2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)

Learning Adaptive Graph Protection Strategy on Dynamic Networks via Reinforcement Learning

Arie Wahyu Wijayanto
Department of Computer Science, School of Computing
Tokyo Institute of Technology, Japan
ariewahyu@net.c.titech.ac.jp

Tsuyoshi Murata
Department of Computer Science, School of Computing
Tokyo Institute of Technology, Japan
murata@c.titech.ac.jp

***Abstract*—Graph protection strategies aim to suppress the epidemic propagation in a network by allocating protection resources to maximize the ratio of surviving node. Research on this topic has been active and promising due to its wide-range applications. However, most of the recent developments are simulated by assuming that the network structure remains static during epidemics. Moreover, the existing protection schemes are limited to the simplified pre-emptive and post-emptive schemes. The pre-emptive scheme protects the most critical nodes of networks prior to epidemic spreading, behaving as a prevention mechanism. In post-emptive schemes, the protections are allocated in the presence of epidemics, when the attacks have already spread over the network, simulating a late curative response. Given a limited k resource budget, both of those schemes spend the whole resources in a single chance. In this paper, we introduce a novel adaptive protection scheme by gradually protecting nodes in response to the incoming attacks. We consider the adaptive scheme in a more challenging network structure, the dynamic networks. We propose the n-step fitted Q-learning for training the model under reinforcement approach. We further incorporate graph embedding as a feature-based representation of the network state. We also demonstrate the potential of our proposal as a non-deterministic approach for this graph protection problem. Experimental results show that our proposed model effectively restrains epidemic propagation in real-world network datasets.**

***Index Terms*—graph protection, reinforcement learning, time-evolving networks, node immunization, dynamic networks**

I. Introduction

Dynamic networks are networks which change over time by the inclusion and removal of nodes and edges [1], [2]. With the advancement of online social networks such as Facebook, Twitter, Instagram, etc., dynamic networks have become an important topic of studies. Capturing the mechanism of information spreading in this dynamic networks is a challenging task. Furthermore, most social networks have a highly dynamic nature and evolve rapidly over time.

In graph protection problem, we are given an input network and a limited k budget of resources. We aim to select nodes which need protections to maximize the ratio of surviving nodes during epidemic spreading [3], [4]. Intuitively, we should focus to localize epidemic spreading by disconnecting the network. This problem has been studied intensively under the assumption that the underlying network structure remains unchanged as the epidemic propagates. However, the graph protection problem in dynamic networks has received limited attention due to its more complicated topology.

Among the limited work on dynamic networks, most are focusing either on the pre-emptive or post-emptive scheme. The former protects the most critical nodes of networks prior to an epidemic attack, simulating as prevention efforts [4], [5]. In the latter scheme, the protections are distributed in the presence of epidemics, at the point when the attacks have spread over the network, reenacting as delayed reactions [6]–[8]. Given a limited k resource budget, both of those schemes spend entire budget in a solitary shot.

Another challenge in this field is that most of the current works emphasize deterministic approaches such as degree centrality, betweenness, connectivity, etc [5], [9], [10]. Despite its effectiveness, the deterministic approach constantly selects critical nodes by one criterion without considering what is best for a current input network structure. Selecting the highest degree nodes exclusively could result in protecting only a particularly dense area of networks. In this paper, we introduce a novel adaptive protection scheme by gradually selecting nodes to respond to the new incoming attacks which changes over time. We propose the n-step fitted Q-learning for training the model under reinforcement approach. We incorporate graph embedding as a feature-based representation of the network states. We further demonstrate the potential of reinforcement learning and feature-based representation as a non-deterministic policy for this problem. In the experimental evaluation, we show that our proposal effectively restrains epidemic propagation in real-world networks.

