

# Identifying Cost-effective Debunkers for Multi-stage Fake News Mitigation Campaigns

Xiaofei Xu

School of Computing Technologies,  
RMIT University  
Melbourne, Victoria, Australia  
s3833028@student.rmit.edu.au

Ke Deng

School of Computing Technologies,  
RMIT University  
Melbourne, Victoria, Australia  
ke.deng@rmit.edu.au

Xiuzhen Zhang\*

School of Computing Technologies,  
RMIT University  
Melbourne, Victoria, Australia  
xiuzhen.zhang@rmit.edu.au

## ABSTRACT

Online social networks have become a fertile ground for spreading fake news. Methods to automatically mitigate fake news propagation have been proposed. Some studies focus on selecting top  $k$  influential users on social networks as debunkers, but the social influence of debunkers may not translate to wide mitigation information propagation as expected. Other studies assume a given set of debunkers and focus on optimizing intensity for debunkers to publish true news, but as debunkers are fixed, even if with high social influence and/or high intensity to post true news, the true news may not reach users exposed to fake news and therefore mitigation effect may be limited. In this paper, we propose the multi-stage fake news mitigation campaign where debunkers are dynamically selected within budget at each stage. We formulate it as a reinforcement learning problem and propose a greedy algorithm optimized by predicting future states so that the debunkers can be selected in a way that maximizes the overall mitigation effect. We conducted extensive experiments on synthetic and real-world social networks and show that our solution outperforms state-of-the-art baselines in terms of mitigation effect.

## CCS CONCEPTS

• **Information systems** → **Social networks**; • **Computing methodologies** → **Reinforcement learning**; • **Mathematics of computing** → *Stochastic processes*.

## KEYWORDS

Fake News Mitigation, Social Network, Reinforcement Learning, Multivariate Hawkes Process

## ACM Reference Format:

Xiaofei Xu, Ke Deng, and Xiuzhen Zhang. 2022. Identifying Cost-effective Debunkers for Multi-stage Fake News Mitigation Campaigns. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22)*, February 21–25, 2022, Tempe, AZ, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3488560.3498457>

\*Xiuzhen Zhang is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

WSDM '22, February 21–25, 2022, Tempe, AZ, USA.

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-9132-0/22/02...\$15.00  
<https://doi.org/10.1145/3488560.3498457>

## 1 INTRODUCTION

With the rapid development of the Internet and mobile devices, people tend to spend more time online and interact with others through social network platforms. Although the social network brings real-time and free news, without the professional editing service, the credibility of news on the social network is lower than traditional media sources. For example, one can intentionally generate some misleading news mixed with true news and spread them on the social network [1]. Despite acknowledging that there might exist misleading information on social networks, about half (53%) of U.S. adults say they get news from social networks “often” or “sometimes”.<sup>1</sup> The spread of fake news at such a scale poses threat to online information security for the online population and society.

To counteract fake news and misinformation, there have been manual fact-checking services as well as algorithms for automatic fake news detection (See [21] for a survey). Notwithstanding these efforts for fake news detection, to effectively counteract fake news at the network scale, arguably it is more important to actively propagate the true news containing correction information such as checked-facts to mitigate the spread of fake news on social networks.

There have been limited studies on automatic fake news mitigation. Some studies heuristically select users of high social influence – having a large number of followers – as debunkers [18, 19] to propagate true news to mitigate the spread of fake news. The assumption is that influential users on the social network produce wide propagation of true news. But research has shown that overall influence on the social network may not translate to wide mitigation information propagation as expected [6]. A recent study assumes a given set of users as debunkers who can broadcast true news to mitigate the spread of fake news [5]. A reinforcement learning agent is trained to decide the optimal intensity for debunkers to post true news with the aim to mitigate fake news spread. Given the dynamic propagation of fake news on the social network, the true news posted by specific debunkers, even if in high intensity, may not reach users exposed to fake news. As a result, this approach may not achieve effective mitigation for the whole social network. In summary, existing studies have not considered how to dynamically select debunkers.

In this paper, we study the problem of selecting debunkers for the multi-stage fake news mitigation campaign and formulate it as a reinforcement learning problem.<sup>2</sup> Based on the *current propagation*

<sup>1</sup><https://www.journalism.org/2021/01/12/news-use-across-social-media-platforms-in-2020/>

<sup>2</sup>A mitigation campaign is also known as an episode in reinforcement learning.

*state* of fake and true news on social networks, our solution dynamically selects debunkers within budget at each stage with the objective to maximize the cumulative mitigation effect – more true news will be propagated to users exposed to more fake news – across stages of the campaign. Different from studies by Saxena [18, 19], our selected debunkers are not necessarily the most influential users on social networks but only those with maximum influence over the users who have been exposed to fake news. Different from the study by Farajtabar [5], our debunkers are dynamically selected for each stage according to the fake news propagation state at the time, that is, the set of debunkers may differ from stage to stage.

It is highly challenging to select multiple debunkers within budget at each stage to achieve optimal fake news mitigation. The search space includes all possible user combinations which increase exponentially with the number of users on social networks. So, a greedy strategy can be adopted where a mitigation policy is trained via a reinforcement learning framework to select one user with the highest cumulative mitigation reward as the debunker, and the policy is repeatedly applied to select multiple debunkers until the budget is exhausted. Such greedy strategy however, may have significant mitigation overlap – the true news posted by debunkers are received by the same users – and as a result, the overall mitigation effect is limited. To address this issue, we propose a greedy algorithm optimized by predicting future states so that the debunkers can be selected in a way that minimizes mitigation overlap and maximizes the overall mitigation effect. Our proposed model DQN-FSP extends the deep  $Q$ -network [14] with future state prediction via the RNN model.

We conducted extensive experiments with synthetic and real-world social network datasets to evaluate DQN-FSP. Experiment results show that DQN-FSP outperforms state-of-the-art baselines. In summary, the contributions of our study are threefold:

- We introduce the problem of selecting cost-effective debunkers within budget for multi-stage mitigation campaigns.
- We formulate the campaign as a reinforcement learning problem to train a mitigation policy that optimizes debunker selection at each stage to maximize the cumulative mitigation across stages for the campaign.
- We propose a greedy algorithm optimized with the RNN model to minimize the mitigation overlap between selected debunkers to improve the overall mitigation effect.

