Attacking Fake News Detectors via Manipulating News Social Engagement

Haoran Wang
Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
hwang219@hawk.iit.edu

Yingtong Dou
Department of Computer Science, University of Illinois Chicago, Chicago, IL, USA
Visa Research, Palo Alto, CA, USA
ydou5@uic.edu

Canyu Chen
Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
cchen151@hawk.iit.edu

Lichao Sun
Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA
james.lichao.sun@gmail.com

Philip S. Yu
Department of Computer Science, University of Illinois Chicago, Chicago, IL, USA
psyu@uic.edu

Kai Shu
Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
kshu@iit.edu

ABSTRACT

Social media is one of the main sources for news consumption, especially among the younger generation. With the increasing popularity of news consumption on various social media platforms, there has been a surge of misinformation which includes false information or unfounded claims. As various text- and social context-based fake news detectors are proposed to detect misinformation on social media, recent works start to focus on the vulnerabilities of fake news detectors. In this paper, we present the first adversarial attack framework against Graph Neural Network (GNN)-based fake news detectors to probe their robustness. Specifically, we leverage a multi-agent reinforcement learning (MARL) framework to simulate the adversarial behavior of fraudsters on social media. Research has shown that in real-world settings, fraudsters coordinate with each other to share different news in order to evade the detection of fake news detectors. Therefore, we modeled our MARL framework as a Markov Game with bot, cyborg, and crowd worker agents, which have their own distinctive cost, budget, and influence. We then use deep Q-learning to search for the optimal policy that maximizes the rewards. Extensive experimental results on two real-world fake news propagation datasets demonstrate that our proposed framework can effectively sabotage the GNN-based fake news detector performance. We hope this paper can provide insights for future research on fake news detection.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; • **Information systems** → **Social networks**.

WWW '23, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3583868

KEYWORDS

Social Network; Fake News Detection; Adversarial Robustness

ACM Reference Format:
Haoran Wang, Yingtong Dou, Canyu Chen, Lichao Sun, Philip S. Yu, and Kai Shu. 2023. Attacking Fake News Detectors via Manipulating News Social Engagement. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3543507.3583868

1 INTRODUCTION

With the burgeoning of social media, inaccurate or unfounded information (i.e., *misinformation*) is also circulating on social media, which demotes people's belief in truth and science [6, 40]. Unlike traditional news media, social engagement like commenting and sharing expedite the spread of misinformation and exaggerate its influence at scale. Recent research has pointed out that misinformation has been hindering the promotion of vaccines and threatening public health during the COVID-19 global pandemic [26].

To combat massive misinformation on social media, many machine learning based misinformation detectors are proposed [48]. Besides the methods utilizing natural language processing techniques to check the news content and its writing style to verify its veracity [21, 36, 46], recent works have begun to leverage news social engagement using graph models for fact-checking [3, 27, 30, 35]. Compared to the straightforward NLP-based methods, social-engagement-based methods regard engaged users as an integral part of news posts. Based on the theory and evidence that news consumers have preferences on news content (i.e., the *echo chamber*) [10, 16, 27, 29, 39], the engagement patterns of misinformation and fact are also different. Moreover, the prevalent bots and fraudsters engaged with fake news posts also differentiate their engagement patterns from regular ones [37].

Despite the rapid development of automatic fact-checking, most fake news detectors are static models vulnerable to adversarial attacks. Similar to many security problems, we must acknowledge that misinformation detection is an armed race between content moderators and malicious actors aiming at manipulating public opinion or gaining money through incited social engagement.

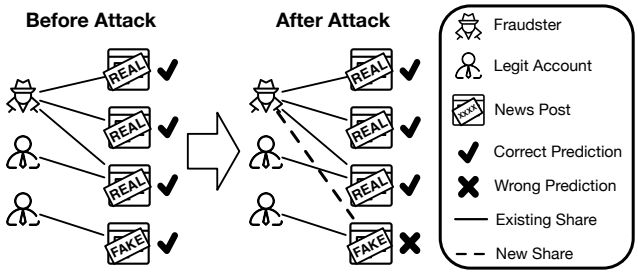

Figure 1: An illustration of attacking a fake news classifier via manipulating news posts' social engagement. The classifier misclassifies the fake news after the fraudster who has shared many real news posts shares it.

Therefore, it is imperative to enhance the robustness of misinformation detectors. Though some recent works have investigated the robustness of NLP-based misinformation detectors [1, 18, 19, 24, 49], no work has probed the robustness of social-engagement-based misinformation detectors. [24] and [28] are two closest works to ours. However, they either do not consider social engagement-based detectors or do not model the diverse fraudster type in the misinformation campaign.

We use Figure 1 to demonstrate the vulnerability of social engagement based misinformation detectors. Many existing works [30, 35] model news social engagement on social media as a heterogeneous graph where users and news posts are nodes, and an edge means a user has shared the post. Graph Neural Networks (GNNs) [15, 22, 45] have been widely leveraged to encode the above social engagement graph and predict the veracity of news posts. Many GNNs are designed to encode the neighboring node information to enhance the prediction performance of the target node. To exploit this property, as shown in Figure 1, the fraudster who has shared many real news can flip the GNN-based misinformation detector's prediction on a target fake news by sharing it. Because the newly added real news neighbors will alleviate the suspiciousness of the target fake news.

To analyze the robustness of social-engagement-based misinformation detectors, inspired by GNN robustness research [42], we propose to attack GNN-based misinformation detectors by simulating the adversarial behaviors of fraudsters. However, the real-world misinformation campaign delivers three non-trivial challenges for attack simulation: **(1)** To evade detection while promoting fake news on social media, malicious actors can only manipulate the controlled user accounts to share different social posts. However, most of the previous GNN adversarial attack works assume all nodes and edges can be perturbed, which is impractical. **(2)** Many deployed GNN-based fake news detectors are grey-box models with various model architectures tailored to the heterogeneous user-post graph. Thus, the gradient-based optimization method used by previous works [50] cannot be utilized to devise an attack. **(3)** Real-world evidence [31, 44] shows that various coordinated malicious actors have engaged in the misinformation campaign. Different types of malicious actors have different capabilities, budgets, and risk appetites. For instance, key opinion leaders have stronger influence than social bots but cost more to cultivate.