Summary of Contributions. In contrast to most prior studies, there are four main contributions of our work:

- **Dynamic Networks.** We omit the simplified assumption that the underlying network structure remains unchanged during the epidemic. We evaluate our model to networks which evolve dynamically over time.
- **Adaptive Scheme.** Instead of spending all the available protection budget at a single time point (preceding or succeeding the epidemic), we propose to gradually selecting nodes in response to the incoming attack.
- **Non-deterministic Policy.** Contrary to most existing works which define a pre-determined protection policy regardless of the given network structure, we introduce a non-deterministic policy. Using reinforcement learning, we introduce the stochasticity to the protection.
- **Protection Threshold.** We prove the existence of a protection threshold, when is achieved, no more pro-

978-1-5386-7325-6/18/$31.00 ©2018 IEEE
DOI 10.1109/WI.2018.00-41

tection resources required as the network is adequately protected. The protection threshold is equal to the minimum vertex cover nodes which can fully disconnect the network.

II. Preliminaries

Definition 1. Graph Protection Problem

We are given the input as follows: an undirected connected graph $G = (V, E)$, SIS propagation model, and a budget k. We define θ to be the surviving ratio of vertices that remain uninfected at the end of infection propagation. Our goal is to find S, a subset of k vertices such that θ is maximized. The protection is performed by removing corresponding edges of S from G to get a new graph $G^{(S)}$.

Definition 2. Dynamic Network

Let $\{1, \cdots, T\}$ be a finite set of discrete time steps. Let $V = \{1, \cdots, n\}$ be a set of individuals. Let $G_t = (V_t, E_t)$ be a graph representing the snapshot of the network at time t. A dynamic network $G = (V, E)$ is a series $\langle G_1, \cdots, G_T\rangle$ of static networks where each $G_t = (V_t, E_t)$ is a snapshot of individuals and their interactions at time t such that $V = \bigcup_t V_t$ and $E = \bigcup_t E_t \cup \bigcup_{t-1}(v_t, v_{t+1})$.

For consistency, the time during which the individuals are observed is assumed as finite. Following the definition by [11] and [1], the temporal length of G is assumed to be divided into discrete steps $\{1, \cdots, T\}$. The interaction between a pair of individuals takes place within one time step [11]. The non-trivial problem of appropriate time discretization is beyond the scope of our work.

Definition 3. SIS Propagation Model

Susceptible-Infected-Susceptible (SIS) model define that each node in graph G with N number of nodes would be in one of the following two states: *susceptible* and *infected*. Let $\mathcal{S}(t)$ be the number of susceptible nodes, and let $\mathcal{I}(t)$ be the number of infected individuals at time t. At each timestamp t, susceptible nodes can be infected by their infected neighbors with probability β. Also, each infected node can get recovered to susceptible state with recovery probability δ. This model can be formalized as nonlinear differential equations:

$$\frac{ds}{dt} = -\beta is, \frac{di}{dt} = \beta is - \delta i, \tag{1}$$

being $s(t) = \mathcal{S}(t)/N$ and $i(t) = \mathcal{I}(t)/N$ the respective proportions of states at time t.

Definition 4. Adaptive Graph Protection Problem on Dynamic Networks Let $G = (V, E)$ be an undirected dynamic graph as an input, with a series of a known sample $\langle G_1, \cdots, G_T\rangle$ of snapshots where each $G_t = (V_t, E_t)$ represent a static network at time t. Let k be a given budget of protection resources and G start at $t = T_0$ to $t = T_t$. The protection and attack of epidemic take place alternately at each time t by turns, while the epidemic propagates to time $t + 1$ under SIS epidemic model continuously until the end of G. Let us denote S, a subset of k protected nodes from graph G and θ be the ratio of surviving nodes of graph G at the end of epidemics. The protection is performed by removing corresponding edges of S from G to get a new graph $G^{(S)}$. Under random attack strategies, k nodes are randomly initialized as infected nodes at each turn such that $k = \sum_{t=1}^{T} k_t$. The goal is to find $S \in V$ such that θ is maximized, subject to the size of S is equal to constraint k, i.e., calculating the following combinatorial optimization:

$$\begin{gathered} S^* = \arg\max_{S \in V} \theta_G(S) \\ \text{s.t. } |S| = k,\, k = \sum_{t=1}^{T} k_t \end{gathered} \tag{2}$$

III. Related Work

Graph protection strategies have mostly been studied by assuming the static topologies of network structure. Chen et al. proposed NetShield [12] and Netshield+ [5] which use the properties of matrix perturbation to find a set of nodes in static networks to be pre-emptively protected [12]. Zhang and Prakash [7], [13] developed DAVA and DAVA-fast, two post-emptive polynomial-time heuristics methods. NIIP [8] extracts a maximum directed acyclic graph from a static network then implements a Monte Carlo simulation to approximate the distribution of k over each time point t given the probability of a functional node getting infected. Wang et al. investigated a rumor blocking in static networks by considering dynamic Ising propagation model which consists of the individual tendency and global popularity of the rumor [14]. Under the constraint of user experience utility, they proposed DRIMUX method to protect a set of nodes in t time interval to limit the spreading of rumor.

