2022 30th Mediterranean Conference on Control and Automation (MED)
June 28 - July 1, 2022. Athens, Greece

2022 30th Mediterranean Conference on Control and Automation (MED) | 978-1-6654-0673-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/MED54222.2022.9837237

Attack-Resilient Multi-Agent Flocking Control Using Graph Neural Networks

Chandreyee Bhowmick, Mudassir Shabbir, and Xenofon Koutsoukos
Institute for Software Integrated Systems, Vanderbilt University, Nashville, USA
Email: {chandreyee.bhowmick, mudassir.shabbir, xenofon.koutsoukos}@vanderbilt.edu

***Abstract*—Flocking control of a group of mobile agents has been recently investigated using Graph Convolution Networks (GCNs). The design relies on training using a centralized controller but the resulting GCN controller is based on communication between the agents. The agents receive sensor measurements which are incorporated into the states and shared between the neighbors. However, the paradigm is prone to adversarial attacks. In this paper, we consider the problem of designing GCN-based distributed flocking control that is resilient to attacks on the communicated information. We consider an attack model that is used to compromise the inter-agent communication and may inject arbitrary signals. Our control design uses a coordinate-wise median-based aggregation function. It is shown that the GCN-based controller using the proposed aggregation method is resilient against attacks on the communication between the agents, whereas the typical average-based aggregation fails to maintain the flock structure. Robustness analysis is performed to show that the proposed method is resilient whenever a majority of the agents in the neighborhood can be trusted. Simulation results and analysis are presented that validate the merits of the proposed approach.**

***Index Terms*—Flocking control, graph convolutional network, imitation learning, resilient aggregation.**

I. Introduction

A multi-agent system can be described as a group of intelligent units capable of communicating with each other and cooperatively implementing different types of tasks [1]. This can be realized by various types of robots and have wide application areas including military surveillance, exploration, and rescue operation. Flocking control is one such behavior designed for multi-agent systems, where the agents move at the same velocity while avoiding collision with each other. Design of flocking control has received a lot of research attention due to its wide range of applications, including connected automated vehicles [2], fleet of ground robots [3], and others.

Flocking and other coordination-based tasks can be performed using a centralized controller, which receives sensor data from all the agents, and sends back the appropriate control action to each of these agents. Such a centralized controller can achieve the optimal performance, but it suffers from scalability and point-of-failure issues. Therefore, for a large group of agents, distributed control becomes a necessity. In such a control scheme, the agents communicate with their neighbors to decide on the suitable control actions. However, the design of an optimal controller in such a setting is a challenging problem [4]. Some of these challenges stem from the fact that in a distributed setting, the agents can only communicate with their immediate neighbors. This issue can be overcome by the use of *imitation learning* [5]. The idea has been employed in applications of single-agent system such as autonomous driving [6] and quadrotor navigation [7]. However, in multi-agent system with swarms, this learning becomes challenging as the number of agents becomes large.

Imitation learning can be effectively employed using Graph Convolutional Networks (GCNs) [8], [9]. GCNs are highly suitable for control of large-size multi-agent systems, as their architecture is based on information exchange using the underlying topology. For a known communication graph, this architecture can be used in large-scale networks to incorporate information from multi-hop neighbors. This improves the performance of the designed control significantly over local controllers that rely only on the information received from the immediate neighbors. The work in [10] develops a distributed control algorithm using GCNs, where information from multi-hop neighbors are aggregated using simple averaging.

While cooperation among agents can improve learning performance in an ideal scenario, the overall performance of the network may deteriorate in cases when some of the agents may be influenced by an external adversary [11]. GCN 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 corrupted information with their neighbors, they affect them in turn, and cascading propagation of this harmful information may potentially deteriorate the performance of the entire network. This makes it critical for the healthy agents to have a mechanism to maintain performance in the presence of compromised agents whose identity maybe unknown to them. Thus, the design of resilient algorithms has become a vital area of research in distributed learning. The main goal is to enable the normal agents to be able to mitigate the effect of corrupted parameters shared by the adversarial agents. One of the most effective ways of achieving this goal is by using resilient aggregation, where the normal agents update their parameters based on the shared information [12], [13], while filtering out the effect of adversarial information as much as possible.

