Data-Driven Immunization

Yao Zhang*, Arvind Ramanathan°, Anil Vullikanti* †, Laura Pullum°, and B. Aditya Prakash*
*Department of Computer Science, Virginia Tech
†Biocomplexity Institute, Virginia Tech
°Oak Ridge National Laboratory
Email: {yaozhang, badityap}@cs.vt.edu, vsakumar@vt.edu, {ramanathana, pullumll}@ornl.gov

***Abstract*—Given a contact network and coarse-grained diagnostic information like electronic Healthcare Reimbursement Claims (eHRC) data, can we develop efficient intervention policies to control an epidemic? Immunization is an important problem in multiple areas especially epidemiology and public health. However, most existing studies focus on developing pre-emptive strategies assuming prior epidemiological models. In practice, disease spread is usually complicated, hence assuming an underlying model may deviate from true spreading patterns, leading to possibly inaccurate interventions. Additionally, the abundance of health care surveillance data (like eHRC) makes it possible to study data-driven strategies without too many restrictive assumptions. Hence, such an approach can help public-health experts take more practical decisions.**

In this paper, we take into account propagation log and contact networks for controlling propagation. We formulate the novel and challenging *Data-Driven Immunization* problem without assuming classical epidemiological models. To solve it, we first propose an efficient sampling approach to align surveillance data with contact networks, then develop an efficient algorithm with the provably approximate guarantee for immunization. Finally, we show the effectiveness and scalability of our methods via extensive experiments on multiple datasets, and conduct case studies on nation-wide real medical surveillance data.

I. Introduction

Vaccination and social distancing are among the principle strategies for controlling the spread of infectious diseases [1], [2]. CDC (Centers for Disease Control) guidelines for vaccine usage are typically based on age groups, e.g., for young children and seniors—these do not result in optimal interventions, which minimize outcomes such as the total number of infections [1]. Additionally, most work on designing immunization algorithms from a data-mining viewpoint have focused on developing innovative strategies which assume knowledge of the underlying disease model [3], [4] or make assumptions of very fine-grained individual-level surveillance data [5].

Recent trends have led to the increasing availability of electronic claims data and also capabilities in developing very realistic urban population contact networks. This motivates the following problem: given a contact network, and a *coarse-grained* propagation log like electronic Health Reimbursement Claims (eHRC), can we learn an efficient and realistic intervention policy to control propagation (such as a flu outbreak)? Further, can we do it directly without assuming any epidemiological models? Influenza viruses change constantly, hence designing interventions optimized for specific epidemic model parameters is likely to be suboptimal [6].

The diagnostic propagation log data provides us with a good sense of how diseases spread, while contact networks tell us how people interact with others. We take into account both for immunization and study the *data-driven immunization* problem. Some of the major challenges include: (i) the scale of these datasets (eHRC consists of billions of records and contact networks have millions of nodes), and (ii) eHRC data is anonymized, and available only at a zip-code level. The main contributions of our paper are:

(a) Problem Formulation. We formulate the Data-Driven Immunization problem given a contact network and the propagation log. We first sample the most likely "social contact" cascades from the propagation log to the contact network. and then pose the immunization problem at a location level, and show it is NP-hard.

(b) Effective Algorithms. We present efficient algorithms to get the most-likely samples, and then provide a contribution-based greedy algorithm, ImmuConGreedy, with provably approximate solutions to allocate vaccines to locations.

(c) Experimental Evaluation. We present extensive experiments against several competitors, including graph-based and model-based baselines, and demonstrate that our algorithms outperforms baselines by reducing upto 45% of the infection with limited budget. Furthermore, we conduct case studies on nation-wide real medical surveillance data with billions of records to show the effectiveness of our methods. To the best of our knowledge, we are the first to study realistic immunization policies on such large-scale datasets.

Due to page restrictions, we omit most proofs giving sketches where possible.

II. Preliminaries

We give a brief introduction of the propagation data eHRC and contact networks we used in this section.

Propagation Data (eHRC). The propagation data for this study was primarily based on IMS Health claims data, *electronic Healthcare Reimbursement Claims* (eHRC), which consists of over a billion claims for the period of April 1st, 2009 - March 31st, 2010. The claims data consists of reimbursement claims recorded electronically from health care practitioners received from all parts of the US, including urban and rural areas. The dataset, its features, and its overall coverage/completeness are described in detail in [7], [8]; for this study, we used daily flu reports, based on ICD-9 codes 486XX and 488XX and individual locations (zip-code) recorded in the claims. Prior to our study, we obtained internal Institutional Review Board approval for analyzing the dataset.

Activity Based Populations. We use city-scale activity based populations as contact networks (see [9], [10] for more details). These models are constructed by a "first-principles" approach, and integrate over a dozen public and commercial datasets, including census, land use, activity surveys and transportation networks. The model includes detailed demographic attributes at an individual and household level, along with normative activities. These models have been used in a number of studies on epidemic spread and public health policy planning, including response strategies for smallpox attacks [10] and the National strategy for pandemic flu [2].

III. Problem Formulations

Table I lists the main notation used throughout the paper.

Table I
TERMS AND SYMBOLS

Symbol	Definition and Description
$G(V, E)$	graph G with the node set V and the edge set E
R	propagation log
$\mathbf{N}$	infection matrix for the propagation log R
$N(L_\ell, t_i)$	the number of patients at t_i in L_ℓ
t_0	the earliest timestep $t_0 = 0$
n	number of locations
$L = \{L_1, \ldots, L_n\}$	set of locations
m	number of vaccines
$\mathbf{x}$	vaccine allocation vector $[x_1, \ldots, x_n]'$
k	number of samples in $\mathcal{M}$
$\mathcal{M}$	set of sampled cascades $\{\mathbf{M}_1, \ldots, \mathbf{M}_k\}$
$\mathbf{M}$	a sampled cascade
$SI_{\mathbf{M}}$	the starting infected node set in $\mathbf{M}$
$\sigma_{G,\mathbf{M}}(\mathbf{x})$	the expected number of nodes $SI_{\mathbf{M}}$ can reach when $\mathbf{x}$ is given
$\rho_{G,\mathbf{M}_i}(\mathbf{x})$	$\sigma_{G,\mathbf{M}}(\mathbf{0}) - \sigma_{G,\mathbf{M}}(\mathbf{x})$
$\alpha_{\mathbf{M},\ell}$	number of nodes that have at least one parent in $\mathbf{M}$ at location L_ℓ
S_ℓ	the initial starting node set at location L_ℓ, where $\lvert S_\ell \rvert = N(L_\ell, t_0)$

We use $G(V, E)$ to denote an undirected unweighted graph and $L = \{L_1, ..., L_n\}$ to denote a set of locations. $V_i \subseteq V$ denotes the set of nodes at location L_i; we assume there are no overlapping nodes between locations. Large medical surveillance data, like `eHRC` is usually anonymous due to privacy issues. Hence, in this paper, we assume the number of infections are given. Formally, the propagation log R is an infection matrix $\mathbf{N}$ $((t_{max} + 1) \times n)$, where t_0 and t_{max} are the earliest and last timesteps. Each element $N(L_\ell, t)$ represents the number of patients in R at location L_ℓ at time t. Each row vector $\mathbf{N}(t) = [N(L_1, t), \ldots, N(L_n, t)]$ represents the number of infections at time t, and each column vector $\mathbf{N}_{L_\ell} = [N(L_\ell, t_0), \ldots, N(L_\ell, t_{max})]^T$ represents the number of infections at location L_ℓ.

Interactions and Surveillance. A contact network G models people's interactions with others, which is a powerful tool to control epidemics. For example, Prakash et al. [11] showed that the first eigenvalue of the adjacency matrix of G is related to the epidemic threshold. An epidemic will be quickly extinguished given a small epidemic threshold. Several effective algorithms have been proposed to minimize the first eigenvalue to control epidemics [3], [12], [4]. However, all of them assume an underlying epidemiological model like Susceptible-Infected-Recovered (SIR) [13]. In addition, they are strictly graph-based methods without looking into rich medical surveillance data. Though graph-based methods can provide us with good baseline strategies, they do not take into account particular patterns of a given virus. On the other hand, the disease propagation data R like `eHRC`, can give us a coarse-grained picture of infections. However, there is very little information on how an epidemic spreads via person-to-person contacts from R. Hence, we believe the disease propagation data R, along with a contact network G, can help us develop better and more implementable interventions to control an epidemic. For example, we can take the surveillance data of the past flu season to allocate vaccines for the current flu season.