## 2 RELATED WORK

There have been many studies on the detection of fake news on social network. To reduce the cost and time of manual fact-checking, automatic detection of fake news and prediction of the credibility for social network posts have been proposed, using features such as network features [3], multi-modal features [23] or combined features [22][21]. Other studies detect fake news spreaders on social networks using linguistic and personality features [20].

Beyond fake news detection, research on strategies for posting counter true news such as fact-checked contents to mitigate the spreading of fake news on the social network is attracting more attention. Mitigation studies can be categorised into two main classes. One class of studies focus on selecting debunkers to maximize the spread of truthful information to counteract the fake news spread

on social networks. Some studies heuristically select top  $k$  most influential users as debunkers [18, 19]. Their assumption is that users with high social influence produce wide propagation of true news on social networks. But research has shown that overall influence on the social network may not translate to wide mitigation information propagation as expected [6].

Another class of studies focus on optimizing the intensity of posting true news for a given set of debunkers to counteract fake news spread on social networks [5, 7, 8]. As true news are only posted by specific debunkers, even if in high intensity, their propagation may not reach users frequently exposed to fake news given the unknown origin and dynamic propagation of fake news on the social network; the mitigation effect may not be optimal.

Our study falls into the first class of studies of selecting debunkers. But different from previous studies of selecting  $k$  debunkers [18, 19], we focus on selecting cost-effective debunkers within budget for a multi-stage mitigation campaign. Considering the dynamic news diffusion behaviour of users, we aim for a multi-stage mitigation policy such that each stage dynamically selects debunkers according to the *current* propagation state of fake news while at the same time achieving maximal mitigation effect across stages.

Reinforcement learning (RL) has been used in fake news mitigation research [5, 8]. The objective is to predict the intensity for posting true news given specific debunkers, where the value of intensity is continuous. The reinforcement learning agent making decisions requires information on not only the propagation state of fake news and true news but also the environmental factors to infer the future state if applying the intensity. In contrast, our study aims to select top debunkers in multiple stages in reaction to the dynamic fake news propagation and our proposed reinforcement learning framework minimizes the requirement for information about the environment.

The Hawkes process [9] and the multivariate Hawkes process [12] have been widely applied in modelling information propagation on social networks [25] [17]. They model the propagation either in a self-exciting way [17] or in a mutual-exciting way [25]. We use the Hawkes process to model the spread of both fake news and true news in a mutual-exciting way.

## 3 THE MULTIVARIATE HAWKES PROCESS

It is immoral to experiment with real users and spread fake news on real-world social networks, even for research purpose. We use the Multivariate Hawkes process to simulate information propagation on social networks. For modeling news propagation on social networks, the *temporal point process* has been widely used [24] [16]. It can be implemented in neural networks like *recurrent marked temporal point process* [4], *neural Hawkes process* [13] and *neural general temporal point process* [15]. In particular, *multivariate Hawkes process* [12] is a variant of temporal point process and has been widely applied in fake news research [5] [11] [21] [7] [8].

Briefly, the *Hawkes process* is a stochastic temporal point process model with self-excitement which stimulates the occurrence of a sequence of events. Each event occurrence will excite the process to raise the occurrence probability of the next event [9]. Being a

counting process, the Hawkes process can be represented as:

$$N(t) = \sum_{t_\ell \leq t} h(t - t_\ell). \quad (1)$$

where  $t_l$  is the occurring time of  $l$ -th event,  $t$  is the current time,  $h(v)$  is the standard Heaviside function such that  $h(v) = 1$  if  $v \geq 0$  and  $h(v) = 0$  if  $v < 0$  [5]. Equation 1 counts the occurrence of event from time 0 to time  $t$ .

To characterise the self-excitement of the Hawkes process, the conditional intensity function is defined to estimate the probability of an event occurrence during an infinitesimal period of time on the condition of history. Formally,

$$\lambda(t) = \mu + \sum_{t_\ell < t} \phi(t - t_\ell). \quad (2)$$

where  $\mu$  is the base (background) intensity and  $\phi(v)$  is the kernel function. The base intensity is independent of the previous event occurrences while the kernel function is the intensity excited by previous event occurrences. In this paper, we use Hawkes kernel with exponential decay which can be represented as  $\phi(v) = \alpha e^{-\omega v} h(v)$  where  $\alpha$  is the self-exciting coefficient and  $\omega$  is the ratio of kernel decay.

The *Multivariate Hawkes process* (MHP) with  $n$  dimensions is applied in the case of  $n$  event types. In this setting, the conditional intensity function is defined to estimate the occurrence probability of event type  $i$  during an infinitesimal period of time on the condition of history:

$$\lambda_i(t) = \mu_i + \sum_{j=1}^n \sum_{t_{j,\ell} < t} \phi_{i,j}(t - t_{j,\ell}). \quad (3)$$

where  $t_{j,\ell}$  is the occurring time of  $l$ -th event of type  $j$ ,  $\mu_i$  is the base intensity of type  $i$ ,  $\phi_{i,j}(v)$  is the kernel function. MHP expands the Hawkes kernel function from self-excitement to mutual excitation. So,  $\phi_{i,j}(v) = \alpha_{ij} e^{-\omega v} h(v)$  where  $\alpha_{i,j} \in \mathbf{A}$  is the coefficient indicating to which extent event occurrence of type  $j$  influences event occurrence of type  $i$ .  $\mathbf{A}$  is the coefficient matrix.

The propagation of news – both fake and true news – on social networks is modelled by MHP in this study. Specifically, the occurrence of event type  $i$  refers to that user  $i$  posts a piece of news on social networks. For fake news, the intensity functions are  $\lambda^F(t) = (\lambda_1^F(t), \dots, \lambda_n^F(t))^\top$  with base intensity  $\mu^F = (\mu_1^F, \dots, \mu_n^F)$ . For true news, the intensity functions are  $\lambda^M(t) = (\lambda_1^M(t), \dots, \lambda_n^M(t))^\top$  with base intensity  $\mu^M = (\mu_1^M, \dots, \mu_n^M)$ .

## 4 PROBLEM STATEMENT

A social network can be modelled as a directed graph  $G = (U, E)$  where  $U$  denotes graph nodes representing social network users, and  $E$  denotes edges between nodes representing the “following” relationship between users.<sup>3</sup> On graph  $G$ , the origin of fake news is a set of nodes that spread fake news following the MHP at a given intensity; fake news mitigation is achieved by spreading true news to nodes that received fake news. Without loss of generality, we assume all nodes in  $U$  agree to participate in fake news mitigation

<sup>3</sup>We use terms “network” and “graph”, and “user” and “node” interchangeably.