Resilient aggregation has been widely applied to design attack-resilient algorithms in reinforcement learning (RL) and federated learning (FL). In RL, performance enhancement has been achieved by average-based aggregation [14], or consensus-based method [12]. A coordinate-wise trimmed

mean was used in [13] to achieve resilience. In FL, various methods have been developed which are shown to be effective against Byzantine attacks, for examples coordinate-wise median [15], coordinate-wise trimmed mean [16], Krum and multi-Krum [17], Bulyan and multi-Bulyan [18].

In this work, we address the problem of attack-resilient aggregation for design of flocking control using GCNs. First, we show that the GCN-based flocking control using average aggregation as proposed in [10] is vulnerable to attacks, and the controller fails to achieve a flock formation. Instead, we propose a resilient aggregation method using coordinate-wise median (CM) to incorporate the information received from the neighbors. The contributions of this paper are the following:

- We develop an adversarial model that arbitrarily corrupts the parameters shared from the designated attack agents. We show that average aggregation in GCN-based controller is ineffective under the presence of such an adversary.
- A resilient aggregation method using coordinate-wise median is proposed for incorporating information from other agents in designing this flocking controller. We perform breakdown point analysis to verify that the aggregated states for each normal agent remain bounded even when the attack signals are arbitrarily large, provided that in the neighborhood of each agent, the normal nodes outnumber the attacked nodes. Note that the condition is imposed only on the immediate neighborhood even though the communication occurs between multi-hop neighbors.
- We perform experiments to verify the robustness of our proposed method and the results verify the claims regarding the performance of the modified GCN controller we propose in this paper.

The notation used in the paper is fairly standard. The cardinality of a finite set A is denoted by $|A|$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$, and $\lfloor . \rfloor$ is the flooring function. Before presenting the results of this work, some preliminary concepts and ideas are discussed in the next section.

II. Preliminaries and Problem Statement

We consider a team of N mobile agents spatially distributed with flocking as their dynamic task. A flocking controller allows these agents to move together while avoiding collision with each other, where the velocity variance between agents is as minimum as possible. The agents are point mass systems having double integrator dynamics, given by

$$\begin{aligned} \dot{r}_i &= v_i, \\ \dot{v}_i &= u_i, \end{aligned} \tag{1}$$

for $i = 1, 2, \cdots, N$. Here r_i and v_i denote the position and velocity, respectively, of agent i. The control input u_i is the acceleration to the system. The system is discretized by the sampling time T_s and the modified signals are denoted with a subscript t. The flocking controller consists of two components - one responsible for alignment of velocity between agents, and the other avoids possible inter-agent collisions. The relative position of agent j with respect to agent i is denoted as $r_{ij,t} = r_{j,t} - r_{i,t}$. The approach of artificial potential function (APF) is used for collision avoidance [19], and the function is chosen as

$$U_t(r_i, r_j) = \begin{cases} \frac{1}{\|r_{ij,t}\|^2} + \log \|r_{ij,t}\|^2, & \|r_{ij,t}\| < \rho \\ \frac{1}{\|\rho\|^2} + \log \|\rho\|^2, & \text{otherwise} \end{cases} \tag{2}$$

where ρ is the communication radius of the mobile agents, dependent on the sensors used. The potential U_t diverges at $\|r_{ij,t}\| = 0$, has a minimum when the distance between agents is $\|r_{ij,t}\| = 1$, and is indifferent when $\|r_{ij,t}\| > \rho$. Thus, it makes the agents maintain a minimum distance of $\|r_{ij,t}\| = 1$ from each other, for all $i, j \in [N], i \neq j$. The global centralized control action for agent i is designed as

$$u^*_{i,t} = -\sum_{j=1}^{N}(v_{i,t} - v_{j,t}) - \sum_{j=1}^{N} \nabla_{r_i} U_t(r_i, r_j), \tag{3}$$

where $\nabla_{r_i} U_t(r_i, r_j)$ is the gradient of the collision avoidance function $U_t(r_i, r_j)$ with respect to $r_{i,t}$. The global controller is characterized by the policy π^*.