To overcome the above challenges, we devise a dedicated Multi-agent Reinforcement Learning (MARL) framework, while none of the previous GNN robustness work was used. Specifically, to simulate the real-world behavior of fraudsters who share different posts, we harness a deep reinforcement learning framework to flip the classification result of a target news node by modifying the connections of users who shared the post. We model the MARL framework as a Markov Game where the agents work coordinately to flip the classification result. Overall, our contributions are:

- To the best of our knowledge, we are the first work to probe the robustness of GNN-based fake news detectors from a social engagement perspective. Although there have been previous works on attacking fake news detectors using NLP methods, attacking fake news detectors by manipulating the social engagement of news targets has not been studied.
- We leverage a MARL framework to perform targeted attacks on GNN-based fake news detectors to simulate real-world misinformation campaigns. Specifically, we modeled fraudsters as agents with different costs, budgets, and influences in our framework.
- Our experiment results show that our proposed MARL framework could effectively flip the GNN prediction results. We discuss the vulnerabilities of GNN-based fake news detectors and provide insights on attack strategies and countermeasures.

The rest of the paper is organized as follows. In Section 2, we introduce related work. In Section 3 and 4, we introduce the problem definition and proposed framework. In Section 5, we report our experiment results and analysis. Finally, we discuss the limitation and future work of this paper in Section 6.

2 RELATED WORK.

In this section, we review the related work on (1) graph neural network-based fake news detection; (2) adversarial attack on graph neural networks; and (3) adversarial attack on fake news detection.

2.1 GNN-based Fake News Detection

We can categorize the existing GNN-based misinformation detection works into two major categories according to their graph prototypes: **1)** Propagation-based work [17, 27, 29, 41]: these works model the sharing sequence of a news post as a tree-structured propagation graph with the news post as the root node and edges representing shared relations between users. It can be formulated as either a propagation graph classification or a root node classification task. The propagation graph is infeasible for adversarial attacks because the attacker needs to employ a lot of users to share the target post to flip its classification results. At the same time, such operations are naive for optimization and easily captured by simple outlier measurements. **2)** Social-context-based work [5, 30, 35, 47]: all users and their shared news posts (e.g., tweets) formulate a bipartite graph (as shown in Figure 1) where an edge means a user shared the post and the objective is training a GNN to classify the news post nodes. Note that previous works usually add the publisher as the third type of node connecting to social posts. In this paper, we only consider the common-used graph prototype (i.e., user-post bipartite graph) as it is easier to manipulate in practice.

2.2 Adversarial Attack on GNNs

As GNNs attain excellent performance on many graph mining tasks, their robustness against adversarial attacks has drawn increased attention in recent years [42]. RL-S2V [8] and Nettack [50] are two early GNN attacking algorithms aiming at lowering the GNNs' node classification performance via add/deleting edges or modifying node features under a given budget. Following these work, other works have begun to investigate the GNN robustness under

different tasks, e.g., link prediction [4], knowledge graph embedding [34], and community detection [25]. However, none of the previous works have attempted to attack GNN-based fake news detectors, which have recently become popular amid massive adversaries engaging in fake news spread [37]. Compared to the previous works using reinforcement learning to attack GNNs, our work utilizes a multi-agent setting to mimic the real-world misinformation campaign. In addition, to simulate the real-world attack setting, we only manipulate the edges of the news social engagement graph since it is unlikely that attackers can modify news posts.

2.3 Adversarial Attack on Fake News Detectors

Given a wide array of machine learning-based fake news detectors, only a few works have investigated the robustness or vulnerabilities of fake news detectors [1, 9, 18, 19, 23, 24, 49]. Among those works, [19] examines the robustness of text-based news veracity classifiers over time and against attacks crafted by manipulating news sources. [1, 23, 49] probe the robustness of NLP-based fake news detectors by devising various attacks that distort the news content or inject adversarial texts. Nash-Detect [9] and AdRumor-RL [28] study the robustness of graph-based spam detectors and rumor detectors respectively, using the reinforcement learning framework. MALCOM [24] carries out the attacks from another perspective which modifies the comments of each piece of news to fool the fake news detectors leveraging multi-source data. PETGEN [18] simulates the behavior of malicious users on social media by generating a sequence of texts to attack sequence-based misinformation detectors. Unlike previous work, we are the first to explore the robustness of social context-based fake news detectors using a multi-agent reinforcement learning framework.

3 PROBLEM FORMULATION

We formulate the problem of attacking social-engagement-based fake news detectors as attacking GNNs on a user-post sharing graph. In this section, we first define GNN-based fake news detection and then introduce our adversarial attack objective.

3.1 GNN-based Fake News Detection

A user-post sharing graph is defined as a bipartite graph $\mathcal{G} = \{U, V, E, \mathbf{X}_u, \mathbf{X}_v, Y\}$, where $U = (u_0, \cdots, u_i)$ is a set of users, $V = (v_0, \cdots, v_j)$ is a set of news posts, and the edge $e_{ij} = (u_i, v_j) \in E$ indicates the user u_i has shared the news post v_j. $\mathbf{X}_u$ and $\mathbf{X}_v$ are two feature matrices of user nodes and news nodes, respectively. According to previous works [10, 17, 30], the feature vectors of users and news can be composed of their text representations or handcrafted features. Following [10], we use the 300-dimensional Glove embeddings of users' historical posts and news post text to represent $\mathbf{X}_u(i,:)$ and $\mathbf{X}_u(j,:)$, respectively. We use $\mathbf{X}$ to represent all node features for convenience. $y_j \in Y$ represents the label of $v_j \in V$ where 1 (0 resp.) represents fake news (real news resp.).

To detect fake news based on $\mathcal{G}$, a general GNN framework [15, 45] can be applied to it. Concretely, to learn a news post v's representation, a GNN aggregates its neighbors' information recursively:

$$\mathbf{h}_v^{(l)} = \sigma\left(\mathbf{h}_v^{(l-1)} \oplus \mathrm{AGG}^{(l)}\left(\mathbf{h}_u^{(l-1)}, (u,v) \in E\right)\right), \tag{1}$$

where l is the GNN layer number. AGG is the GNN aggregator that aggregates neighbor embeddings. Common aggregators employ attention [45], mean [15], and summation [15]. $\oplus$ represents the operation that combines the embedding of v at the last GNN layer and its aggregated neighbor embeddings. Common approaches include concatenation and summation. Similarly, the representation of u can be learned by the same process shown in Eq. (1).