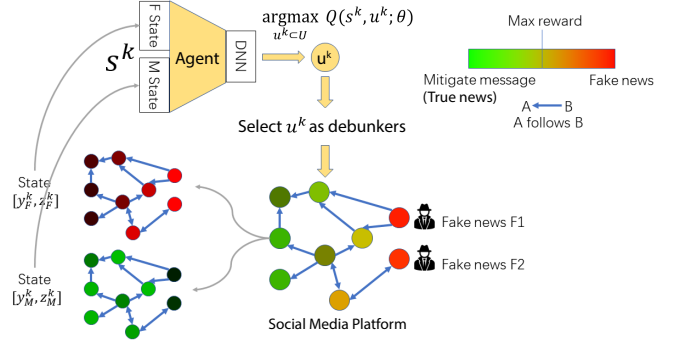


Figure 1: The  $k$ -th stage of a mitigation campaign.

campaigns.<sup>4</sup> Node  $i$  in the graph comes with mitigation cost  $c^i$ . Nodes with more followers will have a higher cost.

A mitigation campaign comprises multiple stages. For the  $k$ -th stage with budget  $l^k$ , a number of nodes are selected as *debunkers* to spread true news on the network following the MHP, under the constraint that the total mitigation cost of debunkers cannot be over  $l^k$ . The aim of a mitigation campaign is to learn the optimal mitigation policy such that the debunkers selected at each stage can maximize the cumulative mitigation effect (or reward) across all stages of the campaign, defined as:

$$V^\pi(s^0) = \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k R^k \mid s^0 \right]. \quad (4)$$

where  $s^0$  is the propagation state of fake news and true news on the graph at the beginning of the mitigation campaign,  $k$  is the identifier of a mitigation stage, and  $\gamma^k$  and  $R^k$  are the discount factor and the reward at mitigation stage  $k$  respectively.

## 5 METHODOLOGY

Given a mitigation campaign of  $N$  stages, we apply reinforcement learning to learn the optimal mitigation policy. The proposed framework is called DQN-FSP, namely *Deep Q-network with Future State Prediction*. Figure 1 shows conceptually how DQN-FSP works at stage  $k$  of a mitigation campaign. The input of agent is  $s^k$  which represents the propagation state of fake news and true news on the (social) network at the beginning of stage  $k$ . Following the mitigation policy learnt so far, the agent selects a number of nodes  $u^k$  as debunkers within budget  $l^k$  to perform the  $k$ -th stage mitigation. At the end of the  $k$ -th stage mitigation, the reward  $R$  is evaluated and the current propagation state of fake news and true news on the network is updated to  $s^{k+1}$ . The  $(k+1)$ -th stage mitigation runs in a similar way if  $k+1 < N$ .

### 5.1 State, Action and Reward

At the beginning of the  $k$ -th stage mitigation, the propagation state of fake news (F) and true news (M) on the network,  $s^k$ , is defined as:

$$s^k = \left[ y_F^k; y_M^k; z_F^k; z_M^k; e \right]. \quad (5)$$

<sup>4</sup>If a subset of nodes  $U' \subset U$  agree, the proposed method selects debunkers from  $U'$  and no other adaption is required.

where  $y_F^k$  and  $y_M^k$  are the conditional intensity of users for fake news and true news respectively. Let  $*$  be either  $M$  or  $F$ .

$$y_*^k = (y_{*,1}^k, y_{*,2}^k, \dots, y_{*,n}^k). \quad (6)$$

where  $y_{*,i}^k = \sum_{j=1}^n \sum_{t_j, \ell < t_k} \phi_{i,j}(t_k - t_{j,\ell})$  for  $1 \leq i \leq n$ . It is equivalent to  $\lambda_i(t_k)$  without  $\mu_i$  (Equation 3), i.e., the conditional intensity of user  $i$  at time  $t_k$  where  $\mu_i$  is ignored since it is a constant.

In the time period from  $(t_k - \Delta_T)$  to  $t_k$ , the number of fake news and true news posted by users on the network are represented as  $z_F^k$  and  $z_M^k$  respectively. They are components of current propagation state of fake news and true news  $s^k$  in Equation 5. Let  $*$  be either  $M$  or  $F$ .

$$z_*^k = (z_{*,1}^k, z_{*,2}^k, \dots, z_{*,n}^k). \quad (7)$$

where  $z_{*,i}^k = \frac{1}{\Delta_T} (N_i(t_k) - N_i(t_k - \Delta_T))$  for  $1 \leq i \leq n$ .

In Equation 5,  $s^k$  also includes  $e$ , which is a vector  $(e_1, \dots, e_n)$  where  $e_i$  for  $1 \leq i \leq n$  is the number of followers of user  $i$  in the network. Note that  $e$  is the only information about the environment that is required by our method and it is typically available directly from any social network. It is noteworthy that in previous studies [5], the reinforcement learning agent requires more information about the environment (such as the ‘‘following’’ relationship between users and the coefficient between users) to predict future state that the selected action may lead to. In our approach, the future state is predicted using an RNN model (Section 5.2) which does not require such environment information.

Based on input  $s^k$ , the agent follows the mitigation policy and takes an action such that a set of users  $u^k$  are selected as debunkers within stage budget, where  $\text{argmax}_{u^k \subset U} Q(s^k, u^k; \theta)$ , and  $\theta$  is the set of parameters of mitigation policy learnt so far. In other words, given  $s^k$ , selecting  $u^k$  will maximize the expected cumulative reward, or mitigation effect,  $Q(s^k, u^k; \theta)$ .