As discussed earlier, the need for distributed control becomes inevitable for large group of agents, and the simpler distributed controller is the one that relies on information received only from the neighbors. The interaction between the agents can be represented as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the agents are denoted by the node set $\mathcal{V}$, and the communication between them is captured by the edge set $\mathcal{E}$, i.e., $(i, j) \in \mathcal{E}$, if the agents i and j share their state information with each other. The set of nodes that agent i directly shares information with are called its neighbors, and is denoted by the set $\mathcal{N}_{i,t} = \{j : (i, j) \in \mathcal{E}\}$, where the subscript t denotes the time instant. In the case of flocking, the graph $\mathcal{G}$ is dynamic, due to the mobility of the agents, which have a limited communication range ρ. The local controller can be designed in similar form as (3), while the terms are summed up over the neighboring agents. This controller takes longer for all the agents to coordinate their velocities, and often cannot achieve the required formation.

To allow information exchange between agents that are distantly placed in the graph, aggregation-based GCNs has been employed for designing the flocking controller [9], [10]. Traditionally, GCNs incorporate node features that carry important information about the system and the assigned task. Here the features represent the state vector. For agent i, the state vector at discrete time index t is given by

$$[x_t]_i = \left[\sum_{j \in \mathcal{N}_{i,t}} (v_{i,t} - v_{j,t}), \sum_{j \in \mathcal{N}_{i,t}} \frac{r_{ij,t}}{\|r_{ij,t}\|^4}, \sum_{j \in \mathcal{N}_{i,t}} \frac{r_{ij,t}}{\|r_{ij,t}\|^2} \right] \tag{4}$$

Clearly, this feature vector contains information about the average velocity and also nonlinear functions of inter-agent distance, averaged over the neighborhood. The state information broadcast by agent i at time instant t is denoted by $[\chi_t]_i$.

The GCN-based controller achieves improved performance over local controller by incorporating information from distant neighbors, formally known as the *multi-hop neighbors*. The multi-hop neighborhood is defined recursively, and the idea is

explained as follows. The 0-hop neighborhood is defined as the node itself, $\mathcal{N}_{i,t}^0 = \{i\}$ for all time instants t. The 1-hop neighborhood of i is defined as the set of neighbors of i, i.e., $\mathcal{N}_{i,t}^1 = \mathcal{N}_{i,t}$. The k-hop neighbors of i at time instant t are the nodes that are $(k-1)$-hop neighbors at time $t-1$ of the neighbors of i. The k-hop neighborhood of agent i is thus defined as

$$\mathcal{N}_{i,t}^k = \{j^{'} \in \mathcal{N}_{j(t-1)}^{k-1} : j \in \mathcal{N}_{i,t}\}. \tag{5}$$

The idea of multi-hop neighborhood recognizes the fact that information can propagate through a graph-structured network.

Aggregation GCNs operate on networked graphs to process useful information by repeatedly communicating with the neighboring agents [9]. However, aggregation GCNs typically operate on fixed graphs. In this application, the communication graph is dynamic, as the information exchange between agents depend on their spatial locations. A modified aggregation GCN, called *delay aggregation GCN*, was proposed in [10] that operates on a time-varying graph structure. This architecture allows information flow from multi-hop neighbors, while also recognizing the fact that the information exchange cannot happen instantaneously. A delay factor is introduced while propagating this information, where the units of delay in the received information equals the number of hops the sender is away from the receiver node. Considering this, the information available at node i at time instant t includes the collective states from all its multi-hop neighbors with appropriate delays, given as

$$\Psi_{i,t} = \bigcup_{k=0}^{K-1} \left\{ \left[x_{(t-k)}\right]_j : j \in \mathcal{N}_{i,t}^k \right\}, \tag{6}$$

where K is the depth of information propagation. A larger value of K would improve the performance of the GCN-controller, but at an increased cost of communication and computation.