To classify the news post $v \in V$, a GNN classifier f takes the $\mathcal{G}$ as input where X_u and X_v are node features of u and v. It maps $v \in V$ to $y \in (0,1)$ after feeding the h_v at the last layer to an MLP and softmax layer. The GNN classifier can be trained on partially labeled post nodes with the following cross-entropy loss in a semi-supervised fashion:

$$\mathcal{L}_{GNN}(\mathcal{G}, f_\theta) = \sum_{v_j \in \mathcal{V}} -\log\left(y_j \cdot \sigma((f_\theta(E, \mathbf{X})_j)\right). \tag{2}$$

3.2 Adversarial Attacks on GNN-based Fake News Detectors

At a high level, our problem can be regarded as attacking GNN-based node classification but with practical constraints to simulate real-world misinformation campaigns. Specifically, the objective of the attacking method is to flip the GNN classification results of target social posts via maneuvering controlled malicious social media user accounts to share new posts. Note that we assume attackers can only perturb the graph by controlling malicious users to share news posts and not delete existing shared news posts. We make this assumption because in a real-world setting, even though the users can delete existing shared posts, the record of shared relations may still exist in the database. Considering the massive social network data and the diverse fake news detectors employed by the platform, we assume the unknown target GNN is pre-trained on clean data in our problem setting (i.e., the training data is not poisoned by the adversary). Also, we assume that we have knowledge about the type of GNN the detector is trained on, but we do not have access to its model parameters. Thus, our problem is a grey-box evasive structural attack on the GNN-based node classification task. We formally define our attack objective as:

$$\begin{aligned} &\max_{U_c, E_a} \sum_{v \in V_T} \mathbb{1}(f_{\theta^*}(E', \mathbf{X})_v \neq y_v) \\ &\text{s.t. } \theta^* = \arg\min_\theta \mathcal{L}_{GNN}(\mathcal{G}, f_\theta), \\ &\quad |U_c| \leq \Delta_u, |E_a| \leq \Delta_e, \end{aligned} \tag{3}$$

where U_c, E_a, and V_T represent a set of controlled users, manipulated edges, and target news posts, respectively. $\mathcal{G}$ represents the clean graph and E' is the set of perturbed edges. Δ_u (Δ_e resp.) represents the budget of controlled users (modified edges resp.). The above adversarial objective essentially maximizes the misclassification rate of the target social posts.

4 METHODOLOGY

In practice, misinformation campaigns are carried out by coordinated fraudsters manipulating social user accounts to evade detection. In this section, we first elucidate the property of attackers motivated by real-world misinformation campaigns. Then we present the multi-agent reinforcement learning framework we use to probe the robustness of GNN-based fake news detectors.

4.1 Attacker Property

4.1.1 Attacker Knowledge. As introduced in Section 3.2, our attack is a grey-box attack meaning the attackers only have knowledge

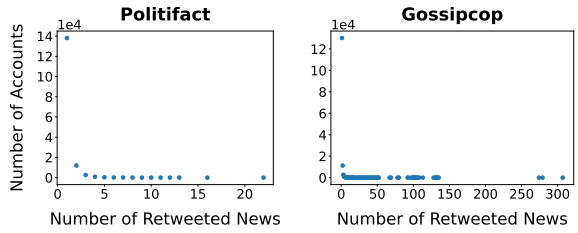

Figure 2: The user account number distribution according to the amount of news shared by them. We sample accounts in different ranges to represent different user types.

about the architecture of the GNN-based classifier, but not its model internals like weights or coefficient values. To attack the GNN classification results of target posts, we assume the attacker can sense the entire graph, including the features and labels of the user and post nodes as well as their connections. The above setting is practical since social media information is publicly accessible and the attackers can easily infer node features and labels given fruitful related works in misinformation research.

4.1.2 Attacker Capability. To imitate the real-world behavior of fraudsters as much as possible, we define the capability of users controlled by fraudsters (i.e., U_c) as follows:

- *Direct Attack*: $u \in U_c$ shares the $v \in V_T$ directly if $(u, v) \notin E$. In real-world settings, given that a controlled user shared many posts from trustworthy sources that seemed to be legitimate, it will help alleviate the suspiciousness of a fake news post if the user shares the post.
- *Indirect Attack*: For $u \in U_c, v \in V_T, (u, v) \in E$, we carry out the attack by controlling u to share $v' \notin V_T$. The indirect attack exploits the neighbor aggregation mechanism of GNNs by exerting influence on the target post by changing its neighborhood. In practice, for a controlled user having shared the target post with fake new, one can let the controlled user share posts from trustworthy sources to mislead the GNN's prediction on the target fake news post. Since indirect attack does not directly modify the edge between controlled user and target news post, it is less noticeable than direct attack.

Note that as previously mentioned in Section 3.2, attackers are only allowed to add edges between user nodes and news post nodes.

4.1.3 Agent Configuration. Real-world evidence shows that multiple groups of malicious actors are engaged in the misinformation campaign [32, 37, 40]. Besides singletons who act individually, most misinformation campaigns are executed coordinately by professional agencies since it would reach the campaign goal faster while maximizing the utilities of existing resources. From the adversarial attack perspective, different types of controlled user accounts have distinct influences on target posts and different budgets. For instance, the bot users are usually low-cost and with a higher budget. However, these bot users have few historical records; thus each bot user has limited influence on target posts [37]. The crowd workers with credible and rich social profiles are usually expensive, but they have a stronger influence on target posts.

To model the above distinct malicious actor groups, previous single-agent RL frameworks are not applicable [8, 43]. Therefore, we leverage a MARL, which not only enables the personalized configuration for each group but also helps simulate the coordinated behavior between different groups. Specifically, we define three agents which control three distinct groups of user accounts according to the malicious accounts introduced in [40]. We divide the user accounts based on the number of news they have shared, Figure 2 shows the distribution of the number of news that users have shared in Politifact and Gossipcop datasets. Table 1 compares the key properties of the following agents.

1) **Agent 1 (Social Bots):** Social bots registered and fully controlled by automated programs have been proven to engage in fake news spreading by many works [2, 37]. The first agent controls the bot users, and it has a low cost and high budget. We randomly select the users with only one connection in our datasets to represent the newly created bot users.