For the  $k$ -th stage mitigation, the reward is evaluated by *correlation maximization* [5]. It is principled on that users exposed to fake news are also exposed to true news. The reward measure is defined as:

$$r(s^k, u^k) = \frac{1}{n} \mathcal{M}^k(t_{k+1}; s^k, u^k)^\top \mathcal{F}^k(t_{k+1}; s^k, u^k) \quad (8)$$

where  $\mathcal{M}^k(*) = (\mathcal{M}_1^k(*), \dots, \mathcal{M}_n^k(*))$  and  $\mathcal{F}^k(*) = (\mathcal{F}_1^k(*), \dots, \mathcal{F}_n^k(*))$ .

$$\mathcal{M}_i^k(t; s^k, u^k) = \frac{1}{t - t_k} \sum_{j=1}^n b_{ij} (N_j^M(t) - N_j^M(t_k)). \quad (9)$$

$$\mathcal{F}_i^k(t; s^k, u^k) = \frac{1}{t - t_k} \sum_{j=1}^n b_{ij} (N_j^F(t) - N_j^F(t_k)). \quad (10)$$

where  $t_k$  is the starting time of the  $k$ -th mitigation stage,  $t$  is the current time,  $N_j^M(t) - N_j^M(t_k)$  is the number of times that user  $j$  spread true news during the time period from  $t_k$  to  $t$ ,  $N_j^F(t) - N_j^F(t_k)$  is the number of times that user  $j$  spreads fake news during the time period from  $t$  to  $t_k$ .  $\mathbf{B}$  is an adjacency matrix. Given users  $i$  and  $j$ , for  $b_{ij} \in \mathbf{B}$ , if user  $j$  follows user  $i$  in the network we have  $b_{ij} = 1$ , otherwise  $b_{ij} = 0$ .

## 5.2 The Multi-stage Mitigation Campaign

A mitigation campaign consists of a sequence of mitigation stages. Following some mitigation policy, the agent takes actions to select as many users as possible as debunkers within the budget at each stage so that the cumulative reward for the campaign is maximized. But the search space for debunkers is exponentially large with respect to the total number of users. A feasible solution is the greedy strategy of applying some mitigation policy to select debunkers at each stage. A straightforward policy is to select one debunker with the highest reward each time; the policy is then repeatedly applied to select multiple debunkers until the budget for the stage is exhausted. But this policy may lead to the issue of mitigation overlap where the true news posted by multiple debunkers are received by the same users. With our model DQN-FSP, we propose the policy to select multiple debunkers. We next describe these two policies in detail.

**5.2.1 A One-debunker Mitigation Policy.** At the beginning of the  $k$ -th stage, a one-debunker mitigation policy modelled as the DQN is applied to select one user  $u^k$  ( $|u^k| = 1$ ) as the debunker given state  $s^k$ . That is, the selected user  $u^k$  can lead to the maximum  $Q(s^k, u^k; \theta)$ . At the end of the  $k$ -th stage, the reward  $R^k$  is evaluated and the current propagation state of fake news and true news on the graph is updated to  $s^{k+1}$ . A new training data instance  $(s^k, u^k, R^k, s^{k+1})$  of the one-debunker mitigation policy is created. Specifically,  $Q(s^k, u^k; \theta)$  is updated as:

$$Q(s^k, u^k; \theta) = \mathbb{E} \left[ R^k + \gamma \max_{u^{k+1}} Q(s^{k+1}, u^{k+1}; \theta) \mid s^k, u^k, \theta \right], \quad (11)$$

where  $\gamma \in [0, 1]$  is a discount factor to control the influence of future reward. When  $\gamma = 1$ , all future rewards are fully considered and treated equally. When  $\gamma = 0$ , only instant reward is considered. The loss used to train the network is calculated as follows:

$$L(\theta) = \mathbb{E}_{s^k, u^k, R^k, s^{k+1}} \left[ \left( y^{DQN} - Q(s^k, u^k; \theta) \right)^2 \right], \quad (12)$$

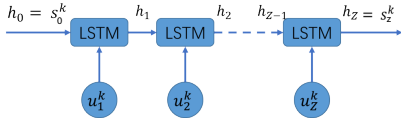
$$y^{DQN} = \left( R^k + \gamma \max_{u^{k+1}} Q(s^{k+1}, u^{k+1}; \theta^-) \right). \quad (13)$$

where  $\theta$  is the weight of the online network while  $\theta^-$  is the weight of the target network, which is updated with the online network regularly.

The learning procedure of the one-debunker mitigation policy is presented in Algorithm 1 where a single user is selected as the debunker for  $k$ -th mitigation stage. Note that  $U_{legal}^k$  ( $l^k$ ) is the subset of users whose mitigation cost is less than the stage budget  $l^k$ .

**5.2.2 Selecting Multiple Debunkers with DQN-FSP.** To minimize the mitigation overlap, we propose an RNN model to predict the future state that the currently selected debunkers may lead to, and thus when the agent selects the next debunker, it will avoid those debunker candidates with overlapping mitigation effect.

Suppose the one-debunker mitigation policy introduced in Section 5.2.1 has been well trained. At the beginning of  $k$ -th stage, let  $H^k$  be the set of debunkers initialized to be empty. We can use the one-debunker mitigation policy to select the first user  $u_1^k$  and move it from  $U$  to  $H^k$ . In the same way, we can select the second user  $u_2^k$ . To avoid the mitigation effect of  $u_2^k$  overlapping with that of  $u_1^k$ ,



**Figure 2: Future state prediction with the LSTM RNN model.**

we need to know the propagation state  $s_1^k$  of fake news and true news after  $u_1^k$  spreads true news. But  $s_1^k$  is unknown since the  $k$ -th mitigation stage does not start yet.

To conquer this problem, we propose a model based on LSTM [10] to predict the unknown state  $s_1^k$ . As shown in Figure 2, LSTM [10] is used where  $s_0^k$  is the initial state of  $k$ -th mitigation stage and  $s_z^k$  is the state after a sequence of debunkers spread true news. For example, we can predict state  $s_1^k$  after the first debunker  $u_1^k$  spreads true news; with the estimated state  $s_1^k$ , we can predict  $s_2^k$  after the second debunker  $u_2^k$  spreads true news, and so on. The pseudo-code of LSTM-based greedy algorithm is presented in Algorithm 2.

The training data for the LSTM is collected when training the one-debunker mitigation policy in Section 5.2.1. Given  $s^k$  and  $u^k$ , the training data instance of one-debunker mitigation policy is  $(s^k, u^k, R^k, s^{k+1})$ . Training data instance of LSTM is  $(s^k, u^k, s^{k+1})$ . The training process is shown in Algorithm 1.

