in the actor parameters move closer to their optimal values, i.e.,*

$$\left\|(\Delta_{t+1}^\Theta)^{all}\right\| \leq \left\|(\Delta_t^\Theta)^{all}\right\|. \tag{28}$$

Proof. From the definition of $\Delta^\Theta(s_t^k; \theta_t^k)$ and the actor aggregation rule (5), following similar steps as earlier, we get

$$\Delta^\Theta(s_t^k; \theta_{t+1}^k) \leq \sum_{l\in\mathcal{N}_k} b_t(k,l)\cdot \Delta^\Theta(s_t^k; \tilde{\theta}_t^l). \tag{29}$$

Defining the coefficient matrix for actor aggregation as $B_t \in \mathbb{R}^{N\times N}$, where $B_t(i,j) = b_t(i,j)$, the inequality (29) implies

$$(\Delta_{t+1}^\Theta)^{all} \leq B_t(\Delta_t^\Theta)^{all}, \tag{30}$$

Using norm operator in (26), we get

$$\left\|(\Delta_{t+1}^\Theta)^{all}\right\| \leq \rho(B_t)\left\|(\Delta_t^\Theta)^{all}\right\|, \tag{31}$$

where $\rho(B_t) = \lambda_{max}(B_t)$ is the spectral radius of the actor aggregation matrix. Following the same arguments as earlier, where it can be shown that in the case of both medoid and soft-medoid based aggregations, B_t is a row-stochastic matrix, thus $\rho(B_t) \leq 1$ for all t. This implies that $\left\|(\Delta_{t+1}^\Theta)^{all}\right\| \leq \left\|(\Delta_t^\Theta)^{all}\right\|$. □

The performance improvement as shown here is based on a cumulative scale, meaning that the sum of the parameter estimation error of all agents is non-increasing due to cooperation, which does not necessarily mean that individual performances are improved for all the agents. Although, the inequality in Theorems 3 and 4 does not guarantee performance improvement, this is the standard practice for convergence analysis [40].

C. Robustness Analysis

In this work, the adversary targets one node to attack, and the attack is performed through another node. But owing to the communication and parameter sharing between the agents, the performance of some other nodes get affected too, depending

on the structure of the graph. Sequentially, at a certain point, an agent may find that more than one of its neighbors are driven by malicious information. This makes it important to find out how the percentage of malicious neighbors affects the performance. In a network of N connected agents, breakdown point analysis of an agent k is to find out the minimum fraction ϵ for which the parameters estimated by agent k remains bounded in the presence of ϵN attacked agents [21]. In the MARL setup, we often do not consider a fully connected graph, so the breakdown point analysis is performed based on the number of neighbors. Formal definition of the breakdown point is presented below.

Definition 1. *[41] The breakdown point of an estimator $\boldsymbol{T}$ of a collection $\mathbf{X}$ is defined as the smallest fraction m/n of outliers that can produce an unbounded estimate*

$$\epsilon^*(\boldsymbol{T}, \boldsymbol{X}) = \min_{1 \leq m \leq n} \{ \frac{m}{n} : \sup_{\boldsymbol{Y}_m} \|\boldsymbol{T}(\boldsymbol{X}) - \boldsymbol{T}(\boldsymbol{Y}_m)\| = \infty \}, \quad (32)$$

where the supremum is taken over all possible corrupted collections $\boldsymbol{Y}_m$ that are obtained by replacing m data points of $\boldsymbol{X}$ by arbitrary values.

In this work, we are interested in the breakdown point analysis of medoid and soft-medoid aggregation. In actor-critic algorithm, even though the aggregation is done separately for critic and actor, each agent interacts with the same neighbors in both these cases. Here the analysis is presented for critic parameters, the same result holds for the actor parameters.