2) **Agent 2 (Cyborg Users):** According to [40], cyborg users are registered by humans and partially controlled by automated programs. The easy switch of functionalities between humans and bots offers cyborgs unique opportunities to spread fake news. Since those users are camouflaged as human, they usually have more historical engagements (i.e., connections to other posts). In our datasets, we randomly select the users with more than 10 connections to represent the compromised users. The cost, budget, and influence of cyborg agents are between that of the other two agents.

3) **Agent 3 (Crowd Workers):** The crowd workers are usually of high cost since they get paid for each campaign. Meanwhile, they have the strongest influence. We take the users with more than 20 connections, where 100% of them connect to real-news posts to represent the crowd workers.

4.2 Attack Framework

In the real world, each agent above represents a malicious actor that aims to influence the fake news classification results. Given a set of target news posts V_T, the attack process can be modeled as a multi-agent cooperative reinforcement learning problem where all agents work together to maximize the misclassification rates of target news posts. Figure 3 illustrates the attack process of the proposed MARL algorithm. First, actions made from different agents are aggregated by the center controller; then, aggregated actions are applied to the environment composed of the social engagement graph and the surrogate classifier; the updated state and rewards generated by the classifier are finally sent back to each agent for the next episode of optimization. In this subsection, We first define each component of the MARL framework, then introduce how we leverage deep Q-learning for optimization.

4.2.1 MARL Framework. Different from previous GNN attacks using single-agent RL, which can be modeled as a Markov Decision

Table 1: The comparison of properties among bots, cyborgs, and crowd worker agents.

Agent	User	Cost	Influence	Budget
1	bot	low	low	high
2	cyborg	moderate	moderate	moderate
3	crowd worker	high	high	low

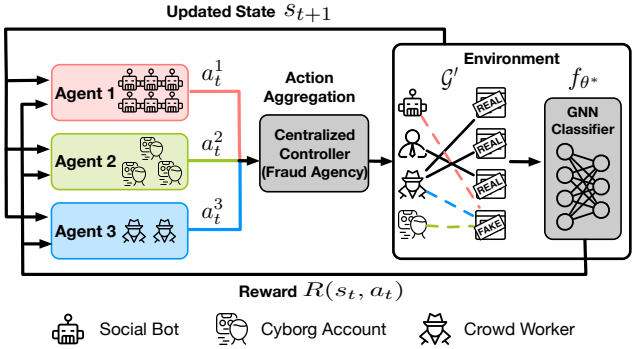

Figure 3: The proposed MARL framework to generate adversarial edge perturbations against GNN-based fake news classifier. See Section 4.2 for more details.

Process (MDP) [8, 43], the MARL framework is a Markov game (MG). We formally define the MG and its components as follows:

Definition 4.1. A Markov game is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{R^i\}_{i\in\mathcal{N}}, \gamma)$, where $\mathcal{N} = \{1, \cdots, N\}$ represent the set of N agents, $\mathcal{S}$ is the state space observed by all agents, $\mathcal{A}^i$ denotes the action space of agent i. $\mathcal{P}$ is the state transition probability given a state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$. R is the reward function that determines the immediate reward received by agent i after a transition from (s, a) to s'. γ is the discount factor to long-term reward.

- **Action.** As defined in Section 4.1.2, each $u \in U_c$ can only add edges based on their connection status to $v \in V_T$. Meanwhile, each agent controls a set of users U_c^i according to Section 4.1.3. We use $a_t^i(u, v)$ to denote the action that adds the edge between user u and post v. Thus, the action space for agent i at time t is $a_t^i \in \mathcal{A}^i \subseteq U_c^i \times V_T$. We use a centralized controller to aggregate agent actions. Specifically, in each episode, the final actions are from the three types of agents with a fixed proportion, this proportion is motivated by real-world misinformation campaigns.
- **State.** Since all agents work cooperatively to attack the same set of target posts V_T against the same classifier f, all agents share the same state at time t represented by $(\mathcal{G}_t', f)$, where $\mathcal{G}_t'$ is the perturbed graph at time t.
- **Reward.** As a grey-box attack, we aim to flip the classification results of the target classifier. Since we have knowledge of the GNN architecture of the detector, we use one of the three GNN models (i.e. GAT, GCN, and GraphSAGE) as our surrogate target classifier and take its classification results on V_T as the reward to guide the agent. Note that the reward is shared by all agents under the cooperative setting. After all agents make the actions under their budgets (i.e., one episode), the reward for each agent towards the target post $v \in V_T$ is:

$$R\left((\mathcal{G}', f_{\theta^*})_v\right) = \begin{cases} 1 : f_{\theta^*}(E', \mathbf{X})_v \neq y_v, \\ -1 : f_{\theta^*}(E', \mathbf{X})_v = y_v. \end{cases} \tag{4}$$

- **Terminal.** After each agent makes finite modifications according to their own budget Δ_e^i, the Markov game stops.

4.2.2 Deep Q-Learning. To solve the above Markov game, we need to find the optimal policy that maximizes the expected value of long-term rewards. Since each agent has its own controlled user accounts and budget, each agent i should have its own policy π^i that $a_t^i \sim \pi^i(\cdot|s_t)$. We use the Q-learning to learn the optimal policy $\pi^{i,*}$ parameterized by a Q-function $Q^{i,*}(s_t, a_t)$. The optimal Q-value for agent i can be represented by the following Bellman equation:

$$Q^{i,*}\left(s_t, a_t^i\right) = R\left(s_t, a_t^i\right) + \gamma \max_{a'} Q^*\left(s_{t+1}, a^{i,\prime}\right), \tag{5}$$

where $a^{i,\prime}$ represents agent i's future action based on state s_t. The above equation suggests a greedy policy where the agent i's best action based on s_t is the action that maximizes the Q-value above:

$$\pi^i\left(a_t^i \mid s_t; Q^{i,*}\right) = \arg\max_{a_t^i} Q^{i,*}\left(s_t, a_t^i\right). \tag{6}$$

For each target post $v \in V_T$, we would like to choose the controlled user $u \in U_c$ with the most influence to v to flip the GNN classification result on v. Thus, using GNNs to parameterize the Q-function could help model each action's value. Specifically, we first employ a two-layer GraphSAGE [15] to obtain the embedding of each post node $h_{v,t}$ in current state s_t according to Eq. (1). Note that we only consider the 2-hop neighborhood of all target nodes and controlled user accounts, which could reduce the state and action space. For agent i at time t with embeddings of controlled user accounts $h_{u,t}, u \in U_c^i$ and the target node $h_{v,t}, v \in V_T$, the Q-value of action $a_t^i(u, v)$ is calculated by the following equation:

$$Q^i\left(s_t, a_t^i(u, v)\right) = \sigma\left(W^1(h_{u,t})\right) \cdot \sigma\left(W^2(h_{v,t})\right), \tag{7}$$

where two liner layers are applied on two end node embeddings before computing their dot product which yields the Q-value of the given action.