---

#### Algorithm 1 Single-debunker Mitigation Policy

---

```

1: Initialize DQN replay memory  $D_{DQN}$ , FSP memory  $D_{fsp}$ 
2: Initialize action-value function  $Q$  with random weights  $\theta$ 
3: Initialize target action-value function  $\hat{Q}$  with weights  $\theta^- = \theta$ 
4: Initialize FSP with random weights  $\theta_{fsp}$ 
5: for episode = 1, E do
6:   Initialize state  $s_0$  and budget for every stage  $l^k$ ;
7:   for k = 1, K do
8:     Observe environment to obtain state  $s_k$ ;
9:     Select action  $u^k = \operatorname{argmax}_{u \in U_{legal}^k(l^k)} Q(s^k, u; \theta)$ ;
10:    Perform the single-debunker  $u^k$  mitigation task;
11:    Observe reward  $r^k$  and updated state  $s^{k+1}$ ;
12:    Store  $(s^k, u^k, r^k, s^{k+1})$  in  $D_{DQN}$ ;
13:    Store  $(s^k, u^k, s^{k+1})$  in  $D_{fsp}$ ;
14:    Update  $\theta$  using sampled minibatch from  $D_{DQN}$ ;
15:    Every  $C$  steps reset  $\hat{Q} = Q$ ;
16:   end for
17:   Update  $\theta_{fsp}$  using sampled minibatch from  $D_{fsp}$ ;
18: end for

```

---

## 6 EXPERIMENTS

We evaluated our DQN-FSP model on synthetic data with controlled settings and real-world social network data. We ran our experiments on a Slurm cluster consisting of 4 CPU nodes (2 x Intel Xeon E5-2450L, 64G Ram) and 1 GPU node (2 x Intel Xeon E5-2650 v2, 64G Ram, 2 x NVIDIA Tesla M40). All deep networks including DQN and LSTM are implemented in the Tensorflow framework and MHP is implemented using the Tick package [2].

### 6.1 Baselines

DQN-FSP is compared against six baselines, including:

---

#### Algorithm 2 Multi-debunker Mitigation with DQN-FSP

---

```

1: Initialize FSP memory  $D_{fsp}$ 
2: Initialize action-value function  $Q$  with weights  $\theta$  trained from single node selection
3: for episode = 1, E do
4:   Initialize state  $s_0$ , budget for every stage  $l^k$ 
5:   for k = 1, K do
6:     Initialize  $H^k$  empty;
7:     Observe state  $s_k$  and set hidden state LSTM to be  $s^k$ ;
8:     while  $\exists u_i \in U, c^i < l^k$  do
9:       Select action  $u^k = \operatorname{argmax}_{u \in U_{legal}^k(l^k)} Q(s^k, u; \theta)$ ;
10:      Feed  $u^k$  into LSTM, obtain predicted  $s^{k'}$ ;
11:      Update  $s^k = s^{k'}$ ,  $l^k = l^k - c^{u^k}$ ;
12:     end while
13:     Perform multi-debunkers  $H^k$  mitigation task;
14:     Observe reward  $r^k$  and updated state  $s^{k+1}$ ;
15:     Store transition  $(s^k, H^k, s^{k+1})$  in  $D_{fsp}$ ;
16:   end for
17:   Update  $\theta_{fsp}$  using sampled minibatch from  $D_{fsp}$ ;
18: end for

```

---

- *Random* (RND): This method is used as a sanity check for all other models, including the baselines. At each stage, it randomly selects a set of nodes as debunkers within budget.
- *Max Influence* (MAX-INF): This method is based on the policy of selecting nodes with the maximal influence [18] [19]. For node  $i$ , the influence is  $p_i^k = z_{M,i}^k z_{F,i}^k$  where  $z_{*,i}^k$  is as defined in Section 5.1.
- *Max Coverage* (MAX-COV): This method is an intuitive baseline that maximizes the number of mitigation nodes within budget at each stage. So this method sorts nodes by their mitigation cost and selects the cheapest node first.
- *Neural Network* (NN): This method is a learning-based baseline. A classifier is trained. For each training data instance, the current propagation state of fake news and true news is input, and output is the direct mitigation reward (Equation 8) after selecting this node as the debunker to spread true news. At each stage, the trained classifier estimates the direct reward for each node and selects a set of nodes within budget with the highest direct rewards.
- *Deep Q-Network* (DQN): As described in Section 5.2.2, this is the straightforward implementation of the one-debunker mitigation policy to select multiple debunkers at each stage. Different from baseline NN, the objective of selection is to maximize cumulative reward rather than the direct reward.
- *Least-squares Temporal Difference* (LTD): This method follows the idea of existing studies where the same set of debunkers are applied at different stages throughout a mitigation campaign [5].

### 6.2 Performance on synthetic data

**6.2.1 Parameter Settings.** Unless stated otherwise, the graph has  $n = 100$  nodes. Between any two nodes, the edge was generated with probability 0.02. The parameters in Equation 3 were set as follows. The coefficient matrix  $\mathbf{A}$  was set as  $\mathbf{A}_{fc} \odot \mathbf{B}$ ,  $\mathbf{A}_{fc} = a_{ij} \sim \mathcal{U}[0, 0.5]$  where  $\mathbf{A}_{fc}$  is the coefficient matrix for all nodes and  $\mathbf{B}$

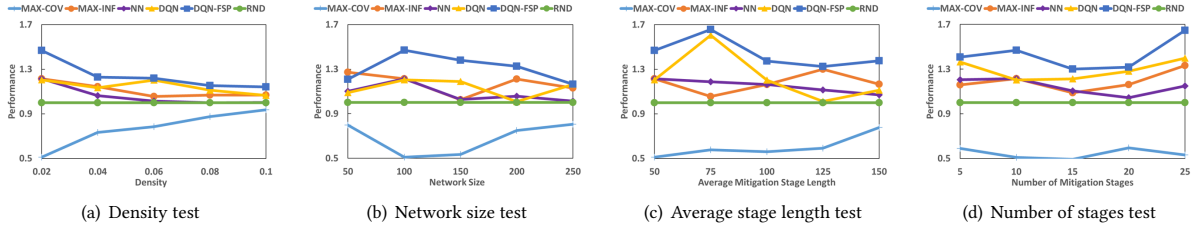


Figure 3: Performance with respect to different settings on synthetic test.

is the adjacency matrix (see Section 5.1). The coefficient matrix  $A$  was scaled such that the spectral radius is 0.8 to keep MHP stable. The parameter  $\omega$  of kernel function was set to 1. The base intensity of mitigation was set to  $\mu^M \sim \mathcal{U}[0, 0.1]$  and the base intensity of fake news was set to  $\mu^F \sim \mathcal{U}[0, 0.2]$  to simulate scenarios that fake news already exist on social networks.