Let the neighborhood of agent k, denoted as $\mathcal{N}_k$ be divided into two disjoint sets, $\mathcal{N}_k^{(h)}$ and $\mathcal{N}_k^{(a)}$, where $\mathcal{N}_k^{(h)}$ denotes the set of healthy neighbors of k, and $\mathcal{N}_k^{(a)}$ is the set of attacked neighbors. Clearly, $\mathcal{N}_k^{(h)} \bigcup \mathcal{N}_k^{(a)} = \mathcal{N}_k$ and $\mathcal{N}_k^{(h)} \bigcap \mathcal{N}_k^{(a)} = \emptyset$. The following theorem shows that the aggregated critic parameter of agent k remains bounded even when the parameters shared by the attacked agents have an infinite norm, provided there is a majority of healthy agents in its neighborhood, i.e., $\left|\mathcal{N}_k^{(h)}\right| \geq \left|\mathcal{N}_k^{(a)}\right|$.

Theorem 5. *Consider the distributed actor-critic algorithm of N cooperating agents with critic parameter update given by* (2)*. The agents aggregate their critic parameters by* (17) *or* (20)*, based on the method used. Some of the agents in the network are under attack and share adversarial values for parameter $\tilde{w}_t$. Then for each healthy agent $k \in [N]$, the critic parameter aggregation has a finite breakdown point of $\epsilon_{SM}^k(\tilde{w}_t^l, l \in \mathcal{N}_k) = \frac{1}{n_k} \left\lfloor \frac{(n_k+1)}{2} \right\rfloor$.*

Proof. The analysis of breakdown point is performed considering the worst-case scenario. To calculate the breakdown point of soft-medoid based aggregation, we need to find the minimal fraction of attacker neighbors of agent k such that $\left\|w_{t+1}^k\right\| < \infty$ does not hold anymore, given $\left\|\tilde{w}_t^l\right\| \to \infty$, $l \in \mathcal{N}_k^{(a)}$, where the aggregated parameters w_{t+1}^k are given as (20). Under this scenario, the aggregated parameters can remain bounded if and only if the soft-medoid based aggregation weights associated with the attacked agents converge to zero, i.e., $c_t(k, l_1) \to 0$, for $l_1 \in \mathcal{N}_k^{(a)}$.

Using (22), we may write

$$\frac{c_t(k,l_1)}{c_t(k,l_2)} = \frac{\exp\Big(-\frac{1}{T_c} \sum\limits_{j \in \mathcal{N}_k} \left\| \tilde{w}_t^j - \tilde{w}_t^{l_1} \right\| \Big)}{\exp\Big(-\frac{1}{T_c} \sum\limits_{j \in \mathcal{N}_k} \left\| \tilde{w}_t^j - \tilde{w}_t^{l_2} \right\| \Big)}$$

$$= \exp\left\{ -\frac{1}{T_c} \left(\sum_{j \in \mathcal{N}_k} \left\| \tilde{w}_t^j - \tilde{w}_t^{l_1} \right\| - \sum_{j \in \mathcal{N}_k} \left\| \tilde{w}_t^j - \tilde{w}_t^{l_2} \right\| \right) \right\},$$

where $l_1 \in \mathcal{N}_k^{(a)}$ and $l_2 \in \mathcal{N}_k^{(h)}$. It has been shown in [42] that the worst-case perturbation is obtained when the perturbation is concentrated on a point mass. Following that, we assume that the perturbed parameters of the attacked agents have only one entry that is infinitely large, and all other entries are zero. Thus, $\left\| \tilde{w}_t^{l_1} \right\| = m$, where $m \to \infty$. Without loss of generality, we further assume that all the healthy agents' parameters have all entries as zero. With these, we get

$$\frac{c_t(k,l_1)}{c_t(k,l_2)} = \exp\left\{ -\frac{1}{T_c} \left[\left(\sum_{j \in \mathcal{N}_k^{(h)}} m \right) - \left(\sum_{j \in \mathcal{N}_k^{(a)}} m \right) \right] \right\},$$

where $m \to \infty$, $l_1 \in \mathcal{N}_k^{(a)}$ and $l_2 \in \mathcal{N}_k^{(h)}$. If the number of attacked neighbors is less than the number of healthy neighbors, i.e., $\left|\mathcal{N}_k^{(a)}\right| < \left|\mathcal{N}_k^{(h)}\right|$, then we have $\lim\limits_{m \to \infty} \frac{c_t(k,l_1)}{c_t(k,l_2)} = 0$. Thus, the aggregated parameters using the soft-medoid based aggregation remains bounded even when the perturbed parameters are placed at infinity, provided we have a majority of healthy neighbors. The breakdown point of soft-medoid based aggregation is thus given as $\frac{1}{n_k} \left\lfloor \frac{(n_k+1)}{2} \right\rfloor$. □