Table 2: Dataset statistics and agent configurations for Politifact and Gossipcop datasets.

Data	$\lvert U\rvert$	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert V_T\rvert$	Agent	Δ_u	Δ_e
POL	276,277	581	1,074,890	62	1	100	1
					2	50	3
					3	20	5
GOS	565,660	10,333	3,084,931	1,547	1	1,000	1
					2	500	3
					3	100	5

5 EXPERIMENTS

As mentioned in Section 4.1.2, indirect attacks do not modify the edges between user nodes and target news nodes directly and are more likely to be used by attackers in practice to evade detection. In this section, we conduct a series of experiments to validate the effectiveness of our proposed framework under the more realistic

Table 3: Performance of surrogate models measured by accuracy and F-1 score.

Models	Politifact		Gossipcop	
	Accuracy	F1	Accuracy	F1
GCN	0.8673	0.8632	0.8278	0.7864
GAT	0.8600	0.8543	0.8423	0.8010
SAGE	0.8034	0.7973	0.8824	0.8636

Table 4: Results of using MARL to perform *indirect* targeted attacks comparing to several baselines. Experiments are repeated five times, and the average success rate is reported.

Method	Politifact						Gossipcop					
	Fake			Real			Fake			Real		
	GAT	GCN	SAGE	GAT	GCN	SAGE	GAT	GCN	SAGE	GAT	GCN	SAGE
RD-Edge	0.14	0.45	0.13	0.11	0.33	0.15	0.06	0.28	0.25	0.08	0.22	0.14
RD-Node	0.12	0.48	0.14	0.13	**0.38**	0.15	0.12	0.32	0.22	0.12	0.23	0.16
RL - A1	0.17	0.42	0.16	0.08	0.07	**0.21**	0.14	0.45	0.23	0.08	0.80	0.16
RL - A2	0.15	0.38	0.16	0.08	0.13	0.18	0.18	0.52	0.32	0.06	0.83	0.24
RL - A3	0.18	0.64	0.19	0.08	0.13	0.18	0.19	0.51	0.31	0.12	0.85	0.22
MARL	**0.33**	**0.92**	**0.28**	**0.22**	0.31	0.19	**0.21**	**0.64**	**0.36**	**0.18**	**0.89**	**0.28**

indirect settings and analyze the factors that affect MARL's attack performance. Then, we present experiment results to compare direct attack against indirect attack. Finally, we discuss some countermeasures that could be used by defenders. Specifically, we aim to answer the following research questions:

- **RQ1**: How does the performance of MARL compare to baselines?
- **RQ2**: What factors affect the performance of MARL?
- **RQ3**: How does direct attack compare with indirect attack?
- **RQ4**: What are some countermeasures against attacks?

5.1 Experiment Settings

In this subsection, we introduce the experiment settings for MARL indirect attacks. We first introduce the datasets, surrogate models, and baseline methods used for the experiments, then we introduce the implementation details.

5.1.1 Datasets. We extract two social engagement graphs from the FakeNewsNet [38] dataset composed of the metadata of fake and real news posts and their engaged users on Twitter from two fact-checking sources: Politifact and Gossipcop. Following [10], we take Glove 300D [33] embedding of a user's historical tweets as its feature and the Glove embedding of the associated news content of a social post as its features. Note that our attack operates the controlled users to share posts; since the number of changed edges for a user is within a tight budget, we assume all node features are unchanged during the attack process.

5.1.2 Surrogate Models. Under the grey-box setting, the attacker only has information about the architecture of the model being attacked. Thus the attack has to be performed on a surrogate model $\mathcal{M}$ that has the same GNN architecture as the target model. For the GNN-based fake news detectors, we include three classic GNN models. Specifically, we use Graph Convolution Network [22], Graph Attention Network [45], and GraphSAGE[15] as our $\mathcal{M}$. Table 3 shows the performance of these surrogate models on Politifact and Gossipcop. We trained these models to ensure that they have similar performance across both datasets. So that we can measure the attack performance of MARL comparably.

5.1.3 Baseline Attack Methods. Due to the attacker's limited capability and restricted candidates of both controlled users and target posts, we cannot take the feature and gradient-based attacks [50] as baselines. To compare the effectiveness of the proposed MARL framework, we compare it with the following baselines:

- Random-Edge (RD-Edge): This is a simple baseline that randomly selects the controlled users and target posts to add edges until meeting the budget.
- Random-Node (RD-Node): This baseline injects new user nodes into the graph and connects them with the target news nodes.
- Single Agent RL: To demonstrate the effectiveness of MARL, we created this baseline by limiting the attacker to a single type of agent. Specifically, we have three baselines named RL-A1 (Bot), RL-A2 (Cyborg), and RL-A3 (Crowd Worker).

Budget and Target Selection Criteria For the Politifact dataset, we randomly sampled 100 bot agents, 50 cyborg agents, and 20 crowd worker agents from Table 1. For the Gossipcop dataset, we randomly sampled 1,000 bot agents, 500 cyborg agents, and 100 crowd worker agents.

Implementation Details For Random-Edge method, we connect edges between sampled attack agents and news targets based on the agent node's degree. Specifically, we randomly connect bot agents with 1 news target, cyborg agents with 3 news targets, and crowd worker agents with 5 new targets. For Random-Node method, we add 5 user nodes for each of the three agent types. We generate the embedding for each node by randomly sampling 20 nodes from each type of agent, and taking the average of their embedding as the new embedding for the injected node. We connect the generated user nodes with target news nodes the same way in the Random-Edge method. We use PyG [13] to implement all GNN algorithms. The MARL algorithm is implemented based on the RL-S2V code provided by [8]. Our code and data are publicly available [1].

Performance Metrics Since we only aim to flip the classification results of a selected group of target posts, we use the success rate (SR) as the metric to evaluate the attack performance, which is the number of misclassified posts divided by the total number of target posts after the attack.