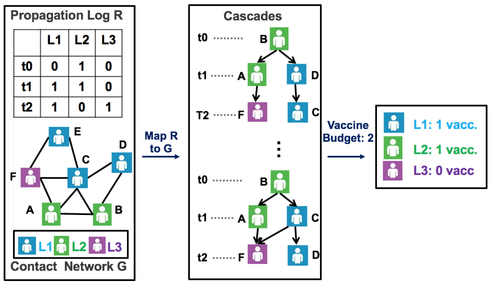


Figure 1. **Overview of our approach. We first generate a set of cascades, then allocate vaccine to different locations.**

Map R to nodes in G. The main challenge of integrating R and G is that R (like `eHRC`) in practice is anonymized. Hence we cannot associate each record in R with a node in G. In this paper, we tackle this challenge by mapping infections from R to nodes in G at the location level. The idea is that at each location L_ℓ and time t_i, we pick $N(L_\ell, t_i)$ nodes in G as infected nodes. Note that we can have multiple choices of mapping R to G. For example, in Figure 1, $N(L_2, t_0) = 1$, hence, we can pick either A or B as infected node at t_0. We denote these choices as $\mathcal{M}$, where $\mathcal{M}$ is a set of cascades. We define a *cascade* $\mathbf{M}$ as follows:

Definition 3.1: (*Cascade*). A cascade $\mathbf{M}$ is a directed acyclic graph (DAG) induced by R and G. Each node $u \in V_{\mathbf{M}}$ is associated with a location L_ℓ and a timestep t_i, where $u \in V_i$ and u is infected at t_i (denoted as $t(u) = t_i$). For node u and v in $\mathbf{M}$, if $e_{u,v} \in E$ and $t(u) = t(v) - 1$, there is a directed edge from u to v in $\mathbf{M}$. We denote $e(u, v) \in E_{\mathbf{M}}$.

We could select $N(L_\ell, t_i)$ nodes uniformly at random as infected nodes in G for each $\mathbf{M}$. However, it is not practical as infection distributions are not uniform. For example, if a node u has an infected neighbor, u can be infected by that node; in contrast, if u does not have any infected neighbor in R, it is unlikely to be infected. Hence, we propose to map R to G according to the SocialContact approach.

SocialContact. We say an infected node u gets infected by "social contact" in G, if u has a direct neighbor that is infected earlier than u. Otherwise, we call a node is infected by external forces. In reality, infectious diseases (like flu, mumps, etc.) usually spread via person-to-person contact. Hence, for a mapped cascade $\mathbf{M}$, we want to maximize the number of nodes caused by SocialContact. Formally, we define $\alpha_{\mathbf{M}} = |\{u|\exists v, e(v,u) \in E_{\mathbf{M}}\}|$, i.e., $\alpha_{\mathbf{M}}$ is the number of nodes that have at least one parent in $\mathbf{M}$. Then maximizing the number of nodes infected by SocialContact is equivalent to maximizing $\alpha_{\mathbf{M}}$. Figure 1 shows two cascades with the best $\alpha_{\mathbf{M}} = 4$: as only the node that starts the infection does not have a parent. To get k cascades with SocialContact in $\mathcal{M}$, we formulate the Mapping Problem:

Problem 3.1: (Mapping Problem). Given a contact network G, propagation log R, and number of cascades k, find $\mathcal{M}^* = \{\mathbf{M}_1^*, \ldots, \mathbf{M}_k^*\}$ where each node u in $\mathbf{M}$ is associated with a location L_ℓ and a time t_i:

$$\mathcal{M}^* = \arg\max_{\mathcal{M}} \sum_{\mathbf{M}_i \in \mathcal{M}} \alpha_{\mathbf{M}_i}, \text{ s.t. } |\mathcal{M}| = k \qquad (1)$$

Remark 3.1: Since we do not specify any epidemiological model (like SIR) for Problem 3.1, it is difficult to define any probability distribution for $\mathcal{M}$. Hence, the sample average approximation approach is not applicable for this problem.

Data-Driven Immunization. Once we generate $\mathcal{M}$, we want to study how to best allocate vaccines to minimize the infection shown in R. Recently, Zhang et al [4] proposed a model-based group immunization problem, in which they uniformly-at-random allocate vaccines to nodes *within* groups—this mimics real-life distribution of vaccines by public-health authorities. We leverage their within-group allocation approach. Let us define $\mathbf{x} = [x_1, \ldots, x_n]'$ as a vaccine allocation vector, where x_i is the number of vaccines given to location L_i. If we give x_i vaccines to location L_i, x_i nodes will be uniformly randomly removed from V_i. The objective is to find an allocation that "break" the cascades most effectively. We define $SI_{\mathbf{M}}$ as the starting 'seed' infected nodes in $\mathbf{M}$, i.e., $SI_{\mathbf{M}} = \{u \in V_{\mathbf{M}}|t_u = t_0\}$, and $\sigma_{G,\mathbf{M}}(\mathbf{x})$ as the expected number of nodes $SI_{\mathbf{M}}$ can reach after $\mathbf{x}$ is allocated to locations in $\mathbf{M}$. Hence, we want to minimize $\sigma_{G,\mathbf{M}}(\mathbf{x})$ to limit the expected infection over any cascade $\mathbf{M} \in \mathcal{M}$. For example, in Figure 1, once 2 vaccines are given to L_1 and L_2, we minimize the number of nodes that B can reach in the two cascades.

For ease of description, let us define $\rho_{G,\mathbf{M}}(\mathbf{x}) = \sigma_{G,\mathbf{M}}(\mathbf{0}) - \sigma_{G,\mathbf{M}}(\mathbf{x})$. $\rho_{G,\mathbf{M}}(\mathbf{x})$ can be thought as the number of nodes we can save if $\mathbf{x}$ is allocated. Since $\sigma_{G,\mathbf{M}_i}(\mathbf{0})$ is constant, minimizing $\sigma_{G,\mathbf{M}}(\mathbf{x})$ is equivalent to maximize $\rho_{G,\mathbf{M}}(\mathbf{x})$. Formally, our data-driven immunization problem *given* $\mathcal{M}$ (from Problem 3.1) is:

Problem 3.2: (Data-Driven Immunization). Given a contact network G, a set of cascades $\mathcal{M}$, and budget m, find a vaccine allocation vector $\mathbf{x}^*$:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \frac{1}{|\mathcal{M}|} \sum_{\mathbf{M}_i \in \mathcal{M}} \rho_{G,\mathbf{M}_i}(\mathbf{x}), \text{ s.t. } |\mathbf{x}|_1 = m \qquad (2)$$

Hardness. Both Problem 3.1 and Problem 3.2 are NP-hard, as they can be reduced from the Max-K-Set Union problem [14] and the DAV problem [5] respectively.

IV. Proposed Method

In this section, we develop two efficient algorithms, MappingGeneration for Problem 3.1, and ImmuConGreedy for Problem 3.2.

A. Generating Cascades from SocialContact

Main Idea: To tackle Problem 3.1, we first focus on a special case where $k = 1$ (find a single cascade $\mathbf{M}$), then extend it to multiple cascades. The challenge here is that even when $k = 1$, Problem 3.1 is still NP-hard. Our main idea to solve this is to first generate $SI_{\mathbf{M}}$ (the seed set), and then generate $\mathbf{M}$ from $SI_{\mathbf{M}}$. In principle, this can be done from checking $SI_{\mathbf{M}}$'s i-hop neighbors. Clearly, $SI_{\mathbf{M}}$'s quality will directly affect $\mathbf{M}$'s quality. However, it is still hard to find $SI_{\mathbf{M}}$ and generate $\mathbf{M}$ from $SI_{\mathbf{M}}$. Instead, we identify a necessary condition for the optimal $\mathbf{M}$, and propose a provable approximation algorithm to find $SI_{\mathbf{M}}$ that satisfies the condition. We make the algorithm faster by leveraging the Approximate Neighborhood Function (ANF) technique. Then we generate the corresponding cascade $\mathbf{M}$ from $SI_{\mathbf{M}}$, and propose a fast algorithm MappingGeneration to extend it to k cascades for Problem 3.1.