The breakdown point for the medoid-based aggregation is the same as the soft-medoid case. It is intuitive to verify that and also follows from the fact that medoid is the generalization of median in higher dimensions, which also has a breakdown point of 0.5. The analysis considers the scenario when the perturbation tends to infinity. However, in practice, the adversary limits the perturbation to the minimum, and thus the weights associated with the attacked agents do not converge to zero.

VII. Evaluation

In this section, we present the evaluation results for a popular off-policy actor-critic algorithm, DDPG [43], applied to MuJoCo continuous control tasks [44] of HalfCheetah and InvertedPendulum2d through the OpenAI Gym interface [45]. We consider a network of $N = 8$ agents, and the average neighborhood size is $\frac{1}{N} \sum\limits_i \mathcal{N}_i = 2.5$. The set of neighbors of the agents is listed as follows: {0, 5, 6}, {1, 3}, {2, 4, 7}, {3, 1, 5}, {4, 2}, {5, 0, 3}, {6, 0}, and {7, 2}. Here each agent in the group performs the same task in independent environments. We compare the medoid and soft-medoid based aggregation methods with three baseline methods: i) no cooperation among agents, ii) aggregation using average, and iii) aggregation using point-wise median. The actor and the critic networks of each agent are assumed to be neural networks (NN) with two hidden layers, containing 400 and 300 neurons, respectively. The

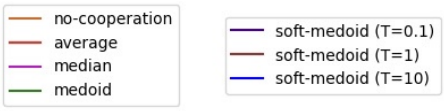
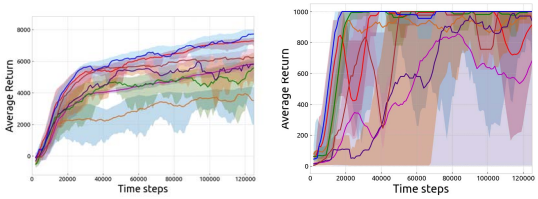


Fig. 1: The legends used in the plots.

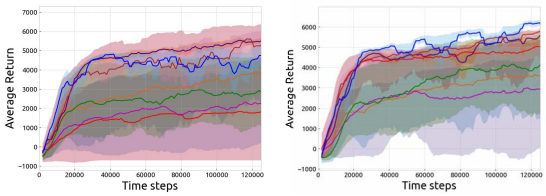


(a) Task: HalfCheetah (b) Task: Inverted-pendulum

Fig. 2: Average reward of the group for various aggregation methods.

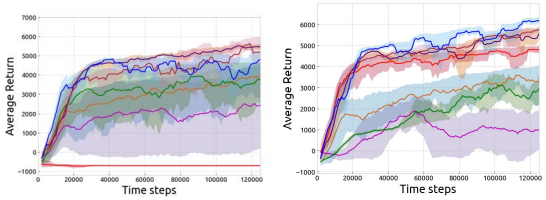
soft-medoid based aggregation method is simulated for three different values of the temperature variable: 0.1, 1, and 10. All the figures shown here follow the legends given in Fig. 1.

The average reward of the group for various aggregation methods is plotted in Fig. 2. It can be verified that the soft-medoid method attains the highest average reward among all the agents. Following the normal scenario, now to validate the proposed attack design, we simulate the average-based attack and the median-based attack. The targeted node is considered to be Agent 0 (the numbering starts at 0), and the attacked node is Agent 6. For both the attacks, it is assumed that the adversary knows the architecture of networks, all the hyper-parameters used in learning, the structure of the graph, etc., and utilizes all this information in running the virtual attacker agent. At each instant, the adversary calculates the values of the malicious NN weights, which is used to replace the communicated information from Agent 6. The average reward of the group performing the task of HalfCheetah is plotted in Fig. 3 for both these attacks. It can be verified from the plots that the targeted aggregation under the particular attacks performs poorly. The soft-medoid method results in the best performance in both these attack scenarios. As explained earlier, the attack is propagated through the network due to coordination among agents. The list of neighbors suggests that there exists a subgraph consisting of the nodes $\{0, 1, 3, 5, 6\}$ that includes both the target and the