5.2 RQ1: Performance of MARL

Since attackers are more likely to use indirect attacks than direct attacks to evade detection in practice, we study **targeted indirect attacks** on both fake and real news in Politifact and Gossipcop. Table 4 reports the attack performance of MARL compared to the baselines. From the table, we make the following observation:

- MARL improves the overall attack performance on fake news across both datasets. Compared to the Random-Edge baseline,

[1] https://github.com/hwang219/AttackFakeNews

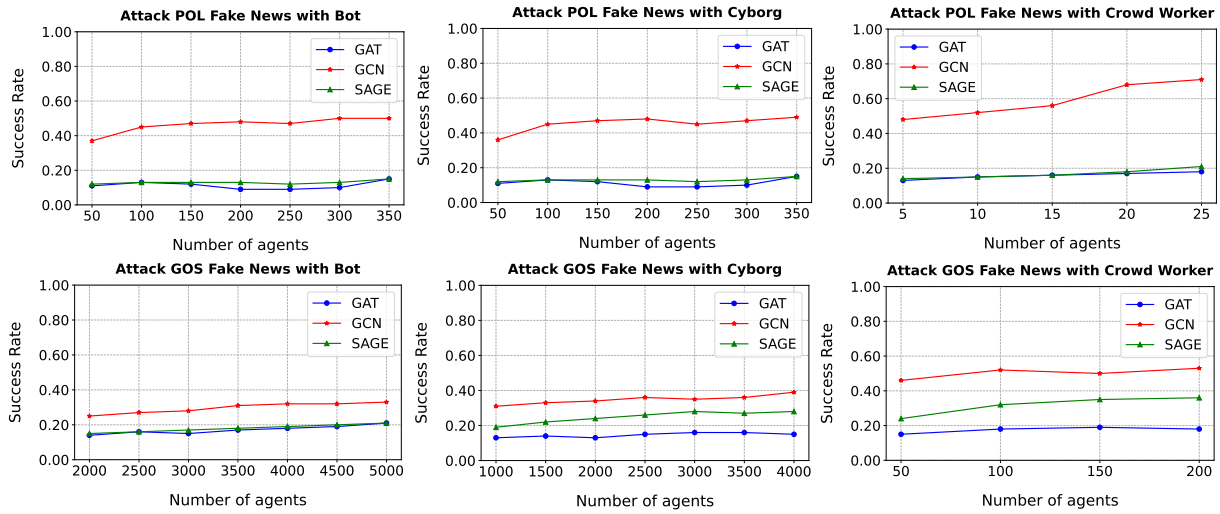

Figure 4: *Indirect* attack performance of different types of agents on fake news in Politifact and Gossipcop datasets. Performance on GAT, GCN, and GraphSAGE are marked in blue, red, and green respectively.

MARL improves the success rate of targeted fake news attacks by [57.6%, 51.1%, 53.6%] respectively for [GAT, GCN, GraphSAGE] on Politifact, and [250%, 128.6%, 44%] on Gossipcop.

- MARL does not improve the performance of attacking real news in Politifact for GCN and GraphSAGE detectors. While MARL improves performance on all three GNN detectors in Gossipcop, the attack performance is worse compared to attacking fake news on GAT and GraphSAGE. This is likely due to the agent configuration of the experiment setting having less influence on real news than on fake news.
- Another interesting finding is that GCN is more sensitive to edge perturbations compared to GAT and GraphSAGE. Attackers can achieve fairly good performance on GCN with only a small amount of edges added to the graph. For instance, we can reach a success rate of 0.48 with just 210 edges added when attacking fake news on Politifact. Comparably, with the same amount of attack budget, the success rate on GAT and GraphSAGE detectors are 0.12 and 0.14 respectively, much lower than GCN. Previous works [7, 8] have also shown that GCNs are vulnerable to structural adversarial attack due to the low breakdown point of their weighted mean aggregation method.

5.3 RQ2: Attack Performance Analysis

In this subsection, we answer RQ2 and provide insights on the factors that affect MARL's attack performance. Specifically, we provide analysis based on agent types and news types.

5.3.1 Agent Types. Figure 4 shows the ablation study results on single agent RL. Specifically, we use RL-A1, RL-A2, and RL-A3 methods to carry out targeted indirect attacks on fake news in Politifact and Gossipcop datasets with increasing attack budgets by using more agents. We make the following observations:

- The overall attack performance increases with incremental attack budget for all three types of agents.
- The performance gain slows down after hitting a threshold. Therefore, attackers need to select the optimum number of agents to perform indirect attacks.
- Crowd worker agents achieve better performance than bot and cyborg agents on all three GNNs given the same amount of attack budgets. This is expected since crowd worker agents have stronger influence and their social posts are connected to real news. Therefore, they exert more influence on fake news.

Based on the above observation, we divide user nodes into "good" and "bad" groups. Specifically, we put users who have more than 80% of the news they shared being fake into the "bad" group and users who have less than 20% of the news they shared being fake into the "good" group.

5.3.2 News Types. Intuitively, we conjecture that the news post node with a higher degree is more robust to attacks than those with lower degrees. To verify this hypothesis, we attack different groups of fake news in Politifact and Gossipcop according to their node degree, or their social popularity. Specifically, we categorize news with less than 10 tweets as low popularity; news with more or equal than 100 tweets as high popularity; and news in between as mid. For this experiment, we use 10 crowd worker agents for Politifact and 50 crowd worker agents for Gossipcop respectively. As shown in Table 5, it is significantly harder to attack news with a higher degree across all three GNNs. Even on the most vulnerable GNN (i.e. GCN), MARL has a significant performance decrease (80%

Table 5: *Indirect* targeted attack on fake news in Politifact and Gossipcop based on their news degrees using crowd worker agents.

News Degrees	Politifact			Gossipcop		
	GAT	GCN	SAGE	GAT	GCN	SAGE
Low	0.16	0.30	0.21	0.14	0.22	0.25
Mid	0.14	0.15	0.11	0.11	0.13	0.12
High	0.03	0.06	0.03	0.02	0.02	0.05

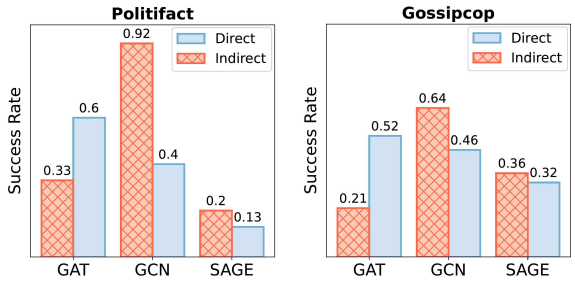

Figure 5: Comparison between the *direct* and *indirect* attack on Politifact and Gossipcop on fake news with degrees less than 10 using "good" users.