The distributed control design attempts to find a policy π that maps the information set $\Psi_{i,t}$ to local control action $u_{i,t}$, and we assume that π can be parameterized [9] using tunable parameters H. The goal of the controller design is to minimize a loss function $\mathcal{L}(\pi, \pi^*)$ that measures how close the system implementing the GCN controller to the global centralized controller is. In this controller design, each agent i averages the information received from all its k-hop neighbors, where $k \in [K-1]$ and stacks them in the cumulative aggregated state vector, which serves as the input to the convolution neural networks (CNNs).

As it has been shown in the literature, in average-based aggregation, instability can be caused even by a single malicious node. In this paper, we design an adversary model and show that the average aggregation GCN fails in the presence of such attacks. Moreover, as an attempt to design a resilient GCN-based flocking controller, we focus on a more robust aggregation function, the coordinate-wise median. In particular, this work aims to address the following:

- Design an attack model that demonstrates the ineffectiveness of average aggregation based GCN controller;
- Design a resilient aggregation method for GCN controller using coordinate-wise median;
- Breakdown point analysis of the proposed controller;
- Demonstration of the performance of the proposed controller in comparison with the baseline methods.

The adversary modeling is discussed in the next section.

III. Adversary Model

In the proposed attack model, some of the agents of the group are controlled by a single adversary. This is a *targeted attack* model, and the set of affected agents is denoted by τ. The attack is not direct; instead, it is realized by intercepting the information exchanged between agents. The attack is assumed to take place during the evaluation period, after the training of the GCN is completed. The adversary modifies the states shared by the attacked agents to some arbitrary values. The following assumptions are made regarding the attack model.

Assumption 1. *The set of adversarial agents τ does not change throughout the operation of the algorithm. However, the information shared by these agents to their respective neighbors may change over time.*

Assumption 2. *The adversary compromises the communication network, but the sensors and actuators associated with the agents are correctly functioning at all times.*

It is important to note that the adversary does not need any knowledge about the algorithm employed by the agents, the hyperparameters used, or the communication graph. The only information needed by the adversary is the size of the state vector of the agents. The attacked agents share the adversarial data with its neighbors, which are in turn shared with their distant neighbors. This way, the attacker can achieve the adversarial objective by attacking only a small group of agents. To formally express the adversary model, the state vector broadcast by an agent i at time instant t is given by

$$[\chi_t]_i = \begin{cases} *, & \text{if } i \in \tau \\ [x_t]_i, & \text{otherwise} \end{cases} \tag{7}$$

where $*$ is used to express an array of arbitrary values. It is shown later in the paper that the GCN controller using average aggregation performs poorly in the presence of this attack. This motivates us to design a resilient aggregation method, which is presented in the next section.

IV. Resilient GCN-based Flocking Control

In this section, the proposed resilient algorithm for GCN based flocking control is presented. Coordinate-wise median (CM) has the property to be resilient against adversarial attacks that motivated us to use this function in the state aggregation.

As explained earlier, the local state vector x_t captures the important features required for designing the controller. This information solely depends on the sensor measurements. To design the GCN based controller, the agents need to aggregate information from their immediate and multi-hop neighbors. At node i, the state information available from its 1-hop neighbors are given as $[\chi_{t-1}]_j$ for all $j \in \mathcal{N}_{i,t}^1$. Agent i aggregates this information as

$$[y_{1,t}]_i = CM\left\{ [\chi_{t-1}]_j \right\}, \quad j \in \mathcal{N}_{i,t}^1, \tag{8}$$

where CM denotes the point-wise median function. Thus, each element of this aggregated vector is the median of the respective entries of the sequence $\left\{ [\chi_{t-1}]_j \right\}$, the states of 1-hop neighboring nodes $j \in \mathcal{N}_{i,t}^1$ observed at time $t-1$. Similarly, the information received from multi-hop neighbors can be aggregated as below

$$\begin{aligned} [y_{2,t}]_i &= CM\left\{ [\chi_{t-2}]_j \right\}, \quad j \in \mathcal{N}_{i,t}^2, \\ &\vdots \\ [y_{K,t}]_i &= CM\left\{ [\chi_{t-K}]_j \right\}, \quad j \in \mathcal{N}_{i,t}^K. \end{aligned} \tag{9}$$