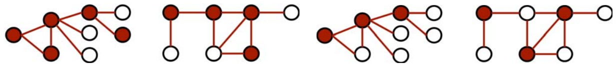
In dynamic networks, VAILDN is introduced as a post-emptive scheme protection [15]. By merging all infected nodes into one supernode and building a tree-like structure, it determines the protected nodes based on each sub-tree benefit comparison. Prakash et al. proposed five different greedy algorithms as pre-emptive protection of the dynamic networks [16]. The methods are based on the pre-defined deterministic protection policies including the highest degree centrality, acquaintance and largest eigenvalue of the system matrix. Liu & Gao investigated a different task of influence blocking in dynamic email networks [17]. They introduced an adaptive Autonomy-Oriented Computing which actively propagates the vaccination patches to counter a virus-embedded email spreading.

To summarize, none of the existing works investigated the adaptive scheme of suppressing the epidemic spreading by graph protection strategies in dynamic networks.

IV. ReProtect: Adaptive Graph Protection Strategy in Dynamic Networks

Our proposal of adaptive graph protection strategy aims to incrementally protect the selected nodes instead of protecting them at once. In this scheme, the protection strategy is divided into several rounds, and each protection round is performed to block the epidemic attack being spread. In each round, we are given a snapshot of the current network structure.

In each protection round, we determine the most critical set of nodes of the current network structure. The main idea of our learning method will be described in the following key points:



(a) Vertex cover (not minimum) (b) Minimum vertex cover

Fig. 1: Vertex covers shown on the same underlying graphs, where a highlighted node indicates that it is contained in the cover. Red colored edge indicates the covered edge.

A. Minimum Vertex Cover (MVC)

Given a graph $G = (V, E)$, a vertex cover is a subset of the vertices $V_c \subseteq V$ such that every edge has at least one of its side in V_c. Thus, this set of vertices V_c in graph G cover every edge in G. A minimum vertex cover is a vertex cover with the smallest possible number of nodes.

Every graph G trivially has a vertex cover where $V_c = V$. Figure 1a shows the vertex cover, and Figure 1b shows the minimum vertex cover for the same graphs. The complexity of vertex cover problem is NP-complete, and that of the minimum vertex cover problem is NP-hard.

Theorem 1 (Protection Threshold)**.** *The protection threshold is the minimum required size of S to disconnect graph G such that no propagation may occur among vertices. Given an undirected connected graph $G = (V, E)$, a minimum vertex cover of G is also a protection threshold of G.*

Proof: A vertex cover V_c of G is a subset of the vertices $V_c \subseteq V$ such that $(u, v) \in E \Rightarrow u \in V_c \vee v \in V_c$. A minimum vertex cover V_c^* is a V_c with the smallest size as follows:

$$V_c^* = \arg\min_{V_c} |V_c| \tag{3}$$

Since all edges in graph G is covered by V_c^*:

$$(u, v) \in E \Rightarrow u \in V_c^* \vee v \in V_c^*, \tag{4}$$

then by removing all corresponding edges in G connected to V_c^* we get $G^{(V_c^*)} = (V_c^*, E^{(V_c^*)})$. Thus, $G^{(V_c^*)}$ has no edge, i.e., $E^{(V_c^*)} = \{\}$, $|E^{(V_c^*)}| = 0$.