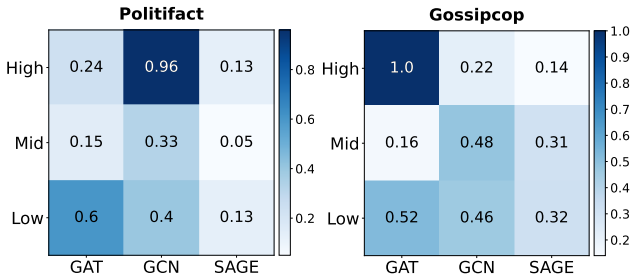

Figure 6: Results of *direct* attack across different types of news based on their degrees. A brighter color suggests better attack performance.

on Politifact and 90.9% on Gossipcop) when attacking fake news with a degree of less than 10 compared to the news with a degree more than 100. Another observation is that GAT is more robust than GCN and GraphSAGE on news with degree between 10 and 100. As shown in Table 5, GAT only has a performance drop of 12.5% and 21.4% on Politifact and Gossipcop respectively when increasing news degree from less than 10 to less than 100. Whereas the performance drop of GCN and GraphSAGE are almost halved on both datasets. This is likely due to the attention mechanism of GAT making it less sensitive to degree changes.

5.4 RQ3: Direct vs. Indirect

Recall from Section 4.1.2, direct attacks mean that attackers modify the edges directly linked from user nodes to target news nodes. Although direct attacks are more obvious in real-world scenarios, they can achieve better performance than indirect attacks due to the direct perturbation of graph structure. For this experiment, we sampled 25 good users for Politifact and 250 good users for Gossipcop to perform targeted attacks on fake news in both datasets. Figure 5 shows the comparison between direct and indirect attacks. We can see that direct attack improves the performance on GAT by 81.8% and 147.6% on both datasets respectively. However, we see a decrease in performance on GCN and GraphSAGE detectors across both datasets. Especially on GCN detectors in the Politifact dataset.

Based on the observation from Figure 5, we are interested in whether the performance of direct attack behaves similarly across news with different degrees. For this experiment, we use the same news degree categorization as in Table 5 and the same agent configuration as in Figure 5 to attack fake news in both datasets. Figure 6 shows that direct attack is effective on news with low and mid degrees on GAT and GCN detectors, while it is less effective on news with high degrees on GraphSAGE.

5.5 RQ4: Countermeasures against Attacks

Based on our experiment findings, we discuss the countermeasures for fraudsters that manipulate news social engagement from two perspectives. **1)** From the machine learning security perspective, there are fruitful research works on defending against graph adversarial attacks [42]. Approaches like adversarial training [12], anomaly detection [11], and robust GNN models [14, 20] can be leveraged to defend the attacks. **2)** From the practical perspective, social media platforms should pay equal attention to both "bot" and seemingly "good" users. As shown in the experiments, attackers can leverage users' good posting history to carry out a successful targeted attack on fake news to foul GNN-based detectors. Since the indirect attack is effective against many GNN detectors, this suggests that the platform should monitor more engagement activities of accounts engaged with the target news instead of the target news itself. Experiment results also show that there is no universally robust model which prompts the platform to adopt diverse trust and safety models.

6 CONCLUSION AND FUTURE WORK

In this paper, we aim to understand the vulnerability of graph neural network-based fake news detectors under structural adversarial attacks. To the best of our knowledge, this is the first work to attack GNN-based fake news detectors. This paper aims to provide insights on how to develop a more robust GNN-based fake news detector against adversarial attacks in the future. We leveraged a multi-agent reinforcement learning framework to mimic the attack behavior of fraudsters in real-world misinformation campaigns. Our experiment results show that MARL improves overall attack performance compared to our baselines and is highly effective against GCN-based detectors.

Even though we have some promising results from the experiments, this paper has two major limitations: **1)** This work only employs a simple heuristic to select users for action aggregation. **2)** The search space of the Q network is considerably large and results in a high computational cost on larger datasets like Gossipcop. Therefore, there are several interesting directions that need further investigation. The first one is to automate the process of selecting optimal agents for action aggregation. The second one is to reduce the deep Q network's search space effectively. Finally, we used a vanilla MARL framework in this paper. It would be interesting to explore a more complex MARL framework for this task.

7 ETHICAL STATEMENT

The Twitter data used in this paper are obtained from Twitter API and meet Twitter user agreement. Although we proposed an adversarial attack framework against GNN-based fake news detectors, our intention is to probe and enhance the robustness of existing detectors. Therefore, we do not endorse this work to be used for unethical purposes in any shape or form.

ACKNOWLEDGMENTS

This work is supported in part by NSF III-1763325, III-1909323, SaTC-1930941, III-2106758, SaTC-2241068, and a Cisco Research Award. This work was done before Yingtong Dou joined Visa Research.

REFERENCES

[1] Hassan Ali, Muhammad Suleman Khan, Amer Alghadhban, Meshari Alazmi, Ahmad Alzamil, Khaled Al-Utaibi, and Junaid Qadir. 2021. All Your Fake Detector are Belong to Us: Evaluating Adversarial Robustness of Fake-News Detectors Under Black-Box Settings. *IEEE Access* 9 (2021).

[2] Alessandro Bessi and Emilio Ferrara. 2016. Social bots distort the 2016 US Presidential election online discussion. *First monday* (2016).

[3] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. In *AAAI*.

[4] Aleksandar Bojchevski and Stephan Günnemann. 2019. Adversarial attacks on node embeddings via graph poisoning. In *ICML*.

[5] Shantanu Chandra, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Graph-based Modeling of Online Communities for Fake News Detection. *arXiv preprint arXiv:2008.06274* (2020).

[6] Canyu Chen, Haoran Wang, Matthew Shapiro, Yunyu Xiao, Fei Wang, and Kai Shu. 2022. Combating Health Misinformation in Social Media: Characterization, Detection, Intervention, and Open Issues. *arXiv preprint arXiv:2211.05289* (2022).