Finding $SI_{\mathbf{M}}$. To find a high quality $SI_{\mathbf{M}}$, we first examine what is the optimal $\mathbf{M}$. According to Eqn. 1, the optimal $\mathbf{M}$ has the maximum value of $\alpha_{\mathbf{M}}$. Let us define $\alpha^*_{\mathbf{M}}$ as the maximum of $\alpha_{\mathbf{M}}$ ($\alpha_{\mathbf{M}} \le \alpha^*_{\mathbf{M}}$). Then we have the following lemma:

Lemma 4.1: $\alpha^*_{\mathbf{M}} = \sum_{t=t_1}^{t_{max}} |\mathbf{N}(t)|_1$, i.e., the number of infections after the earliest time t_0.

Proof: (Sketch) When we map R to G, the *optimal* case for a cascade $\mathbf{M}$ is that every node u with $t(u) > t_0$ has at least one parent in $\mathbf{M}$, and the only nodes that do not have any parents are the ones infected at the earliest time t_0. Hence, $\alpha^*_{\mathbf{M}}$ is the number of nodes that are infected after t_0. ■

Now we know the maximum $\alpha_{\mathbf{M}}$. However, it is hard to find a $SI_{\mathbf{M}}$ with the optimal $\mathbf{M}$ as shown in the next lemma.

Lemma 4.2: Find a set $SI_{\mathbf{M}}$ for the cascade $\mathbf{M}$ with $\alpha_{\mathbf{M}} = \alpha^*_{\mathbf{M}}$ is NP-hard.

According to Lemma 4.2, it is intractable to examine the whole graph to get $SI_{\mathbf{M}}$ for large networks (like `Houston` with 59 million edges in Section V). Hence, instead we will look at each location independently to find $SI_{\mathbf{M}}$, and aggregate the result to generate $\mathbf{M}$.

Let us define $\alpha_{\mathbf{M},\ell}$ as the number of nodes that have at least one parent in $\mathbf{M}$ at location L_ℓ. Similarly to $\alpha_{\mathbf{M}}$, we have $\alpha_{\mathbf{M},\ell} \le \alpha^*_{\mathbf{M},\ell}$ where $\alpha^*_{\mathbf{M},\ell} = \sum_{i=1}^{t_{max}} N(L_\ell, t_i)$. $\alpha^*_{\mathbf{M},\ell}$ is the number of patients after t_0 at location L_ℓ in R, and it is the optimal value for $\alpha_{\mathbf{M},\ell}$. Since we want to find a set of starting nodes, here we define S_ℓ as a node set at location L_ℓ: i.e., $S_\ell = \{v|v \in S \text{ and } v \in V_\ell\}$ where $|S_\ell| = N(L_\ell, t_0)$. For each location L_ℓ, we want to find a set S_ℓ as the starting infected node set, such that S_ℓ will yield a cascade $\mathbf{M}$ that minimizes $\alpha_{\mathbf{M},\ell}$. Our idea is to find S_ℓ that satisfies a necessary condition for the best $\alpha_{\mathbf{M},\ell}$. We denote $CF(S_\ell, t_i) = |\{u|u \in V_l, \exists v \in S_\ell, dist(v,u) \le i\}|$, i.e., the number of nodes that S_ℓ can reach within distance i (i-hops) in

L_ℓ in G. Similarly, we denote $CN(L_\ell, t_i) = \sum_{k=0}^{i} N(L_\ell, t_i)$ (the cumulative number of infections in L_ℓ in R until time t_i). The next lemma will show that for each location L_ℓ, when $\alpha_{\mathbf{M},\ell} = \alpha^*_{\mathbf{M},\ell}$, the constraint in Eqn. 3 must be satisfied.

Lemma 4.3: (Necessary Condition) Given a cascade $\mathbf{M}$ generating from S_ℓ, if $\alpha_{\mathbf{M},\ell} = \alpha^*_{\mathbf{M},\ell}$, then for any timestep $t_i \in [0, t_{max}]$ and all locations L_ℓ, we have

$$CF(S_\ell, t_i) \geq CN(L_\ell, t_i) \quad (3)$$

Proof: (Sketch). If $\alpha_{\mathbf{M},\ell} = \alpha^*_{\mathbf{M},\ell}$, every node that is infected after t_0 has a parent. For any node u that is infected at t_i, u must be within the i-th hops of S_ℓ, which means the number of nodes within the i-hops of S_ℓ is greater than the number of nodes infected at t_i, i.e., $CF(S_\ell, t_i) \geq CN(L_\ell, t_i)$. ■

Lemma 4.3 demonstrates a necessary condition (Eqn. 3) for the maximum $\alpha_{\mathbf{M},\ell}$. Hence, we seek to develop an efficient algorithm that can produce accurate results for the necessary condition. Our idea is to construct a new objective function, which can get the necessary condition for the best $\mathbf{M}$ at location L_ℓ. To do so, we propose the following problem to find $SI_{\mathbf{M}}$:

Problem 4.1: Given graph G and infection matrix $\mathbf{N}$. We want to find $S^* = \{S^*_1, \ldots, S^*_n\}$ s.t., $|S^*_\ell| = N(L_\ell, t_0)$ for any location L_ℓ, such that $S^*_\ell = \arg\min_{S_\ell} \theta(S_\ell)$, $\forall$ location L_ℓ, where $\theta(S_\ell) = \sum_{i=0}^{t_{max}} \mathbb{1}_{CF(S_\ell,t_i)<CN(L_\ell,t_i)}(CN(L_\ell, t_i) - CF(S_\ell, t_i))$.

Here $\mathbb{1}_{CF(S_\ell,t_i)<CN(L_\ell,t_i)}$ is an indicator function: if $CF(S_\ell, t_i) < CN(L_\ell, t_i)$ then it is 1, otherwise 0.

Justification of Problem 4.1. Recall that $\alpha^*_{\mathbf{M},\ell}$ is the optimal value for $\alpha_{\mathbf{M},\ell}$, and $\theta(S_\ell)$ is non-negative. We have the following lemma:

Lemma 4.4: If $\alpha_{\mathbf{M},\ell}$ is optimal, then $\theta(S_\ell) = 0$.

Lemma 4.4 shows that if we minimize $\theta(S_\ell)$, we are able to get the necessary condition for the best $\mathbf{M}$ at location L_ℓ. Therefore, we propose Problem 4.1 to get $SI_{\mathbf{M}}$.

Hardness. Problem 4.1 is NP-hard, as it can be reduced from the set cover problem [14].

Solving Problem 4.1. Let us define $g(S_\ell) = [\sum_{i=0}^{t_{max}} CN(L_\ell, t_i)] - \theta(S_\ell)$. $\sum_{i=0}^{t_{max}} CN(L_\ell, t_i)$ is constant, so minimizing $\theta(S_\ell)$ is equivalent to maximize $g(S_\ell)$.

Lemma 4.5: $g(S_\ell)$ has the following properties: $g(\emptyset) = 0$; it is monotonic increasing and submodular.

Proof: (Sketch). We first show that $CF(S_\ell, t_i)$ is monotone non-decreasing and submodular functions, then extend it to $g(S_\ell)$. Please see details in the appendix. ■

Lemma 4.5 suggests a natural greedy algorithm to solve Problem 4.1. We call it SAMPLENAIVEGREEDY. Each time it picks a node u^* such that $u^* = \arg\max_{u \in V_\ell} \; g(S_\ell \cup \{u\}) - g(S_\ell)$ until $N(L_\ell, t_0)$ nodes have been selected to S_ℓ. We do it for all locations to get $SI_{\mathbf{M}}$.