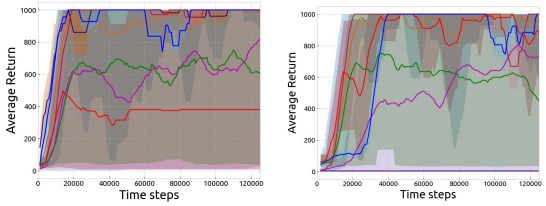
(a) average attack (b) median attack

Fig. 3: Average reward of the group under attack (task: HalfCheetah).



(a) average attack (b) median attack

Fig. 4: Average reward of agents 0,1,3,5,6 under attack (HalfCheetah).



(a) average attack (b) median attack

Fig. 5: Average reward of the group under attack (Inverted-pendulum).

victim agents. To better demonstrate the effect of the attacks, we plot the average reward of these 5 agents, and it is shown in Fig. 4. It can be observed that these agents perform very poorly with the aggregation methods which is targeted by the respective attacks. The results in this section confirm our claim that the average-based attack is stronger than the median-based attack. The average reward of all the agents performing the Inverted-pendulum task, under the two designed attacks are plotted in Fig. 5.

VIII. Conclusion

This paper deals with resilient aggregation in multi-agent reinforcement learning by using medoid and soft-medoid as the aggregation protocols. The attack strategies presented here cannot be used to design attacks against these aggregation methods, which makes them resilient to an extent against the designed attacks. However, the simulation results show that the performance while using medoid aggregation deteriorates against such attacks, while soft-medoid outperforms all other methods. The temperature hyper-parameter in soft-medoid method influences the performance, as depicted in the results. The breakdown point analysis for medoid and soft-medoid assures that the aggregated parameters will remain bounded; however, it does not evaluate the aggregation performance in the presence of these attacked agents compared to the no-attack case. Design of more sophisticated attacks against medoid and soft-medoid attack, and the improvement of performance by using adaptive weights in these aggregation methods are some of the directions for extension of this work.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[3] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[4] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[5] P. Kofinas, A. Dounis, and G. Vouros, "Fuzzy q-learning for multi-agent decentralized energy management in microgrids," *Applied energy*, vol. 219, pp. 53–67, 2018.

[6] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "Drn: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, pp. 167–176, 2018.

[7] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning*, pp. 5872–5881, PMLR, 2018.

[8] J. Li, F. Cai, and X. Koutsoukos, "Byzantine resilient aggregation in distributed reinforcement learning," in *International Symposium on Distributed Computing and Artificial Intelligence*, pp. 56–66, Springer, 2021.

[9] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[10] S. V. Macua, J. Chen, S. Zazo, and A. H. Sayed, "Distributed policy evaluation under multiple behavior strategies," *IEEE Transactions on Automatic Control*, vol. 60, no. 5, pp. 1260–1274, 2014.

[11] D. Jin, J. Chen, C. Richard, J. Chen, and A. H. Sayed, "Affine combination of diffusion strategies over networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2087–2104, 2020.

[12] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv preprint arXiv:1706.02275*, 2017.

[13] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, *et al.*, "Massively parallel methods for deep reinforcement learning," *arXiv preprint arXiv:1507.04296*, 2015.

[14] S. Kar, J. M. Moura, and H. V. Poor, "QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through *Consensus + Innovations*," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, 2013.

[15] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun, "Tactics of adversarial attack on deep reinforcement learning agents," *arXiv preprint arXiv:1703.06748*, 2017.

[16] J. Liu, W. Niu, J. Liu, J. Zhao, T. Chen, Y. Yang, Y. Xiang, and L. Han, "A method to effectively detect vulnerabilities on path planning of vin," in *International Conference on Information and Communications Security*, pp. 374–384, Springer, 2017.