In the default environment settings, we assume that 5 nodes are the fake news spreaders and all the other nodes can be selected as debunkers. A mitigation campaign spans a time window of size 500 and has 10 mitigation stages. The starting time of each mitigation stage is a random value in  $[0, 500]$ . For each node, the mitigation cost is a value in  $[1, 5]$ . A node has a higher mitigation cost if having more followers. For each mitigation stage, the budget  $l^k$  is a random value in  $[5, 50]$ . Once a node is selected as a debunker, the intensity of the node to spread true news is increased by 3.

For the cumulative mitigation reward in Equation 11, the discount factor  $\gamma = 0.8$ . For Equation 7,  $\Delta_T = 25$ . In experiments, 200 mitigation campaigns (i.e., 200 episodes) are directly processed by the baselines *RND*, *MAX-INF* and *MAX-COV*; for *NN*, *DQN* and *DQN-FSP*, 100 of them are training data and other 100 are test data.

**6.2.2 Impact of Environment settings.** The performance is measured by the average cumulative mitigation effect using our *DQN-FSP* and baselines on test data (3 runs on 100 mitigation campaigns) where parameters are randomly set in the given ranges as discussed in Section 6.2.1. In Figure 3, using *RND* as the benchmark, the performance of our *DQN-FSP* and other baselines are presented as the ratio against the benchmark.

Figure 3(a) shows the performance with respect to the density of the social network. By default, a social network has 100 nodes and the edge was generated with a probability of 0.02 between any two nodes. The probability of edge generation increases from 0.02 to 0.1 to simulate different levels of density. We can observe the significant advantage of our *DQN-FSP* against all baselines at different density levels. In particular, *DQN-FSP* outperforming *DQN* shows the effectiveness of our future state prediction strategy. While others have declining performance when the social network becomes more dense (i.e., density changes from 0.02 to 0.1), the performance of *MAX-COV* keeps increasing. This confirms previous findings [5] that if the social network is highly dense, no matter which nodes are selected as debunkers, all nodes will be exposed to true news. But in the real world, social networks are usually sparse. In addition, the experiment results show that *MAX-COV* has clearly worse performance compared with *RND* at different settings, that is, selecting cheapest nodes (i.e., with least followers) is worse than selecting nodes randomly.

Table 1: Statistics of the real-world PHEME dataset

Topic	#Users	Fake tweets	True tweets
Gurlitt (GUR)	98	70	159
Prince Toronto (PRI)	322	483	489
Putin Missing (PUT)	352	251	468

Figure 3(b) shows the performance with respect to the social network size. By default, the social network has 100 nodes and the edge was generated with a probability of 0.02 between any two nodes. The number of nodes changes from 50 to 250 to simulate different sizes of the social network. Our *DQN-FSP* model demonstrated consistently better performance than all baselines.

**6.2.3 Impact of Mitigation Settings.** Figure 3(c) shows the performance with respect to the average stage length (size of the time window of the mitigation campaign changes accordingly). Longer stage length means the actions taken by the agent at the beginning of the stage will last longer and thus have more effect on the environment. From the figure, we can see that the proposed method performs well in all settings.

Figure 3(d) shows the performance of different models with respect to the number of stages in the mitigation campaign. More stages will have more opportunities to optimize mitigation according to the propagation state of fake news and true news at the time and will have more chances to increase the intensity for selected debunkers to spread true news. The results have validated that the performance tends to be improved with an increasing number of stages. With an increasing number of stages, the performance of most methods against *RND* has been improved. At different settings, *DQN-FSP* outperforms all other baselines.

## 6.3 Performance on real-world data

**6.3.1 Dataset and Settings.** PHEME [26], a widely used dataset for rumour spread on Twitter, has been used in our experiments. It includes the source and timestamp of Twitter messages – who post the tweets and when – and the spread of messages – who retweets – for three news topics, including “Gurlitt” (GUR), “Prince Toronto” (PRI) and “Putin Missing” (PUT). Statistics about the dataset is shown in Table 1.

We have used the data to learn the environment parameters ( $A, \mu$ ) using least square loss. The coefficient matrix was scaled so that the spectral radius is 0.8 to keep MHP stable. Since the dataset does not have data for the social network, we generate edges for the network in different settings to test the robustness of the proposed method. Unless stated otherwise, the network density is set to 0.02. The decay of kernel function is set as  $\omega = 1$ . Once a user is selected as a debunker, the mitigation intensity is increased by 1. Mitigation stage and other settings follow the settings for the synthetic data.

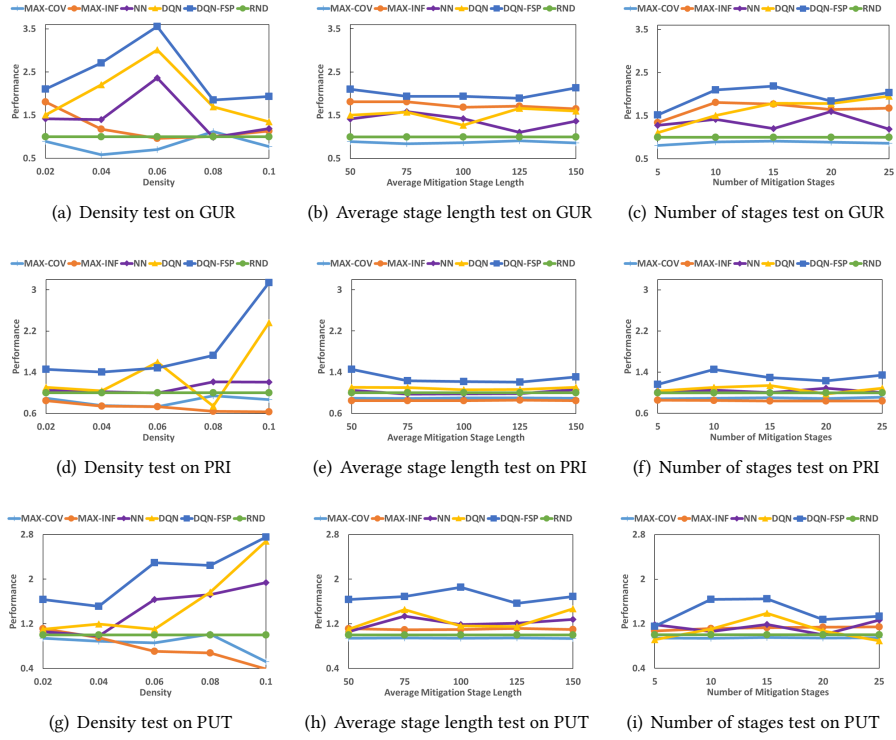


Figure 4: Performance with respect to different settings on real-world data.