Lemma 4.6: For each location L_ℓ, SAMPLENAIVEGREEDY gives a $(1 - 1/e)$-approximate solution to $g(S_\ell)$.

SAMPLENAIVEGREEDY selects a node with the maximum marginal gain of $g(S_\ell)$ iteratively. It takes $O(|V|(|V| + |E|))$ time if we run BFS to get each $CF(S_\ell, t_i)$ for each iteration. The time complexity to get all $|\mathbf{N}(t_0)|_1$ nodes as $SI_{\mathbf{M}}$ is $O(|\mathbf{N}(t_0)|_1|V|(|V| + |E|))$, which is not scalable to large networks. Hence, we need a faster algorithm.

Speeding up SAMPLENAIVEGREEDY. In SAMPLENAIVEGREEDY, each time we recompute $CF(S_\ell \cup \{u\}, t_i)$ for all i, which takes $O(|E| + |V|)$ time. We can speed up this computation by leveraging the ANF (Approximate Neighborhood Function) algorithm [15], which uses a classical probabilistic counting algorithm, the Flajolet-Martin algorithm [16] to approximate the sizes of union-ed node sets using bit strings. Here, we refer to the bit string that approximates $CF(S_\ell, t_i)$ as $\mathbb{F}(S_\ell, i)$. To estimate $CF(S_\ell \cup \{u\}, t_i)$, we first do a bitwise-OR operation: $\mathbb{F}(S_\ell \cup \{u\}, i) = [\mathbb{F}(S_\ell, i) \texttt{ OR } \mathbb{F}(\{u\}, i)]$, then convert it to $CF(S_\ell \cup \{u\}, t_i)$. According to the ANF algorithm, $CF(\cdot, t_i) = \phi(\mathbb{F}(\cdot)) = (2^b)/.77351$, where b is the average position of the leftmost zero bit of the bit string. Since the bitwise-OR operation takes constant time, we can reduce the running time of $CF(S_\ell \cup \{u\}, t_i)$ for all timesteps i from $O(|E| + |V|)$ to $O(t_{max})$.

We propose SAMPLEGREEDY (Algorithm 1), a modified greedy algorithm with bitwise-OR operations for Problem 4.1. It first gets $\mathbb{F}(\{u\}, i)$ for all nodes at location L_ℓ over all timesteps using ANF [15] (Line 2), then follows SAMPLENAIVEGREEDY. However, we use bitwise-OR operations to speed up the computation of $CF(S_\ell \cup \{u\}, t_i)$ (Line 7-8).

Algorithm 1 SAMPLEGREEDY

```
Require: graph $G$, and propagation log matrix $\mathbf{N}$.
 1: for each location $L_\ell$ do
 2:     Get $\mathbb{F}(\{u\}, i)$ for all timestep $i$, all $u \in V_\ell$ using ANF [15]
 3:     $y = N(L_\ell, t_0)$
 4:     $S_\ell = \emptyset$, and $\mathbb{F}(S_\ell, i) = 0$ for all timesteps $i$
 5:     for $i = 1$ to $y$ do
 6:         for each node $u \in V_\ell - S_\ell$ do
 7:             $\mathbb{F}(S_\ell \cup \{u\}, i) = \mathbb{F}(S_\ell, i)$ OR $\mathbb{F}(\{u\}, i)$ for all $t_i$
 8:             $CF(S_\ell \cup \{u\}, t_i) = \phi(\mathbb{F}(S_\ell \cup \{u\}, i))$ for all $t_i$
 9:         end for
10:         $u^* = \arg\max_{u \in V_\ell - S_\ell} g(S_\ell) - g(S_\ell \cup \{u\})$
11:         $S_\ell = S_\ell \cup u^*$
12:     end for
13: end for
14: return $SI_{\mathbf{M}} = \{S_1, \ldots, S_n\}$
```

Lemma 4.7: SAMPLEGREEDY takes $O((|V||\mathbf{N}(t_0)|_1 + |E|)t_{max})$ time.

Generating cascades from $SI_{\mathbf{M}}$. Once we obtain $SI_{\mathbf{M}}$ from Algorithm 1, we can generate $\mathbf{M}$ from $SI_{\mathbf{M}}$. Similar to the result of Lemma 4.2, generating $\mathbf{M}$ from $SI_{\mathbf{M}}$ is also hard. Here we propose a heuristic, the CASCADEGENERATION algorithm (Algorithm 2) for $\mathbf{M}$. Let us define $D^\ell_i = \{u | u \in V_\ell, \exists v \in SI_{\mathbf{M}}, dist(v, u) = i\}$, i.e, a set of nodes in location L_ℓ that $SI_{\mathbf{M}}$ can reach at distance i. We first add $SI_{\mathbf{M}}$ to the cascade $\mathbf{M}$, and compute D^ℓ_i for all time t_i and location L_ℓ by running a BFS starting from $SI_{\mathbf{M}}$ (Line 2). Then we select nodes into $\mathbf{M}$ by running another BFS from $SI_{\mathbf{M}}$ as well: at each distance i from $SI_{\mathbf{M}}$, for each location L_ℓ we pick $N(L_\ell, t_i)$ nodes uniformly at random to $\mathbf{M}$, and add corresponding edges (Line 4-18). Note that we do it by permutating the set D^ℓ_i. $N(L_\ell, t_i)$ nodes are selected as follows: (1) if $|\text{CANDIDATEQUEUE}_l| \geq N(L_\ell, t_i)$ (the constraint in Eqn. 3 follows), we uniformly at random pick $N(L_\ell, t_i)$ to $\mathbf{M}$ from CANDIDATEQUEUE (Line 8-10); (2) otherwise, we add all nodes in CANDIDATEQUEUE to $\mathbf{M}$, record the number of nodes left (Line 11-12), and finally randomly pick other nodes in V_ℓ to $\mathbf{M}$ (Line 18).

Algorithm 2 CASCADEGENERATION

```
Require: Graph $G$, propagation log matrix $\mathbf{N}$, and node set $SI_{\mathbf{M}}$
 1: Add all nodes in $SI_{\mathbf{M}}$ to the cascade $\mathbf{M}$
 2: Compute $D_i^\ell$ for all time $t_i$ (by running BFS from $SI_{\mathbf{M}}$)
 3: PRESET = $SI_{\mathbf{M}}$, NUMLEFTNODE=0
 4: for $i = 1$ to $t_{max}$ do
 5:     for each location $L_\ell$ do
 6:         $\hat{D}_i^\ell = \texttt{Permutate}(D_i^\ell)$
 7:         Add $\hat{D}_i^\ell$ to the end of CANDIDATEQUEUE$_\ell$
 8:         if $|\text{CANDIDATEQUEUE}_\ell| \geq N(L_\ell, t_i)$ then
 9:             CURSET=pop $N(L_\ell, t_i)$ nodes from the top of CANDIDATEQUEUE$_\ell$
10:         else
11:             CURSET=pop all nodes in CANDIDATEQUEUE$_\ell$
12:             NUMLEFTNODE+=($N(L_\ell, t_i) - |\text{CANDIDATEQUEUE}|_\ell$)
13:         end if
14:         Add CURSET to $\mathbf{M}$, and edges from PRESET to CURSET
            if $e(u, v) \in G$ for any $u \in$ PRESET and $v \in$ CURSET
15:     end for
16:     PRESET=CURSET
17: end for
18: Uniformly randomly pick NUMLEFTNODE nodes from $V_\ell$ to $\mathbf{M}$
19: return $\mathbf{M}$
```

Lemma 4.8: CASCADEGENERATION takes $O(|V| + |E|)$ time.