Following Definition 1, protecting the set S of vertices in G is removing all corresponding edges connected to S from G. This is a minimax function of minimizing the size S to get the maximum edges in G covered as follows:

$$S^* = \arg\min_{S} |\arg\max_{E^{(S)}} |E^{(S)}|| \tag{5}$$

Consequently, by protecting minimum vertex cover V_c^*, i.e., $S = V_c^*$, then $G^{(S)}$ has no edge. Hence, a minimum vertex cover V_c^* of G is also a protection threshold of G. ■

Next, we will describe how to select protected nodes from a minimum vertex cover under a limited budget.

B. k Degree Ordered MVC vertices

Let us recall that MVC is a set of vertices without any requirement of ordering. Intuitively, given k budget, selecting any k nodes from V_c^* may result in a different set of nodes. Additionally, not all of the node in MVC should have the same priority to be protected within a limited budget. We consider that the more connected a node v to its neighbors in G, the more critical node v to be protected. Hence, we reorder MVC nodes using their respective degree value. Under the constraint of budget k, we get the top k highest degree nodes of V_c^*.

Algorithm 1: Training Phase: n-step Fitted Q-Learning for the Minimum Vertex Cover

```
   Input: adjacency list of graph $G$
   Output: neural network parameter $\Theta$
 1 Initialize experience replay memory $M$ to capacity $N$
 2 for episode $e = 1, ..., L$ do
 3     Initialize the state to empty $\mathbb{S}_1 = \{\}$
 4     for step $t = 1, ..., T$ do
 5         $v_t = \begin{cases} \text{random node } v \in \bar{\mathbb{S}}_t, & \text{w.p.}\epsilon \\ \arg\max_{v \in \bar{\mathbb{S}}_t} \hat{Q}(h(\mathbb{S}_t), v; \Theta), & \text{otherwise} \end{cases}$
 6         Add $v_t$ to partial solution: $\mathbb{S}_{t+1} = (\mathbb{S}_t, v_t)$
 7         if $t \geq n$ then
 8             Add tuple $(\mathbb{S}_{t-n}, v_{t-n}, \mathbb{R}_{t-n,t}, \mathbb{S}_t)$ to $M$
 9             Sample random batch from $B \overset{iid.}{\sim} M$
10             Update $\Theta$ by SGD over loss function
                 $(y - \hat{Q}(h(\mathbb{S}_t), v_t; \Theta))^2$ for $B$
11     end
12 end
13 return $\Theta$
```

C. Reinforcement Learning

Here, we will explain how to get the set of minimum vertex cover from input graph. Despite the protection threshold guarantee of MVC, finding the MVC nodes of graphs is NP Hard. We consider a reinforcement learning approach to approximate the solution. More specifically, we leverage an n-step fitted Q-Learning [18], [19] to train our model in the neural network framework.

Training Phase

In the training phase, we iteratively let our model to construct a vertex cover (V_c) solution of the input network. We define the RL environment as follows:

- State ($\mathbb{S}$): set of currently selected nodes
- Action ($\mathbb{A}$): add new node v to vertex cover set $\mathbb{S}$
- Reward ($\mathbb{R}$): -1, as our goal is to get the minimum size of vertex cover, we set a penalty for adding a new node into V_c set.
- Termination criteria: all edges are covered

Algorithm 1 illustrates our proposed training phase. In each training iteration, our method return the neural network parameter Θ which succesfully get V_c from graph G. In line 5, we specify how to select a new node by balancing exploration and exploitation. With probability ϵ, we select a random node as an exploration effort, otherwise exploit the best known policy by adding a node which satisfies the evaluation function $\hat{Q}(h(\mathbb{S}_t), v; \Theta)$. $h(\mathbb{S}_t)$ is the representation of state $\mathbb{S}$ in step t. To efficiently train our model, we perform batch processing as described in line 9.

Evaluation Phase

Algorithm 2 illustrates the evaluation phase of our proposed method. To get the best model parameter Θ^*, we evaluate the training result against a set of given graph G available snapshots. We will use this model parameter in the testing simulation of adaptive graph protection scheme.