These aggregated states are then stacked as the *cumulative aggregated state vector*. At agent i, this sequence at time instant t is given by

$$[z_t]_i = \left[[y_{0,t}]_i ; [y_{1,t}]_i ; \cdots ; [y_{K-1,t}]_i , \right] \tag{10}$$

where $[y_{0,t}]_i = [x_t]_i$. The cumulative aggregation state $[z_t]_i$ is constructed for each node i, which is then supplied to a regular convolutional neural network (CNN) of depth L. For each layer $l = 1, 2, \cdots, L$ of the CNN, we have

$$[z_t]_i^{(l)} = \sigma^{(l)}\Big(H^{(l)} [z_t]_i^{(l-1)} \Big), \tag{11}$$

where $[z_t]_i^{(l)}$ is the output of the lth layer, $\sigma^{(l)}$ denotes point-wise nonlinearity, and $H^{(l)}$ are the graph filters containing the learnable parameters. The initialization is performed with the cumulative aggregated state vector, i.e., $[z_t]_i^{(0)} = [z_t]_i$. The output of the last layer should calculate the suitable control action of the agent, i.e., $u_{i,t} = [z_t]_i^{(L)}$. The parameters $H^{(l)}$ are shared across all agents. This implies that once trained, the operation of the controller depends only on the aggregated information. Clearly, the overall architecture can be considered as a modified Graph Convolution Network (GCN) where the aggregation takes place using coordinate-wise median function.

As mentioned earlier, the designed controller aims to imitate the global centralized controller. So, the training set $\mathcal{T}$ consists of sample trajectories $(x_t, \pi^*(x_t))$ obtained from the centralized controller $u_t^* = \pi^*(x_t)$, where u_t^* is the collective control action of all the agents and x_t is the collective feature vector of all the agents. The goal of the learning is to find the optimal parameters H^*, given by

$$H^* = arg\min_{H} \sum_{\left(x_t, \pi^*(x_t)\right) \in \mathcal{T}} \mathcal{L}\Big(u_t, u_t^*\Big),$$

where $\mathcal{L}\big(u_t, u_t^*\big)$ is the quadratic loss function involving the GCN based and the global optimal control actions. The proposed robust aggregation method in design of GCN based controller design achieves resilient performance in the presence of attacks. In addition to demonstrating the effectiveness through simulation results, it is also important to analytically investigate the robustness of the method. The breakdown point analysis of the proposed method is presented in the next section.

V. Robustness Analysis

According to the proposed attack model, the adversary targets a few agents, which broadcast malicious information to their neighbors. As the information propagates through the graph, the performance of many other agents may get affected, due to the information exchange between agents. 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 i aims to find out the minimum fraction ϵ for which the parameters estimated by agent i remains bounded in the presence of ϵN attacked agents. In this work, the connectivity between agents is based on the distance between them, so it is not practical to consider a fully connected graph for this application. The breakdown point analysis is thus performed based on the number of neighbors. Formal definition of the breakdown point is given below.

Definition 1. *[20] The breakdown point of an estimator $\boldsymbol{T}$ of a collection $\mathbf{X}$ of n observations is defined as the smallest fraction m/n of outliers that can produce an unbounded estimate*

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

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.

The breakdown point analysis discussed here corresponds to the aggregation layer of the network, in particular, the input to the convolution network. It is easy to verify that robustness in this aggregation layer determines the robustness of the overall controller.