Extend CASCADEGENERATION to k cascades. We can simply extend Algorithm 2 to k cascades. Note that CASCADEGENERATION permutates the nodes in D_i^ℓ (Line 6), hence, for different permutations, we can generate different cascades. If the constraint in Eqn. 3 holds, at time t_i, we uniformly at random add $N(L_\ell, t_i)$ into $\mathbf{M}$ from $\sum_{j=1}^{i} |D_i^\ell| - \sum_{j=1}^{i-1} N(L_\ell, t_j)$ candidate nodes. If the constraint does not follow, we uniformly at random pick extra nodes from $V - V_{\mathbf{M}}$ to $\mathbf{M}$.

Remark 4.1: The above random process will generate $O(\prod_{L_\ell \in L} \prod_i |D_i^\ell|)$ cascades.

Remark 4.1 shows that we have a large number of cascades. In case if we need more, we can generate extra cascades by ranking the result of SAMPLEGREEDY: instead of picking the best S_ℓ, we pick the top sets (in Algorithm 1 Line 10-11). In practice, as shown in our experiments, we do not need to do this, as we have enough cascades. In addition, our cascades have high quality: the average value of $\alpha_{\mathbf{M}}$ is almost the same as the optimal solution (Table III).

MAPPINGGENERATION. Combining the above results, we propose the MAPPINGGENERATION algorithm (Algorithm 3) to solve Problem 3.1.

Claim 4.1: The time complexity of MAPPINGGENERATION (Algorithm 3) is $O((|V||\mathbf{N}(t_0)|_1 + |E|)t_{max} + \hat{k}(|V| + |E|))$, where $\hat{k}$ is the number of runs for CASCADEGENERATION to get k cascades.

Algorithm 3 MAPPINGGENERATION

```
Require: graph $G$, propagation log $R$
 1: Generate propagation log matrix $\mathbf{N}$
 2: Run SAMPLEGREEDY ($G, \mathbf{N}$) (Algorithm 1) to get $SI_{\mathbf{M}}$
 3: RunCASCADEGENERATION ($G, \mathbf{N}, SI_{\mathbf{M}}$) (Algorithm 2) until $k$
    unique cascades are found for $\mathcal{M}$
 4: return $\mathcal{M}$.
```

B. Data-Driven Immunization

Main Idea: In this section, we solve the Data-Driven Immunization (Problem 3.2) assuming the samples are available. We first show that $\rho_{G,\mathbf{M}_i}(\mathbf{x})$ in Problem 3.2 is neither submodular nor supermodular. We then propose to optimize an alternative credit-based objective function, which is an upperbound of $\rho_{G,\mathbf{M}_i}(\mathbf{x})$ (Problem 4.2). We show that this function is non-negative, increasing and has the diminishing return property. Based on these properties, we propose a greedy algorithm which gives a $(1 - 1/e)$-approximate solution.

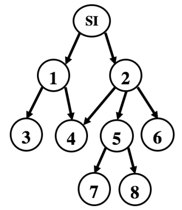


Figure 2. **Counter-Example**

Note that in Problem 3.2, $\rho_{G,\mathbf{M}_i}(\mathbf{x})$ is defined over an integer lattice, and is not a simple set function. If a function $h(\mathbf{x})$ has the diminishing return property over an integer lattice, then for any $\mathbf{x}' \geq \mathbf{x}$ and k, we have $h(\mathbf{x} + \mathbf{e}_k) - h(\mathbf{x}) \geq h(\mathbf{x}' + \mathbf{e}_k) - h(\mathbf{x}')$ ($\mathbf{e}_k$ be the vector with 1 at the kth index). According to [4], there exists a near-optimal algorithm to maximize $h(\mathbf{x})$. Unfortunately, $\rho_{G,\mathbf{M}_i}(\mathbf{x})$ does not follow the diminishing return property.

Remark 4.2: $\rho_G(\mathbf{x}, \mathbf{M}_i)$ does not have diminishing return property. Figure 2 shows a counter-example, where all nodes are in different locations. Suppose $\mathbf{x} = \mathbf{0}$, $\mathbf{x}' = \mathbf{e}_1$, then $\mathbf{x} \leq \mathbf{x}'$, however, $\rho_{G,\mathbf{M}_i}(\mathbf{x} + \mathbf{e}_2) - \rho_{G,\mathbf{M}_i}(\mathbf{x}) = 5$ and $\rho_{G,\mathbf{M}_i}(\mathbf{x}' + \mathbf{e}_2) - \rho_{G,\mathbf{M}_i}(\mathbf{x}') = 8 - 2 = 6$.

Instead, we develop a *contribution* based approach. The idea is if we remove a node u in $\mathbf{M}_i$, the number of nodes u can save is related to u's children. Each child of u can contribute to the savings of removing u. First, let us denote $IN_{\mathbf{M}_i}(S)$ as the set of S's parents in $\mathbf{M}_i$, i.e., $IN_{\mathbf{M}_i}(S) = \{u | e(u, v) \in \mathbf{M}_i, v \in S\}$, and $OUT_{\mathbf{M}_i}(S)$ as the set of S's children in $\mathbf{M}_i$. We define the contribution $C_{G,\mathbf{M}_i}(S)$ recursively,

$$C_{G,\mathbf{M}_i}(S) = |S| + \sum_{v \in OUT_{\mathbf{M}_i}(S)} \frac{|IN_{\mathbf{M}_i}(\{v\}) \cap S|}{|IN_{\mathbf{M}_i}(\{v\})|} C_{G,\mathbf{M}_i}(\{v\}).$$

$\frac{|IN_{\mathbf{M}_i}(\{v\}) \cap S|}{|IN_{\mathbf{M}_i}(\{v\})|}$ is the fraction of savings v contributes to S. The intuition is that since we do not have any propagation models, it is reasonable to assume the infected v should be infected by any of its parents equally, hence v contributes its savings *equally* to each of its parents. Now we define the contribution function over an integer lattice,

$$\zeta_{G,\mathbf{M}_i}(\mathbf{x}) = \sum_S \Pr(S) C_{G,\mathbf{M}_i}(S), \tag{4}$$

where S is a node set sampled from the random process of distributing $\mathbf{x}$ ($|S| = |\mathbf{x}|_1$). Lemma 4.9 will show that $\zeta_{G,\mathbf{M}_i}(\mathbf{x})$ is the upperbound of $\rho_{G,\mathbf{M}_i}(\mathbf{x})$, and it is also lowerbounded by expected number of nodes S can reach.

Lemma 4.9: Given a cascade $\mathbf{M}_i$, $\rho_{G,\mathbf{M}_i}(\mathbf{x}) \leq \zeta_{G,\mathbf{M}_i}(\mathbf{x})$.

We use $\zeta_{G,\mathbf{M}_i}(\mathbf{x})$ to estimate $\rho_{G,\mathbf{M}_i}(\mathbf{x})$. Hence, we formally define the following problem for Problem 3.2.

Problem 4.2: Given a contact network $G(V, E)$, a set of cascades $\mathcal{M}$, and budget m, find a vaccine allocation vector $\mathbf{x}^*$:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \frac{1}{|\mathcal{M}|} \sum_{\mathbf{M}_i \in \mathcal{M}} \zeta_{G,\mathbf{M}_i}(\mathbf{x}), \text{ s.t.} |\mathbf{x}|_1 = m. \quad (5)$$

Lemma 4.10: $\zeta_{G,\mathbf{M}_i}(\mathbf{x})$ has the following properties:

(P_1) $\zeta_{G,\mathbf{M}_i}(\mathbf{x}) \geq 0$ and $\zeta_{G,\mathbf{M}_i}(\mathbf{0}) = 0$.

(P_2) (Nondecreasing) $\zeta_{G,\mathbf{M}_i}(\mathbf{x}) \leq \zeta_{G,\mathbf{M}_i}(\mathbf{x} + \mathbf{e}_i)$ for i.

(P_3) (Diminishing returns) For any $\mathbf{x}' \geq \mathbf{x}$, we have $\zeta_{G,\mathbf{M}_i}(\mathbf{x} + \mathbf{e}_i) - \zeta_{G,\mathbf{M}_i}(\mathbf{x}) \geq \zeta_{G,\mathbf{M}_i}(\mathbf{x}' + \mathbf{e}_i) - \zeta_{G,\mathbf{M}_i}(\mathbf{x}')$.