Algorithm 2: Evaluation Phase: MVC Evaluation

```
Input: snapshots of graph $G$, number of training iteration
       $iter_t$
Output: neural network best parameter $\Theta^*$
Initialize $\Theta^* = 0$ and $|V_c| = 0$
for training $i = 1, ..., iter_t$ do
    Load model $i$, with parameter $\Theta_i$
    Get $V_c$ of $G$ using $\Theta_i$
    if $|V_c|_i < |V_c|$ then
        $\Theta^* = \Theta_i$
end
return $\Theta^*$
```

Algorithm 3: Testing Phase: Adaptive Graph Protection

```
Input: current snapshot of graph $G$, an integer $k$
Output: a set $S$ of $k$ nodes
Initialize $S$ to be empty, $S = \{\}$
Embed each node in $G$ into $d$-dimensional feature vector using
    Eq. 6 and 7
Get set of minimum vertex cover nodes $V_c^*$ (unordered) using
    the best model (with parameter $\Theta^*$)
Reordered set of $V_c^*$ minimum vertex cover nodes with their
    corresponding degree
Get $S$ from top $k$ nodes in $V_c^*$
return $S$
```

Testing Phase
Algorithm 3 shows the testing phase of adaptive graph protection scheme in dynamic networks. We are given an input snapshot of graph G and budget k.

D. Graph Embeddings as Feature-Based Representations

Let us explain our proposal of using graph embeddings as feature-based representation in our framework. Our main consideration as follows:

First, as we are handling the dynamic networks and different given input dataset, we have to deal with graphs of different size and structure. We consider the graph embedding approach as a fixed-length representation. Each node is represented in a feature-based d-dimensional vector. This representation enables us to process different graph size and structure, of which is the natural topologies of dynamic networks that change over time.

Second, in realistic situations of reinforcement learning, we may not possibly learn about every single state, especially if given a large data. There is an excessive number of states to visit in training and to hold the Q-Tables in memory. Instead, we aim to let our model generalize by learning from some small number of training states through experience. Thus, it can generalize that experience to new, similar situations. The graph embedding help us to efficiently train our reinforcement learning model.

We will leverage a neural network architecture over graphs, in particular, Structure2Vec [19], [20], to embed the network state. This graph embedding network will compute a d-dimensional feature embedding μ_v for each node $v \in V$, given the current partial solution $\mathbb{S}$.

To represent each node v, we construct a d-dimensional embedding μ_v. Given a graph $G = (V, E)$, we initialize $\mu_v^{(0)} = 0$, and for every $v \in V$ we update it iteratively in T iterations as follows:

$$\mu_v^{(t+1)} = \\ \text{ReLU }(\vartheta_1 x_v + \vartheta_2 \sum_{u \in N(v)} \mu_u^{(t)} + \vartheta_3 \sum_{u \in N(v)} \text{ReLU } (\vartheta_4 w(u,v))), \tag{6}$$

with x_v is node v own tag, whether being already selected or not not. Selected node will be given tag = 1. Otherwise 0. $\sum_{u \in N(v)} \mu_u^{(t)}$ is the feature of node v neighbors. $w(u,v)$ is the neighbors' edge weight, to consider the weighted connection in weighted graph. While $\vartheta_1, \vartheta_2, \vartheta_3$, and ϑ_4 are the model parameters and ReLU is the rectifier linear unit activation function.

Here we will explain how to get the evaluation function $\hat{Q}(h(S_t), v; \Theta)$ of training phase shown in Algorithm 1. Once the embedding μ_v for each node $v \in V$ is calculated using Eq.6 after T iteration, we get $\mu_v^{(T)}$. The pooled embedding of the entire graph G is then given by

$$\sum_{u \in V} \mu_u^{(T)} \tag{7}$$

Then we can use it to estimate the evaluation function as follows:

$$\hat{Q}(h(\mathbb{S}), v; \Theta) = \vartheta_5^\top \text{ReLU } (\text{concat } (\vartheta_6 \sum_{u \in V} \mu_u^{(T)}, \vartheta_7 \mu_v^{(T)})), \tag{8}$$

being $\sum_{u \in V} \mu_u^{(T)}$ is the pooled embedding of the entire graph. ϑ_5, ϑ_6, and ϑ_7 are the neural network model parameters.

The evaluation function $\hat{Q}(h(\mathbb{S}), v)$ depends on the collection of seven parameters $\Theta = \{\vartheta_i\}_{i=1}^7$ which is learned by training in Algorithm 1 and evaluated in Algorithm 2.