Let the neighborhood of agent i at time instant t, denoted as $\mathcal{N}_{i,t}^1$, be divided into two disjoint sets, ${}_h\mathcal{N}_{i,t}^1$ and ${}_a\mathcal{N}_{i,t}^1$, where ${}_h\mathcal{N}_{i,t}^1$ denotes the set of healthy neighbors of i, and ${}_a\mathcal{N}_{i,t}^1$ is the set of its attacked neighbors. Clearly, ${}_h\mathcal{N}_{i,t}^1 \bigcup {}_a\mathcal{N}_{i,t}^1 = \mathcal{N}_{i,t}^1$ and ${}_h\mathcal{N}_{i,t}^1 \bigcap {}_a\mathcal{N}_{i,t}^1 = \emptyset$. In a similar fashion, the multi-hop neighborhoods of each agent can be divided into disjoint sets, with healthy and adversarial agents. The analysis is performed considering the worst-case scenario, where we aim to find the minimum fraction of attacked neighbors of agent i such that $\|[z_t]_i\| < \infty$ does not hold anymore, given that $\left\| [\chi_{t-k}]_j \right\| \to \infty$ for all $j \in {}_a\mathcal{N}_i^{all}$ and $k \in \{1, 2, \cdots, K-1\}$. Here ${}_a\mathcal{N}_i^{all}$ is the set of all adversarial neighbors of agent i, i.e., ${}_a\mathcal{N}_i^{all} = {}_a\mathcal{N}_{i,t}^1 \bigcup {}_a\mathcal{N}_{i,t}^2 \bigcup \cdots \bigcup {}_a\mathcal{N}_{i,t}^{K-1}$. The following theorem shows that the cumulative aggregation state of agent i remains bounded even when the states shared by the attacked agents have an infinite norm, provided there is a majority of healthy agents in the 1-hop neighborhood of each agent of the group.

Theorem 1. *Consider a graph of N cooperative agents learning distributed flocking controller using graph convolution networks. The input to the convolution network for each agent i is given by the equation* (10)*, where each of its elements are calculated according to the equations* (4)*,* (8)*, and* (9)*. Some of the agents in the network are under attack, and the*

adversary model is given by (7). *Then for each healthy agent* i*, the cumulative aggregated state has a finite breakdown point of* $\frac{1}{|\mathcal{N}_{i,t}^1|}\left\lfloor\left(|\mathcal{N}_{i,t}^1|+1\right)/2\right\rfloor$.

Proof. The cumulative aggregated state of agent i remains bounded if and only if the aggregated neighbor information at the agent i are bounded for up to $(K-1)$-hop neighbors, i.e.,

$$\left\|[y_{0,t}]_i\right\| < \infty, \quad \left\|[y_{1,t}]_i\right\| < \infty, \cdots, \quad \left\|[y_{K-1,t}]_i\right\| < \infty$$

We need to analyze each of these conditions individually. The first term, $[y_{0,t}]_i$ captures the information from the sensor measurements of the agents. As the adversary is assumed to launch the attacks only through the communicated states, this term remains bounded always. The second term, $[y_{1,t}]_i$ is calculated by taking median of $\left\{[\chi_{t-1}]_j\right\}$, $j \in \mathcal{N}_{i,t}^{(1)}$. From the properties of median, $[y_{1,t}]_i$ is bounded as long as there is a majority of normal agents in $\mathcal{N}_{i,t}^{(1)}$, i.e., $\frac{|_a\mathcal{N}_{i,t}^1|}{\mathcal{N}_{i,t}^1} < 0.5$. Following the same analysis, we can argue that $[y_{2,t}]_i$ is bounded when $\frac{|_a\mathcal{N}_{i,t}^2|}{\mathcal{N}_{i,t}^2} < 0.5$. Therefore, the conditions for the aggregation scheme to be robust are given as

$$\frac{|_a\mathcal{N}_{i,t}^p|}{\mathcal{N}_{i,t}^p} < 0.5 \tag{12}$$

for $p \in \{1, 2, \cdots, K-1\}$ and all $i \in [N]$. Thus, the cumulative aggregation state for agent i, given as $[z_t]_i$, remains bounded under the condition that there is a minority of attacked agents in up to $(K-1)$-hop neighborhoods of i.