Given the properties of $\zeta_{G,\mathbf{M}_i}(\mathbf{x})$ in Lemma 4.10, we propose a greedy algorithm, IMMUNAIVEGREEDY for Problem 4.2: each time we give one vaccine to location L_{ℓ^*}, such that

$$\ell^* = \arg\max_{L_\ell} \sum_{\mathbf{M}_i \in \mathcal{M}} \zeta_{G,\mathbf{M}_i}(\mathbf{x} + \mathbf{e}_\ell) - \zeta_{G,\mathbf{M}_i}(\mathbf{x}),$$

until m vaccines are allocated.

Lemma 4.11: IMMUNAIVEGREEDY gives a $(1 - 1/e)$-approximate solution to Problem 4.2.

In IMMUNAIVEGREEDY, since we uniformly randomly distribute vaccines, we can apply the Sample Average Approximation (SAA) framework, i.e., $\zeta_{G,\mathbf{M}_i}(\mathbf{x}) \approx \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} C_{G,\mathbf{M}_i}(S)$, where $\mathcal{S}$ is a set of samples taken from the vaccine allocation process. This approach takes $O(|\mathcal{S}|(|V| + |E|))$ to estimate $\zeta_{G,\mathbf{M}_i}(\mathbf{x})$, and we need to look into $|\mathcal{M}|$ cascades to pick the best location L_{ℓ^*} for one iteration. We have $|L|$ locations and m vaccines. Hence, the total time complexity of IMMUNAIVEGREEDY is $O(m|L||\mathcal{M}||\mathcal{S}|(|V| + |E|))$, which is not practical for large networks. However, we can speed up this naive greedy algorithm.

Speeding up IMMUNAIVEGREEDY. We propose a faster algorithm, IMMUCONGREEDY (*Contribution-based Greedy Immunization*) in Algorithm 4, which takes only $O(m|\mathcal{M}|(|V| + |E|))$ time. The idea is that we can compute the contribution function efficiently when the budget $m = 1$, i.e., all values of $\zeta_{G,\mathbf{M}_i}(\mathbf{e}_\ell)$ in $\mathbf{M}_i$ can be obtained in $O(|V| + |E|)$ time. This is because $\zeta_{G,\mathbf{M}_i}(\mathbf{e}_\ell) = \sum_{u \in V_\ell} \frac{1}{|L_\ell|} C_{G,\mathbf{M}_i}(\{u\})$, and we can get $C_{G,\mathbf{M}_i}(\{u\})$ for all $u \in V$ by traversing $\mathbf{M}_i$ once. For simplicity, let $d_{in}(v) = |IN_{\mathbf{M}_i}(\{v\})|$. We have $C_{G,\mathbf{M}_i}(\{u\}) = 1 + \sum_{v \in OUT_{\mathbf{M}_i}(\{u\})} \frac{1}{d_{in}(v)} C_{G,\mathbf{M}_i}(\{v\})$. If u does not have any children ($OUT_{\mathbf{M}_i}(\{u\}) = \emptyset$), $C_{G,\mathbf{M}_i}(\{u\}) = 1$. Since $\mathbf{M}_i$ is a DAG, we can iteratively obtain $C_{G,\mathbf{M}_i}(\{u\})$ for all $u \in V$ from a reversed order of a topological sort, which takes $O(|V| + |E|)$ time.

In Algorithm 4, we compute contribution function $C_{G,\mathbf{M}_i}(\{u\})$ for all $\mathbf{M}_i$ (Line 4), which takes $O(|\mathcal{M}|(|V| + |E|))$ time. Then we obtain $\sum_{\mathbf{M}_i \in \mathcal{M}} \zeta_{G,\mathbf{M}_i}(\mathbf{e}_\ell)$ for each location L_ℓ by summing up the contribution for each $u \in V_\ell$ (Line 5), which takes $O(|\mathcal{M}||V|)$ time. Once we allocate one vaccine to the best location L_{ℓ^*}, we update each $\mathbf{M}_i$ by uniformly at random removing one node in L_{ℓ^*} (Line 7). This way we can just compute $\sum_{\mathbf{M}_i \in \mathcal{M}} \zeta_{G,\mathbf{M}_i}(\mathbf{e}_\ell)$ instead of $\sum_{\mathbf{M}_i \in \mathcal{M}} \zeta_{G,\mathbf{M}_i}(\mathbf{x} + \mathbf{e}_\ell)$ after the next iteration.

Algorithm 4 IMMUCONGREEDY

```
Require: graph $G(V, E)$, propagation log $R$, and budget $m$
 1: $\mathcal{M}$ =MAPPINGGENERATION $(G, R)$ {Section IV-A}
 2: $\mathbf{x} = 0$
 3: for $j = 1$ to $m$ do
 4:     $\forall \mathbf{M}_i \in \mathcal{M}$: compute $C_{G,\mathbf{M}_i}(\{u\})$ for each node $u$
 5:     $\forall$ location $L_\ell \in L$: compute $\sum_{\mathbf{M}_i \in \mathcal{M}} \zeta_{G,\mathbf{M}_i}(\mathbf{e}_\ell)$
 6:     $\ell^* = \arg\max_{L_\ell} \sum_{\mathbf{M}_i \in \mathcal{M}} \zeta_{G,\mathbf{M}_i}(\mathbf{e}_\ell)$
 7:     $\forall \mathbf{M}_i \in \mathcal{M}$: update $\mathbf{M}_i$ by uniformly at random removing one
        node at location $L_{\ell^*}$
 8:     $\mathbf{x} = \mathbf{x} + \mathbf{e}_{\ell^*}$
 9: end for
10: return $\mathbf{x}$
```

Lemma 4.12: IMMUCONGREEDY takes $O(m|\mathcal{M}|(|V| + |E|))$ time.

V. EXPERIMENTS

We conducted the experiments using a 4 Xeon E7-4850 CPU with 512GB of 1066Mhz main memory[1].

A. Experimental Setup

Networks. We do experiments on multiple datasets (Table II). Stochastic Block Model (SBM) [17] is a well-known graph model to generate synthetic graphs with groups. WorkPlace and HighSchool are social contact networks[2]. Nodes in HighSchool are students from 5 different sections and edges represent two students who are in vicinity of each other. Nodes in WorkPlace are employees of a company with 5 departments and edges indicate two people are in proximity of each other. We treat each section/department as a location. Miami and Houston are million-node social-contact graphs from city-scale activity based synthetic populations as described in Section II. We divided people by their residential zipcodes.

Table II
NETWORK DATASETS

Dataset	Nodes	Edges	Locations
WorkPlace	92	757	5
HighSchool	182	2221	5
SBM	1000	5000	20
Miami	2.2 **million**	50 **million**	74
Houston	2.7 **million**	59 **million**	98

Propagation logs. We have the billion-record eHRC data (described in Section II) as the propagation log R for Miami and Houston. The Miami and Houston have $118K$ and $132K$ patients respectively For SBM, HighSchool, and WorkPlace, we run the well-known SIR model (infection

[1] Code in Python: http://people.cs.vt.edu/yaozhang/data-immu/.

[2] http://www.sociopatterns.org

rate as 0.4, and recovery rate as 0.6) to generate the propagation log R: we first uniformly at random pick 5% nodes at each location as seeds at t_0, then run a SIR simulation to get other infected nodes.

Settings. We set the number of samples $|\mathcal{M}| = 1000$ for MappingGeneration, and number of bitmasks as 32 for computing $\mathbb{F}(\cdot)$ in SampleGreedy (similar to the ANF algorithm [15]).

Baselines. As we are not aware of any direct competitor tackling our problem, we use several baselines to better judge our performance. These baselines have been regularly used for immunization studies. However, none of them take into account both propagation log and contact networks.

(1) Random: uniformly randomly assign vaccines to locations.

(2) PropPopulation: a data based approach: assign vaccines to locations in proportion to population in locations.