V. Experimental Results

A. Dataset

We evaluate our proposed methods on various real-world dynamic network datasets, which summarized in Table I.

TABLE I: Statistics of Dynamic Network Dataset

Name	#nodes	#edges	timespan	snapshot
Infectious [21]	410	17,298	1 hour	8
Hypertext 2009 [22]	113	20,818	12 hours	6
Hospital [23]	75	32,424	1 day	5
PrimarySchool [24]	242	125,773	1 hour	18
Email [25]	986	332,334	30 days	19

B. Comparison Methods

Recall that to the best of our knowledge, there is no previous work has been proposed to handle the adaptive scheme dynamic graph protection problem. Here we compare the performance of the following methods:

- *None*: simulates the condition without any protection.
- *Random*: gives protection to k uniformly random functional nodes.

- *GreedyMVC*: approximates the set of MVC nodes of the input graph by greedily selects the uncovered edge with the maximum sum of degrees of its endpoints [19]. Then protects k nodes from this unordered MVC set.
- *Degree*: selects k highest degree nodes of the current snapshot of the dynamic network at the protection turn.
- ***ReProtect***: our proposed method as described in Algorithm 1 and 2, trained on each available snapshot of dynamic networks. The name is abbreviated from the Reinforcement Learning-based Protection strategy.
- ***ReProtect-p***: our proposed method trained on the perturbed graph of each available snapshot of dynamic networks. The perturbation is performed by probabilistically removing edges from the snapshot graph. Specifically, for each edge, if the edge weight is smaller than the generated random number, the edge will be removed. We introduce this version to provide more training data variety.

C. Evaluation Criteria

We measure the protection effectiveness result using the surviving ratio (θ) of nodes in dynamic network G at the end of epidemics.

D. Experimental Setting

In the training phase, we use the embedding dimension size 64, batch size 64, embedding iteration 5, n_step 5, learning rate 0.0001 and number of training iteration 100,000. For the evaluation phase, we consider the number of evaluation iteration as 100.

For a fair comparison, all of the comparison methods are simulated under the same setting as follows: infection probability $\beta = 0.8$, recovery probability $\delta = 0.2$, and the initial number of attacked nodes = k. We employ the random attack strategies. Finally, all of the experiments are performed on the same machine, Ubuntu 16.06 LTS PC with an Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz CPU and NVIDIA GTX 1080 Ti SLI GPU.

E. Effectiveness Evaluation

On real-world networks, we compare the performance of all comparison methods on five different datasets. Table II shows the result of surviving nodes ration on SIS epidemic model. The results are averaged from 100 simulations under the constraint of budget $k = 0.15N$, with N is the number of nodes in the input graph. Both of our proposed methods consistently reach the highest ratio of surviving nodes. Additionally, the proposed methods with more training data variety using the perturbed graph, namely ReProtect-p achieves a better result than the regular training as in ReProtect.

To evaluate the performance comparison in different budget k, we vary the given k as shown in Figure 2. Both of our proposed methods are able to outperform other competitors align with the increasing given budget in all datasets, while constantly maintain competitive performance in a very small size of budget k. In PrimarySchool and Hypertext 2009 datasets, our proposed methods even can surpass the Degree method from earlier states, the small size of k.

VI. Conclusion

In this paper, we have addressed the adaptive graph protection problem on dynamic networks. We introduce the n-step fitted Q-learning to train our model under reinforcement approach. Using reinforcement learning approach, we introduce more stochasticity to the protection policy to explore more effective result. We further incorporate graph embedding as a feature-based representation of the network states. We demonstrate the potential of our proposals, namely ***ReProtect*** and ***ReProtect-p***, as non-deterministic approaches for the problem. Results of the experimental evaluation in real-world network datasets show that our proposed methods effectively restrain epidemic propagation.

VII. Acknowledgements

This work was supported by Tokyo Tech - Fuji Xerox Cooperative Research (Project Code KY260195), JSPS Grant-in-Aid for Scientific Research(B) (Grant Number 17H01785) and JST CREST (Grant Number JPMJCR1687). A.W.W. thanks to Indonesia Endowment Fund for Education (LPDP) for the educational scholarship.