[7] Liang Chen, Jintang Li, Qibiao Peng, Yang Liu, Zibin Zheng, and Carl Yang. 2021. Understanding structural vulnerability in graph convolutional networks. *arXiv preprint arXiv:2108.06280* (2021).

[8] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. 2018. Adversarial attack on graph structured data. In *ICML*.

[9] Yingtong Dou, Guixiang Ma, Philip S Yu, and Sihong Xie. 2020. Robust spammer detection by nash reinforcement learning. In *KDD*.

[10] Yingtong Dou, Kai Shu, Congying Xia, Philip S. Yu, and Lichao Sun. 2021. User Preference-aware Fake News Detection. In *SIGIR*.

[11] Dongsheng Duan, Lingling Tong, Yangxi Li, Jie Lu, Lei Shi, and Cheng Zhang. 2020. Aane: Anomaly aware network embedding for anomalous link detection. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1002–1007.

[12] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. 2019. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering* 33, 6 (2019), 2493–2504.

[13] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop*.

[14] Simon Geisler, Daniel Zügner, and Stephan Günnemann. 2020. Reliable graph neural networks via robust aggregation. *Advances in Neural Information Processing Systems* 33 (2020), 13272–13284.

[15] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.

[16] Yi Han, Shanika Karunasekera, and Christopher Leckie. 2020. Graph Neural Networks with Continual Learning for Fake News Detection from Social Media. *arXiv preprint arXiv:2007.03316* (2020).

[17] Yi Han, Shanika Karunasekera, and Christopher Leckie. 2021. Continual Learning for Fake News Detection from Social Media. In *ICANN*.

[18] Bing He, Mustaque Ahamad, and Srijan Kumar. 2021. PETGEN: Personalized Text Generation Attack on Deep Sequence Embedding-based Classification Models. In *KDD*.

[19] Benjamin D Horne, Jeppe Nørregaard, and Sibel Adali. 2019. Robust fake news detection over time and attack. *TIST* (2019).

[20] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. 2020. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 66–74.

[21] Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Applications* (2021).

[22] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[23] Camille Koenders, Johannes Filla, Nicolai Schneider, and Vinicius Woloszyn. 2021. How Vulnerable Are Automatic Fake News Detection Methods to Adversarial Attacks? *arXiv preprint arXiv:2107.07970* (2021).

[24] Thai Le, Suhang Wang, and Dongwon Lee. 2020. Malcom: Generating malicious comments to attack neural fake news detection models. In *ICDM*.

[25] Jia Li, Honglei Zhang, Zhichao Han, Yu Rong, Hong Cheng, and Junzhou Huang. 2020. Adversarial attack on community detection by hiding individuals. In *WWW*.

[26] Sahil Loomba, Alexandre de Figueiredo, Simon J Piatek, Kristen de Graaf, and Heidi J Larson. 2021. Measuring the impact of COVID-19 vaccine misinformation on vaccination intent in the UK and USA. *Nature human behaviour* (2021), 1–12.

[27] Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. *ACL* (2020).

[28] Yuefei Lyu, Xiaoyu Yang, Jiaxin Liu, Sihong Xie, and Xi Zhang. 2022. Interpretable and Effective Reinforcement Learning for Attacking against Graph-based Rumor Detection. *arXiv preprint arXiv:2201.05819* (2022).

[29] Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake news detection on social media using geometric deep learning. *ICLR Workshop* (2019).

[30] Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. FANG: Leveraging Social Context for Fake News Detection Using Graph Representation. In *CIKM*.

[31] Diogo Pacheco, Alessandro Flammini, and Filippo Menczer. 2020. Unveiling coordinated groups behind white helmets disinformation. In *Companion proceedings of the web conference 2020*. 611–616.

[32] Diogo Pacheco, Pik-Mai Hui, Christopher Torres-Lugo, Bao Tran Truong, Alessandro Flammini, and Filippo Menczer. 2021. Uncovering coordinated networks on social media: Methods and case studies. *ICWSM* (2021).

[33] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. *EMNLP* (2014).

[34] Mrigank Raman, Aaron Chan, Siddhant Agarwal, PeiFeng Wang, Hansen Wang, Sungchul Kim, Ryan Rossi, Handong Zhao, Nedim Lipka, and Xiang Ren. 2020. Learning to Deceive Knowledge Graph Augmented Models via Targeted Perturbation. In *ICLR*.

[35] Yuxiang Ren and Jiawei Zhang. 2020. Adversarial Active Learning based Heterogeneous Graph Neural Network for Fake News Detection. *ICDM* (2020).

[36] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *CIKM*.

[37] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. 2018. The spread of low-credibility content by social bots. *Nature communications* (2018).

[38] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data* 8, 3 (2020), 171–188.

[39] Kai Shu, Deepak Mahudeswaran, Suhang Wang, and Huan Liu. 2019. Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation. arXiv e-prints (2019). *arXiv preprint arXiv:1903.09196* (2019).

[40] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *KDD Explorations* (2017).

[41] Amila Silva, Yi Han, Ling Luo, Shanika Karunasekera, and Christopher Leckie. 2021. Propagation2Vec: Embedding partial propagation networks for explainable fake news early detection. *Information Processing & Management* (2021).

[42] Lichao Sun, Yingtong Dou, Carl Yang, Ji Wang, Philip S Yu, Lifang He, and Bo Li. 2018. Adversarial attack and defense on graph data: A survey. *arXiv preprint arXiv:1812.10528* (2018).

[43] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. 2020. Non-target-specific node injection attacks on graph neural networks: A hierarchical reinforcement learning approach. In *WWW*.

[44] Luis Vargas, Patrick Emami, and Patrick Traynor. 2020. On the detection of disinformation campaign activity with network analysis. *ACM SIGSAC Conference on Cloud Computing Security Workshop* (2020).

[45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. In *ICLR*.

[46] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In *KDD*.

[47] Xinyi Zhou and Reza Zafarani. 2019. Network-based fake news detection: A pattern-driven approach. *KDD Explorations* (2019).

[48] Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *CSUR* (2020).

[49] Zhixuan Zhou, Huankang Guan, Meghana Moorthy Bhat, and Justin Hsu. 2019. Fake news detection via NLP is vulnerable to adversarial attacks. *arXiv preprint arXiv:1901.09657* (2019).

[50] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *KDD*.