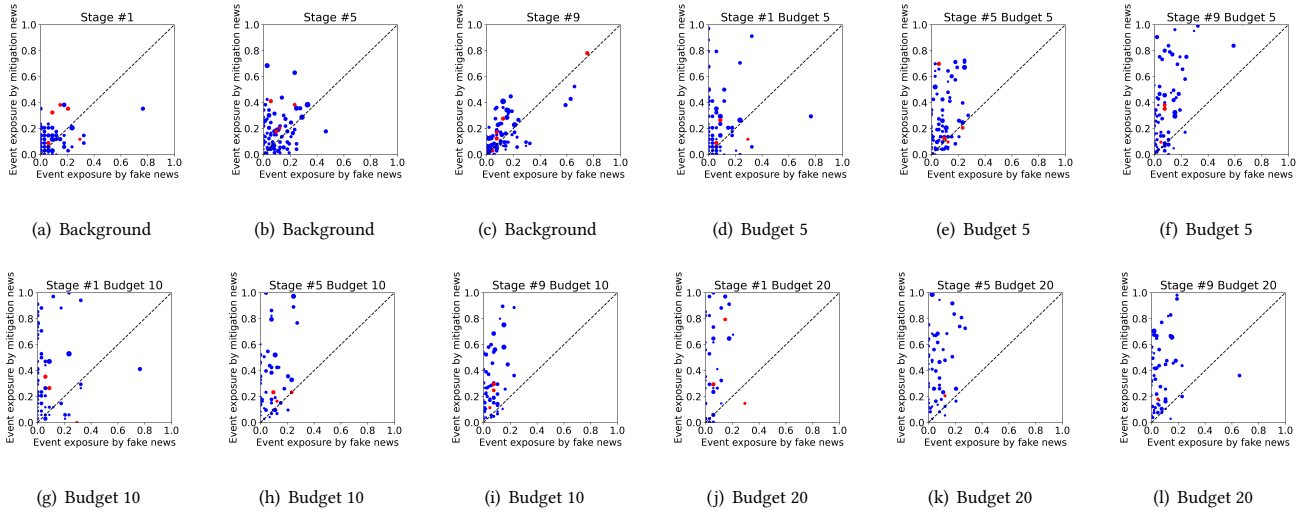


Figure 5: Analysis of DQN-FSP at different budget settings.

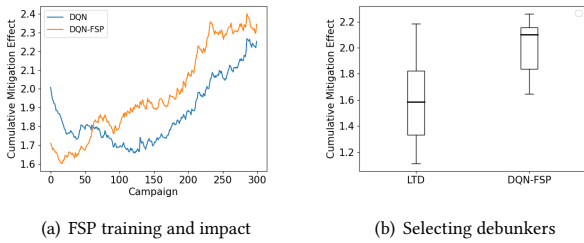


Figure 6: Analysis of DQN-FSP.

6.3.2 *Experiment Results.* The performance is measured by the average cumulative mitigation effect using our DQN-FSP and baselines (3 runs on 100 mitigation campaigns). With the performance of *RND* as the basis, Figure 4 shows the relative performance against *RND* at different settings on different datasets. Wherever it is required, the relevant parameters are randomly set in the given ranges as discussed in Section 6.2.1.

Figure 4 (a)(d)(g) show the performance with respect to the density of the social network. It illustrates that our DQN-FSP outperforms baselines at almost all settings on different datasets. The

performance trends in (d) and (g) are similar and are different from those in (a). The reason is that at different density, if the ratio between edge number and node number is 1 for GUR, the ratio is about 3 for PRI and PUT. Compared with Figure 3 (a) where synthetic data has a ratio similar to that of GUR, we can observe it has a similar performance trend as GUR shown in Figure 4 (a). Moreover, comparing *MAX-INF* with *RND*, the performance of *MAX-INF* declines when the number of edges per node increases. The reason is that all nodes have more influence and thus the influence of randomly selected nodes are comparable with that of nodes selected using *MAX-INF*.

Figure 4 (b)(e)(h) show the performance with respect to the average stage length. As mentioned in the synthetic experiment results, the longer stage length means the actions at the beginning of the stage will last longer and thus have more effect on the environment. The experiment results show our DQN-FSP outperforms baselines at different settings on different datasets.

Figure 4 (c)(f)(i) show the performance with respect to the number of stages. As discussed above, with more stages the agent has more opportunities to optimize mitigation and more chances to increase the intensity for selected debunkers. At all settings on all three topics, our DQN-FSP outperforms the baselines.

## 6.4 Analysis of DQN-FSP

Various aspects of DQN-FSP are analysed next. All discussions are based on results for the real-world data GUR.

**6.4.1 Budget size.** Intuitively, the performance of DQN-FSP is impacted by the budget at each stage. In Figure 5,  $x$ -axis represents the exposure of fake news and  $y$ -axis represents the exposure of true news where each point represents a user and the red points represent the source of fake news. At the end of a mitigation stage, ideally all users are above the diagonal, i.e., they receive more true news than fake news. To make it comparable, a mitigation campaign has 10 stages where the  $k$ -th stage starts at time  $w_k$  and ends at time  $w_{k+1}$ . Figure 5 (a)(b)(c) illustrate the distribution of users based on the number of fake news and true news received at  $w_2$ ,  $w_6$  and  $w_{10}$  (i.e. at the end of stages 1, 5 and 9), when no mitigation is applied and users send and receive fake news and true news based on background intensity. Clearly, many users are under the diagonal, that is, receiving more fake news than true news. Figure 5 (d)(e)(f) present the distribution of users when mitigation campaigns are applied within budget 5 at the end of stages 1, 5 and 9. Clearly, more users are above the diagonal after stages 1, 5 and 9, but there still exist some users below the diagonal. Note that the total number of users is 98 (including 5 fake news spreaders) and the average user mitigation cost is 2.4. The budget 5 implies that debunkers are 2.13% of all users.

Comparing Figure 5 (g)(h)(i) with Figure 5 (d)(e)(f), the only difference is that budget at each stage is 10 and we observe that most users are above the diagonal after stage 1, and more users are moved above the diagonal after stages 5 and 9. Comparing Figure 5 (j)(k)(l) with Figure 5 (g)(h)(i), the only difference is that budget at each stage is increased to 20. It shows almost all users are above the diagonal. The performance between budget 10 and 20 is trivial. It indicates that budget 10 is sufficient. The results in Figure 5 verify that, with the DQN-FSP mitigation policy, users exposed to more

fake news will receive more true news and more budget results in more effective mitigation at an early stage.