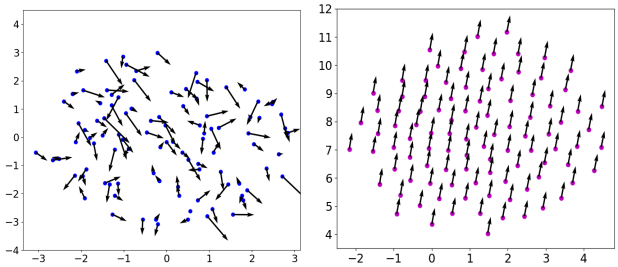
To simplify this condition, we use the definition of recursive neighborhood. It was shown that $[y_{1,t}]_i$ is bounded if there is a minority of attacked agents in the 1-hop neighborhood of agent i. From the point of view of the whole network, for the aggregated information from the 1-hop neighbors to be bounded, there should be a minority of attacked agents in the 1-hop neighborhood of all the agents at all times.

Now, by definition, the 2-hop neighbors of agent i at time instant t are the collective 1-hop neighbors of agent $j \in \mathcal{N}_{i,t}^1$ at time $(t-1)$. By the condition derived earlier, and Assumption 1, each of $j \in \mathcal{N}_{i,t}^1$ has a minority of attacked 1-hop neighbors at time instant $(t-1)$. Thus, the 2-hop neighbors of agent i has a minority of attacked agents. The same argument can be applied recursively up to $(K-1)$-hop neighbors of all agents. It is therefore shown that the cumulative aggregation state remains bounded when the number of normal 1-hop neighbors outnumbers the number of 1-hop adversarial neighbors, for all the nodes in the graph. □

This concludes the breakdown point analysis of the proposed method, which is consistent with the breakdown point of the coordinate-wise median function.

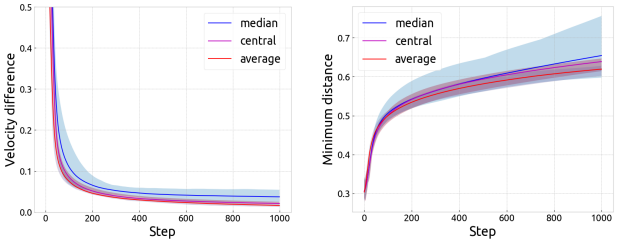
VI. Evaluation

In this section, we present evaluation results of the proposed robust aggregation based flocking controller applied to a swarm of 100 mobile agents with continuous-time second order dynamics (1), having a communication range of $\rho = 2$m. The discretization time period is $T_s = 0.01$s. The initial positions of the agents are chosen randomly from a ball of radius 10m, and their initial velocities are sampled uniformly from the interval $\left[-v_{init}, +v_{init}\right]$, where $v_{init} = 3$m/s.



(a) t = 0 (b) t = 300

Fig. 1: Position and velocity of the agents at two different time instants using the proposed controller under normal scenario.

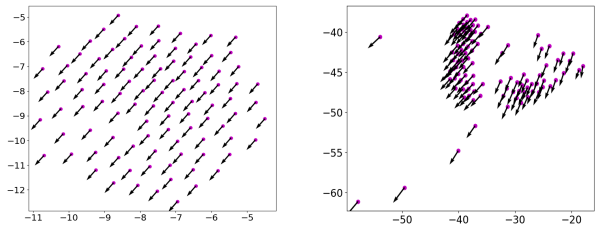


(a) Average magnitude of velocity difference (b) Average minimum distance between neighboring agents

Fig. 2: Properties of the flocks using different algorithms under normal scenario.

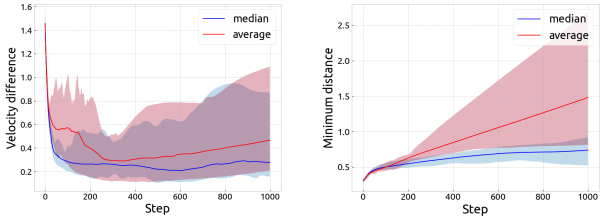
Each agent uses an aggregation GCN, where the input features are given by (4), and the cumulative aggregated state vector is computer using (10) with $K = 3$. This goes as the input to a fully connected neural network with two hidden layers, each having 64 neurons and $Tanh$ activation function. To train the GCN filters, 240 trajectories are generated, each of length 200 steps, from the global centralized controller. The Adam optimizer is used with learning rate 1×10^{-5}. The learned controller was then tested using 20 trajectories, each having 200 steps. To evaluate the performance of our proposed controller, the global centralized controller (3) and the average aggregation based GCN controller are used as baselines.