References

[1] C. Bakker, M. Halappanavar, and A. Visweswara Sathanur, "Dynamic graphs, community detection, and riemannian geometry," *Applied Network Science*, vol. 3, no. 1, p. 3, Mar 2018.

[2] H. Zhuang, Y. Sun, J. Tang, J. Zhang, and X. Sun, "Influence maximization in dynamic social networks," in *2013 IEEE 13th International Conference on Data Mining*, Dec 2013, pp. 1313–1318.

[3] A. W. Wijayanto and T. Murata, "Pre-emptive spectral graph protection strategies on multiplex social networks," *Applied Network Science*, vol. 3, no. 1, p. 5, Apr 2018.

[4] A. W. Wijayanto and T. Murata, "Flow-aware vertex protection strategy on large social networks," in *The 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2017)*, ser. ASONAM 2017, Aug 2017, pp. 58–63.

[5] C. Chen, H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau, "Node immunization on large graphs: Theory and algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 113–126, Jan 2016.

[6] Y. Zhang, A. Ramanathan, A. Vullikanti, L. Pullum, and B. A. Prakash, "Data-driven immunization," in *2017 IEEE International Conference on Data Mining (ICDM)*, vol. 00, Nov. 2017, pp. 615–624.

[7] Y. Zhang and B. A. Prakash, "Data-aware vaccine allocation over large networks," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 2, pp. 20:1–20:32, Oct. 2015.

[8] C. Song, W. Hsu, and M. L. Lee, "Node immunization over infectious period," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '15. New York, NY, USA: ACM, 2015, pp. 831–840.

[9] C. Buono and L. A. Braunstein, "Immunization strategy for epidemic spreading on multilayer networks," *EPL (Europhysics Letters)*, vol. 109, no. 2, p. 26001, 2015.

[10] D. Zhao, L. Wang, S. Li, Z. Wang, L. Wang, and B. Gao, "Immunization of epidemics in multiplex networks," *PLOS ONE*, vol. 9, no. 11, pp. 1–5, 11 2014.

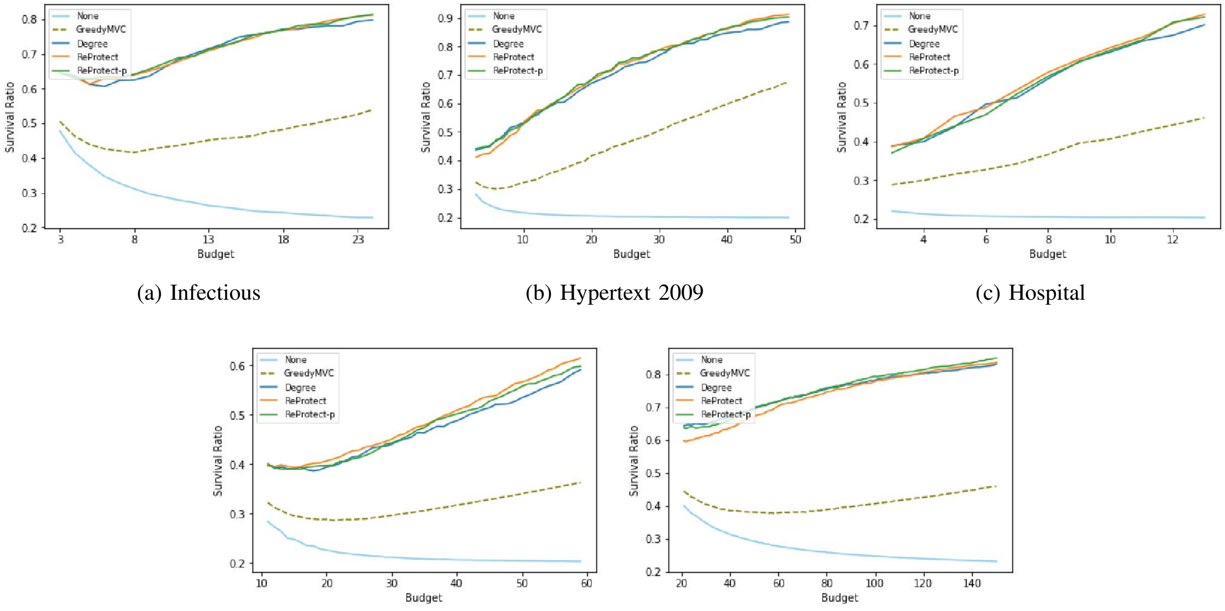
[11] Habiba, Y. Yu, T. Y. Berger-Wolf, and J. Saia, "Finding spread blockers in dynamic networks," in *Advances in Social Network Mining and Analysis*, L. Giles, M. Smith, J. Yen, and H. Zhang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 55–76.