**6.4.2 Future state prediction training and impact.** As discussed above, to minimize the mitigation overlap, we propose an RNN model to predict the future state that the currently selected debunkers may lead to. So, when the agent selects the next debunker, it will avoid those debunker candidates with overlapping mitigation effect. The future state prediction (FSP) accuracy plays a significant role in DQN-FSP. This experiment compared our DQN-FSP against baseline *DQN* at different sizes of training data. From 6 (a), we observe that baseline *DQN* has better performance than DQN-FSP when the training dataset has less than 50 campaigns (episodes). This is because the RNN model is not well trained yet and FSP accuracy is not sufficiently good. Once the training dataset has more than 50 campaigns, FSP has a better performance and in turn, the performance of DQN-FSP becomes better than *DQN* consistently.

**6.4.3 Selecting debunkers.** We compare our DQN-FSP with baseline *LTD* [5] where the same set of debunkers are applied at different stages throughout a mitigation campaign. We have run *LTD* for 50 times (i.e., 50 campaigns) where each campaign randomly selects users as debunkers within the same campaign budget (equally split to stages). The distribution of cumulative mitigation effects for the 50 runs is shown in 6 (b). For DQN-FSP, we have trained 5 different mitigation policies since various settings are randomly selected in specified ranges. Using each of the trained mitigation policies, a mitigation campaign is executed within the same campaign budget as that for *LTD* (but randomly split into stages). The distribution of cumulative mitigation effects for the 5 runs is shown in Figure 6 (b) as well. Clearly, the performance of DQN-FSP is significantly better than that of *LTD*, which shows the benefit of selecting debunkers compared to the fixed debunker approach in previous studies [5] for the multi-stage campaign.

## 6.5 Limitations

Note that similar to existing studies (e.g., [5]), our proposed mitigation policy assumes that the truth value – true or fake – for social media news posts are established and fed to the mitigation process. Errors of upstream fake news detection models therefore can propagate into the mitigation model. Future work can address this limitation in an end-to-end framework.

## 7 CONCLUSION

This paper studied the problem of selecting debunkers for multi-stage fake news mitigation campaigns on social networks. We proposed a reinforcement learning framework to learn a mitigation policy that selects multiple debunkers dynamically within budget for each stage so that the selected debunkers can maximize the overall cumulative mitigation effect across stages. To address the issue of selecting debunkers from an exponentially large search space, we proposed a greedy algorithm with future state prediction so that debunkers are selected in a way that minimizes mitigation overlap and maximizes the overall mitigation effect. For future work, we will model other aspects of fake news propagation for more effective mitigation.



## 8 ACKNOWLEDGMENTS

This research was supported partially by the Australian Government through the Australian Research Council’s Discovery Projects funding scheme (project: DP200101441, DP210100743) and Linkage Projects funding scheme (project: LP180100750).

## REFERENCES

- [1] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives* 31, 2 (2017), 211–36.
- [2] Emmanuel Bacry, Martin Bompain, Stéphane Gaïffas, and Soren Poulsen. 2017. Tick: a Python library for statistical learning, with a particular emphasis on time-dependent modelling. *arXiv preprint arXiv:1707.03003* (2017).
- [3] Adrien Benamira, Benjamin Devillers, Etienne Lesot, Ayush K Ray, Manal Saadi, and Fragkiskos D Malliaros. 2019. Semi-supervised learning and graph neural networks for fake news detection. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 568–569.
- [4] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1555–1564.
- [5] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. 2017. Fake news mitigation via point process based intervention. In *International Conference on Machine Learning*. PMLR, 1097–1106.
- [6] Mehrdad Farajtabar, Xiaojing Ye, Sahar Harati, Le Song, and Hongyuan Zha. 2016. Multistage campaigning in social networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 4725–4733.
- [7] Mahak Goindani and Jennifer Neville. 2020. Cluster-Based Social Reinforcement Learning. *arXiv preprint arXiv:2003.00627* (2020).
- [8] Mahak Goindani and Jennifer Neville. 2020. Social reinforcement learning to combat fake news spread. In *Uncertainty in Artificial Intelligence*. PMLR, 1006–1016.
- [9] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.
- [10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [11] Remi Lacombe. 2018. Fake News Mitigation in Social Networks.
- [12] Thomas Josef Liniger. 2009. *Multivariate hawkes processes*. Ph. D. Dissertation. ETH Zurich.
- [13] Hongyuan Mei and Jason Eisner. 2016. The neural hawkes process: A neurally self-modulating multivariate point process. *arXiv preprint arXiv:1612.09328* (2016).
- [14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.
- [15] Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. 2019. Fully neural network based model for general temporal point processes. *arXiv preprint arXiv:1905.09690* (2019).
- [16] Julio Cesar Louzada Pinto and Tijani Chahed. 2014. Modeling multi-topic information diffusion in social networks using latent Dirichlet allocation and Hawkes processes. In *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*. IEEE, 339–346.
- [17] Marian-Andrei Rizoïu, Young Lee, Swapnil Mishra, and Lexing Xie. 2017. Hawkes processes for events in social media. In *Frontiers of multimedia research*. 191–218.
- [18] Akрати Saxena, Wynne Hsu, Mong Li Lee, Hai Leong Chieu, Lynette Ng, and Loo Nin Teow. 2020. Mitigating misinformation in online social network with top-k debunkers and evolving user opinions. In *Companion Proceedings of the Web Conference 2020*. 363–370.
- [19] Akрати Saxena, Harsh Saxena, and Raluca Gera. 2020. k-TruthScore: Fake News Mitigation in the Presence of Strong User Bias. In *International Conference on Computational Data and Social Networks*. Springer, 113–126.
- [20] Anu Shrestha, Francesca Spezzano, and Abishai Joy. 2020. Detecting Fake News Spreaders in Social Networks via Linguistic and Personality Features. In *CLEF*.
- [21] Kai Shu, H Russell Bernard, and Huan Liu. 2019. Studying fake news via network analysis: detection and mitigation. In *Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining*. Springer, 43–65.
- [22] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* 19, 1 (2017), 22–36.
- [23] 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 *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*. 849–857.
- [24] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1513–1522.
- [25] Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*. PMLR, 641–649.
- [26] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLoS one* 11, 3 (2016), e0150989.