First, to demonstrate the performance of the proposed controller without adversaries, the initial position and velocity of the agents are plotted in Figure 1(a), and the same are plotted after 300 time steps in Figure 1(b). It can be verified that with the proposed method, the agents are able to achieve a flock formation after a few time steps. To compare the performance of this controller with the baseline methods, the average magnitude of velocity difference for all the agents are plotted in Figure 2(a). The average minimum distance from the neighbors for all the agents is plotted in Figure 2(b). These plots depict the mean values over all the testing episodes, and the shaded areas represent the standard deviation. It can be verified that the performance of the median based method is close to the baseline methods. The evaluation results demonstrate the controller performance under a normal scenario without attacks. To verify the claim about the resilient properties of the proposed controller, we



(a) Median aggregation (b) Average aggregation

Fig. 3: Position and velocity of the agents at $t = 300$ using different aggregation methods under adversarial attack.



(a) Average magnitude of velocity difference (b) Average minimum distance between neighboring agents

Fig. 4: Properties of the flocks using different algorithms under adversarial scenario.

simulate the controllers under the proposed attack model (7). We assume that the adversary has control over 5 agents of the group. Each element of the communicated states from these attacked agents is randomly chosen from the interval $[-0.3, 0.3]$. As shown in Figure 3(a), the agents are able to achieve the flock formation even in the presence of the attack, when the proposed robust aggregation based controller is in use. In contrary, when the aggregation is performed using averaging method, the agents get dispersed, and the formation is not achieved, as shown in Figure 3(b). Similar to the normal scenario, the average velocity mismatch of the agents and the average minimum distance from the neighbors are plotted in Figure 4(a) and Figure 4(b), respectively. The average aggregation method under the attack is not cohesive enough, as the average minimum distance between neighbors increases, and thus the flock loses its formation. Also, the median-based method achieves a lower velocity mismatch than the average method. These plots altogether support our claim that the proposed median based GCN controller has resilient property against the proposed attack model.

VII. Conclusion

This work proposes a resilient aggregation method for designing a GCN-based flocking controller. The information from multi-hop neighbors are aggregated using coordinate-based median. We proposed an attack model that exploits the communication between agents, and can compromise the formation when the average aggregation is used, whereas the proposed method is shown to be robust, provided that the required condition is satisfied as found from the breakdown point analysis. The simulation results show that the proposed controller's performance is comparable to the average-based method in the normal scenario without attacks. Under attack, the method is resilient in contrast to the average-based method that may become unstable in the presence of a single adversarial node.

In the future, we also aim to evaluate the proposed resilient method against other attack models, including those relevant to the specific flocking problem as well exploring other resilient aggregation methods.

References

[1] J. Ferber and G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*, vol. 1. Addison-Wesley Reading, 1999.

[2] R. M. Murray, "Recent research in cooperative control of multivehicle systems," 2007.

[3] A. E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, no. 2, pp. 97–120, 2008.

[4] H. S. Witsenhausen, "A counterexample in stochastic optimum control," *SIAM Journal on Control*, vol. 6, no. 1, pp. 131–147, 1968.

[5] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[6] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," tech. rep., 1989.

[7] A. Giusti, J. Guzzi, D. C. Cireşan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro, *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2015.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[9] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2018.

[10] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Conference on robot learning*, pp. 671–682, PMLR, 2020.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation," in *Uncertainty in Artificial Intelligence*, pp. 261–270, PMLR, 2020.

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

[17] 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.

[18] 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.

[19] S. S. Ge and C.-H. Fua, "Queues and artificial potential trenches for multirobot formations," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 646–656, 2005.

[20] 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.