[17] Y. Xiang, W. Niu, J. Liu, T. Chen, and Z. Han, "A pca-based model to predict adversarial examples on q-learning of path finding," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 773–780, IEEE, 2018.

[18] Y. Lin, S. Gade, R. Sandhu, and J. Liu, "Toward resilient multi-agent actor-critic algorithms for distributed reinforcement learning," in *2020 American Control Conference (ACC)*, pp. 3953–3958, IEEE, 2020.

[19] J. Li, W. Abbas, and X. Koutsoukos, "Resilient distributed diffusion in networks with adversaries," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 1–17, 2019.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

[21] S. Geisler, D. Zügner, and S. Günnemann, "Reliable graph neural networks via robust aggregation," *arXiv preprint arXiv:2010.15651*, 2020.

[22] F. A. Oliehoek, M. T. Spaan, and N. Vlassis, "Optimal and approximate q-value functions for decentralized pomdps," *Journal of Artificial Intelligence Research*, vol. 32, pp. 289–353, 2008.

[23] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International Conference on Machine Learning*, pp. 4295–4304, PMLR, 2018.

[24] C. Schroeder de Witt, J. Foerster, G. Farquhar, P. Torr, W. Boehmer, and S. Whiteson, "Multi-agent common knowledge reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 9927–9939, 2019.

[25] F. Christianos, L. Schäfer, and S. V. Albrecht, "Shared experience actor-critic for multi-agent reinforcement learning," *arXiv preprint arXiv:2006.07169*, 2020.

[26] D. Maravall, J. de Lope, and R. Domínguez, "Coordination of communication in robot teams by reinforcement learning," *Robotics and Autonomous Systems*, vol. 61, no. 7, pp. 661–666, 2013.

[27] T. Chu, S. Chinchali, and S. Katti, "Multi-agent reinforcement learning for networked system control," *8th International Conference on Learning Representations (ICLR)*, 2020.

[28] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[29] S. V. Macua, A. Tukiainen, D. G.-O. Hernández, D. Baldazo, E. M. de Cote, and S. Zazo, "Diff-dac: Distributed actor-critic for multitask deep reinforcement learning," *arXiv preprint arXiv:1710.10363*, 2017.

[30] W. Liu, P. Zhuang, H. Liang, J. Peng, and Z. Huang, "Distributed economic dispatch in microgrids based on cooperative reinforcement learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2192–2203, 2018.

[31] I. Ilahi, M. Usama, J. Qadir, M. U. Janjua, A. Al-Fuqaha, D. T. Huang, and D. Niyato, "Challenges and countermeasures for adversarial attacks on deep reinforcement learning," *IEEE Transactions on Artificial Intelligence*, 2021.

[32] T. Chen, J. Liu, Y. Xiang, W. Niu, E. Tong, and Z. Han, "Adversarial attack and defense in reinforcement learning-from ai security view," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[33] M. Figura, K. C. Kosaraju, and V. Gupta, "Adversarial attacks in consensus-based multi-agent reinforcement learning," *arXiv preprint arXiv:2103.06967*, 2021.

[34] J. Lin, K. Dzeparoska, S. Q. Zhang, A. Leon-Garcia, and N. Papernot, "On the robustness of cooperative multi-agent reinforcement learning," in *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 62–68, IEEE, 2020.

[35] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, pp. 5650–5659, PMLR, 2018.

[36] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[37] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 118–128, 2017.

[38] R. Guerraoui, S. Rouault, *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *International Conference on Machine Learning*, pp. 3521–3530, PMLR, 2018.

[39] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[40] Y. Zhang and M. M. Zavlanos, "Distributed off-policy actor-critic reinforcement learning with policy consensus," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4674–4679, IEEE, 2019.

[41] H. P. Lopuhaa and P. J. Rousseeuw, "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," *The Annals of Statistics*, pp. 229–248, 1991.

[42] C. Croux, G. Haesbroeck, and P. J. Rousseeuw, "Location adjustment for the minimum volume ellipsoid estimator," *Statistics and Computing*, vol. 12, no. 3, pp. 191–200, 2002.

[43] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[44] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, IEEE, 2012.

[45] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.