[12] H. Tong, B. A. Prakash, C. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau, "On the vulnerability of large graphs," in *2010 IEEE International Conference on Data Mining*, Dec 2010, pp. 1091–1096.

[13] Y. Zhang and B. A. Prakash, "Dava: Distributing vaccines over networks under prior information," in *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 2014, pp. 46–54.

TABLE II: Ratio of surviving nodes on real-world network datasets

Dataset	None	Random	GreedyMVC	Degree	**ReProtect**	**ReProtect-p**
Infectious	0.2318	0.7528	0.5155	0.7793	0.7894	**0.8003**
Hypertext 2009	0.2063	0.6114	0.3819	0.6197	0.6377	**0.6449**
Hospital	0.2031	0.6508	0.4425	0.6725	0.7052	**0.7075**
PrimarySchool	0.2280	0.8362	0.3999	0.8382	0.8371	**0.8409**
Email	0.2310	0.7890	0.4579	0.8261	0.8335	**0.8469**



(a) Infectious

(b) Hypertext 2009

(c) Hospital

(d) PrimarySchool

(e) Email

Fig. 2: Effectiveness evaluation on different number of available budget (k). Both of our proposals (green and orange colored lines) outperform the other competitors. Higher is better.

[14] B. Wang, G. Chen, L. Fu, L. Song, and X. Wang, "Drimux : Dynamic rumor influence minimization with user experience in social networks," *IEEE Trans. on Knowl. and Data Eng.*, vol. 29, no. 10, pp. 2168–2181, 2017.

[15] J. Zhan, T. Rafalski, G. Stashkevich, and E. Verenich, "Vaccination allocation in large dynamic networks," *Journal of Big Data*, vol. 4, no. 1, p. 2, Dec 2017.

[16] B. A. Prakash, H. Tong, N. Valler, M. Faloutsos, and C. Faloutsos, "Virus propagation on time-varying networks: Theory and immunization algorithms," in *Machine Learning and Knowledge Discovery in Databases*, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 99–114.

[17] J. Liu and C. Gao, "Adaptive immunization in dynamic networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6804 LNAI, pp. 673–683, 2011.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, 2015.

[19] E. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6339–6349.

[20] H. Dai, B. Dai, and L. Song, "Discriminative embeddings of latent variable models for structured data," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2702–2711.

[21] L. Isella, J. Stehl, A. Barrat, C. Cattuto, J. Pinton, and W. Van den Broeck, "What's in a crowd? analysis of face-to-face behavioral networks," *Journal of Theoretical Biology*, vol. 271, no. 1, pp. 166–180, 2011.

[22] L. Isella, J. Stehla, A. Barrat, C. Cattuto, J.-F. Pinton, and W. V. den Broeck, "What's in a crowd? analysis of face-to-face behavioral networks," *J. of Theoretical Biology*, vol. 271, no. 1, pp. 166–180, 2011.

[23] P. Vanhems, A. Barrat, C. Cattuto, J.-F. Pinton, N. Khanafer, C. Regis, B.-a. Kim, B. Comte, and N. Voirin, "Estimating potential infection transmission routes in hospital wards using wearable proximity sensors," *PLoS ONE*, vol. 8, no. 9, p. e73970, 09 2013.

[24] V. Gemmetto, A. Barrat, and C. Cattuto, "Mitigation of infectious disease at school: targeted class closure vs school closure." *BMC infectious diseases*, vol. 14, no. 1, p. 695, Dec 2014.

[25] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '17. New York, NY, USA: ACM, 2017, pp. 601–610.