(3) PropInfection: a data based approach: assign vaccines in proportion to total number of infections in locations.

(4) Degree: a graph based approach: calculate the average degree d_{L_i} of each location L_i, and independently assign vaccines to L_i with probability $d_{L_i}/\sum_{L_k \in L} d_{L_k}$.

(5) ImmuModel: a model based approach: apply the *model-driven group immunization* algorithm (the QP version) in [4]. ImmuModel aims to minimize the spectral radius of a contact graph. Spectral radius is the first eigenvalue of the graph, which has been proven to be the threshold of an epidemic in the graph [11]. We set edge weights to be 0.24 according to [8].

B. Results

In short, we demonstrate that our immunization algorithm ImmuConGreedy outperforms other baselines on all datasets. We also show our approach is robust as the size of the propagation log R varies. In addition, we show that our sampling algorithm SampleGreedy provides accurate results for generating cascade samples. Finally, we study the scalability of our approach.

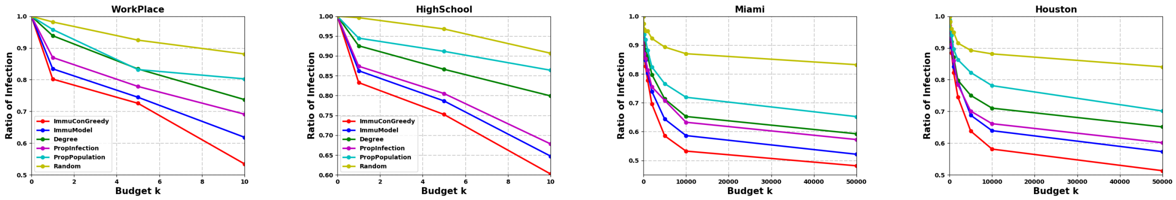
Effectiveness of ImmuConGreedy. Figure 3 shows results of minimizing the spread on cascades for the whole log R. In all datasets, ImmuConGreedy consistently outperforms others. WorkPlace and HighSchool have < 200 nodes, hence we varied m till 10. However, even with the small budget 10, ImmuConGreedy can reduce 45% of the infection, which is about 10% better than the second best ImmuModel. For Miami and Houston with upto $2.7 million$ nodes, ImmuConGreedy can reduce about 50% of the infection on the cascades generated by SocialContact with only $50K$ vaccines. Model-based ImmuModel and data-based PropInfection perform better than Random and Degree as they take into account either epidemic threshold in the contact graph or the eHRC data. However, ImmuConGreedy easily outperforms them, as it leverages both contact networks and the eHRC data.

We also study how to leverage the rich log data to develop vaccine interventions in the future. To do so, we split the eHRC data into training parts and testing parts: we get the vaccine allocations from the training parts (the fall regime of flu from Aug 2009 - Oct 2009), and apply the allocations to the testing parts (the winter regime of flu from Nov 2009 - Feb 2010) to examine how effective our approach ImmuConGreedy is. Figure 4 shows the results of infection reductions on the cascades generating from the testing data. ImmuConGreedy consistently outperforms others in both Miami and Houston: it can reduce about 25% of the infection with only $5K$ vaccines, compared to other baselines like ImmuModel and PropInfection.

We use simulations of the SIR model to evaluate the performance of ImmuConGreedy on the activity based urban social contact networks (described in Section II). These were first calibrated to get the same outbreak size as in the eHRC data for these cities. We then choose a random subset of individuals in each zipcode to be vaccinated, based on the allocation by ImmuConGreedy. We find the reduction in the number of infections is quite substantial in many cases. For instance, for Miami, for a budget of $50K$ vaccines, the ImmuConGreedy allocation leads to more than 50% reduction, compared to a random allocation.

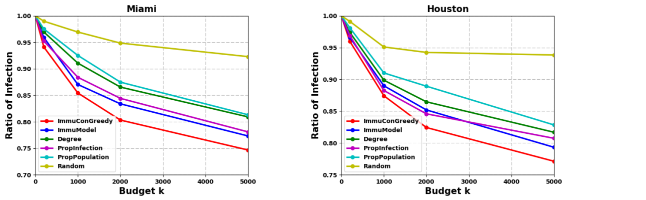
Robustness of ImmuConGreedy. We study how sensitive ImmuConGreedy is, as the size of the propagation log R varies next. To do so, we first generate synthetic propagation log R from the SIR model, then manually change the size of R as the input of our data. Finally, we compare ImmuConGreedy to the model based approach ImmuModel. For each dataset, we generate R by running a SIR simulation (with the infection rate 0.4 and the recovery rate 0.6 for WorkPlace, HighSchool and SBM, and the infection rate 0.24 and timesteps to recovery 7 for Miami according to [8]). Once R is generated, we change the size of R by extracting a portion $[\mathbf{N}(t_0), \ldots, \mathbf{N}(t_{max})]$ as the input (p% of R). For example, suppose $t_{max} = 20$ and $p = 50$, we use $[\mathbf{N}(t_0), \ldots, \mathbf{N}(t_{10})]$ as the propagation log. Since we know all configurations come from the SIR model, we expect the model-based approach ImmuModel to do better than ImmuConGreedy. However, as p increases, as more data is used, ImmuConGreedy should approach ImmuModel. Figure 5 shows the results: as expected, for all datasets, clearly as p increases, ImmuConGreedy becomes better. Interestingly for smaller datasets like WorkPlace, HighSchool, SBM, even with only 25% of data, we can get upto 85% of the performance. For large networks like Miami, we need more data: however, when all the data is used, compared to ImmuModel, ImmuConGreedy can achieve 90% of the savings.

Effectiveness of MappingGeneration. We also study the performance of MappingGeneration by comparing $\alpha_{\mathbf{M}}$ to the optimal value α^* (Problem 3.1). We obtain α^* using the brute-force algorithm. See Table III: $\hat{\alpha}_{\mathcal{M}}$, the average value of $\alpha_{\mathbf{M}}$ over all sampled cascades, is almost the same as α^* for all datasets. For example, in SBM, $\hat{\alpha}_{\mathcal{M}}$ is 107.9, a difference of only 1.1 from α^*. In addition, we found that α^* is exactly the same as the number of nodes that are infected



(a) WorkPlace (b) HighSchool (c) Miami (d) Houston

Figure 3. **Effectiveness of ImmuConGreedy on the whole R. Infection ratio r vs. Vaccine budget m. Infection ratio $r = \frac{\sum_{\mathbf{M}_i \in \mathcal{M}} \sigma_{G,\mathbf{M}_i}(\mathbf{x})}{\sum_{\mathbf{M}_i \in \mathcal{M}} \sigma_{G,\mathbf{M}_i}(\mathbf{0})}$. Lower is better. ImmuConGreedy consistently outperforms other baselines over all datasets.**



(a) Miami (b) Houston

Figure 4. **Effectiveness of ImmuConGreedy for the testing data. Infection ratio r vs. Vaccine budget m. Lower is better. ImmuConGreedy consistently outperforms other baselines for both Miami and Houston.**

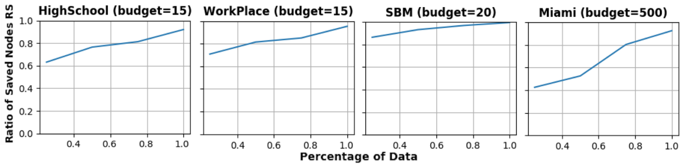


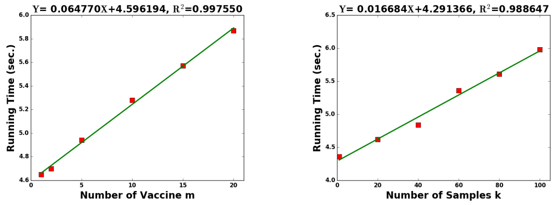
Figure 5. **Robustness of ImmuConGreedy as data size varies. Ratio of saved nodes RS vs. percentage of used log data $p\%$. $RS = \frac{S_{\text{Data}}}{S_{\text{Model}}}$. S_{Data} (S_{Model}): the number of nodes we can save when vaccines are allocated according to ImmuConGreedy (ImmuModel). Percentage of used log data p: $[\mathbf{N}(t_0), \ldots, p\%\mathbf{N}(t_{max})]$. Higher: ImmuConGreedy is closer to ImmuModel.**

after the first timestep t_0, which suggests the best scenario for SocialContact is that only nodes which are infected at the earliest time are not caused by social contact.

Scalability. Figure 6 shows the running time of MappingGeneration and ImmuConGreedy w.r.t. the vaccine budget m and the number of cascades k on SBM. For Figure 6(a) we set $k = 100$, while for Figure 6(b) we set $m = 20$. We observe that as m increases and k increases, the running time scales linearly (figures also show the linear-fit with R^2 values). Consistent with the time complexity bounds for our algorithms in Section IV, large datasets need fairly extensive time. For example, Miami takes about 2 days to get $5K$ vaccines. This is still reasonable: importantly, note that we expect to run immunization algorithms for infectious epidemics, so the solution quality is much more critical than the fastest running time.

Table III
MappingGeneration. $\hat{\alpha}_{\mathcal{M}}$: average of $\alpha_{\mathbf{M}}$ over all $\mathbf{M} \in \mathcal{M}$; α^*: optimal value of $\alpha_{\mathbf{M}}$; $N = \sum_{t=t_1}^{t_{max}} |\mathbf{N}(t)|_1$.

Dataset	$\hat{\alpha}_{\mathcal{M}}$	α^*	N
WorkPlace	79.2	83.0	83
HighSchool	165.2	170.0	170
SBM	107.9	109.0	109



(a) Varying m (b) Varying k

Figure 6. **Scalability. (a) total running time of MappingGeneration and ImmuConGreedy vs. vaccine budget m; (b) total running time of MappingGeneration and ImmuConGreedy vs. number of cascade samples k.**

C. Case Studies

We conduct case studies to analyze vaccine allocations per zipcode for both Houston and Miami. Figure 7 shows the total population, the total #patients in the eHRC data, the total #vaccines taken in the eHRC data[3], the total #vaccines from ImmuModel, and the total #vaccines from ImmuConGreedy, respectively.

Figure 7(a), (b), (c), (d) and (e) show the case study for Houston. First, the areas with zipcode 77030 and 77024 in Figure 7(b) have the largest number of patients, and vaccine allocations from both eHRC (Figure 7 (c)), and ImmuConGreedy (Figure 7 (e)) also prefer these areas. Second, vaccines taken in the eHRC data do not follow the total population (Figure 7(a)), but roughly follow the distribution of #patients in eHRC. This may suggest the immunization strategy in practice is to give vaccines based on the proportion of reported patients. Third, ImmuModel distributes 38% of vaccines to three areas (77002, 77008 and 77056), which are the center of Houston Metropolitan Area (like downtown and uptown) with a large number of interactions in the contact network. However, both data-based and model-based approaches do not perform well (see Figure 3). Our method, ImmuConGreedy, gives 43% of vaccines to the areas 77030, 77024 and 77002. The first two areas have the highest infections in eHRC, while the last one is essential for minimizing the epidemic threshold as ImmuModel suggests. Hence, ImmuConGreedy considers both eHRC and contact networks. It is interesting that the Texas Medical Center (one of the largest medical centers in the world) is in 77030, and Houston downtown is in 77002. Hence, ImmuConGreedy targets regions with high risk of influenza outbreak.

[3] We extract vaccine reports based on ICD-9 codes V04.81. These are actual vaccine allocations as given in the eHRC data.

Figure 7(f), (g), (h), (i) and (j) show the case study for Miami. First, vaccines taken in eHRC (Figure 7(h)) follow the distribution of #patients as well (Figure 7(g)). Second, IMMUMODEL distributes 31% of vaccines in one area with zipcode 33165 (Figure 7(i)). We believe this area with large number of households, is critical to minimize the spectral radius of the contact network in Miami. However, both data-based and model-based approaches do not perform well in Miami as well (as shown in Figure 3). Interestingly, as shown in Figure 7(j), our approach, IMMUCONGREEDY, gives most of the vaccines (29%, 18%) to areas with the largest number of patients (33140 and 33176 respectively). We observe that difference from Houston, in Miami IMMUCONGREEDY tend to prefer data-based approaches. However, the areas adjacent to 33165, which IMMUMODEL targets, also get higher vaccine allocations than others—this means IMMUCONGREEDY also takes into account information in the contact network. In fact, the areas IMMUCONGREEDY targets indeed have high risk of an influenza outbreak: they are either tourist attractions (33140) or residential areas (33176). For example, 33140 belongs to Miami Beach, which is a famous place with large transient population.

VI. RELATED WORK

We review closely related work next. Remotely related work includes those on blogs and propagations [18], and viral marketing [19] (e.g. Goyal et al. [20] studied the influence maximization problem using a data-based approach).

Epidemiology. The early canonical textbooks and surveys include [13], [21], which describe the fundamental epidemiological models like the so-called SIS and SIR models. Epidemic thresholds (minimum virulence of a virus that causes an epidemic) for various models have been extensively studied [22], [11]. In practice, viruses are always changing, and hence assuming a prior model may be suboptimal.

Immunization. There has been a lot of work on developing optimal strategies to control propagation over graphs. Cohen et al [23] proposed the popular *acquaintance* immunization policy, while Aspnes et al. [24] developed inoculation policies for victims of viruses using game theory. Tong et al. [3], [12], Van Mieghem et al. [25], and Prakash et al. [26] studied the problem of minimizing the spectral radius (epidemic threshold) of the graph for a variety of models. In addition, other immunization work in the literature has been proposed based on differential equation methods [1], [27]. The most related work includes Zhang et al. [4] who studied the immunization at the group scale, while Zhang et al. [5] and Khalil et al. [28] developed several model based efficient algorithms for immunization given partial information of infections. All past work proposed either model-based or graph-based approaches for immunization. Instead we leverage rich surveillance health care data together with the network information for the problem of controlling disease spread.

eHRC. Previous studies have pointed to the utility of eHRC data to identify trends in epidemic incidence across the US [29], [30]. Leveraging eHRC, the spatial and temporal patterns of flu incidence during 2009-2010 pandemic flu season have been discovered [7]. In addition, Malhotra et al. used sequential pattern mining techniques to reveal common sequences of clinical procedures administered to patients for a variety of medical conditions from eHRC [31]. In sum, none studied the immunization problem with the eHRC data.

VII. CONCLUSIONS

This paper addresses the novel problem of controlling epidemics in presence of coarse-grained health surveillance data and population contact networks. We formulate the Data-Driven Immunization problem, which first aims to align the propagation log with contact networks, and then allocate vaccines to minimize spread in the data. We develop an efficient approach MAPPINGGENERATION to obtain high quality cascades, and then give an approximation algorithm IMMUCONGREEDY with provable solutions for immunization on sampled cascades. We demonstrate the effectiveness of our method through extensive experiments on multiple datasets including nation-wide real electronic Health Reimbursement Claims data. Finally, case studies in Miami and Houston metropolitan regions show that our allocation strategies take both the network and surveillance data into account to effectively distribute vaccines.

Future work can include investigating other sampling strategies, incorporating more data sources, and studying vaccine allocations to other groups, such as demographics like age.

Acknowledgements This paper is based on work partially supported by the NSF (IIS-1353346), the NEH (HG-229283-15), ORNL (Order 4000143330) and from the Maryland Procurement Office (H98230-14-C-0127), and a Facebook faculty gift to BAP. AV is partially supported by the following grants: DTRA CNIMS Contract HDTRA1- 11-D-0016-0010, NSF BIG DATA Grant IIS-1633028 and NSF DIBBS Grant ACI-1443054, NSF EAGER SSDIM-1745207. Publication of this article was also funded by ORNL LDRD funding to AR. Oak Ridge National Laboratory (ORNL) is operated by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725. The US Government retains and the publisher, by accepting the article for publication, acknowledges that the US Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

REFERENCES

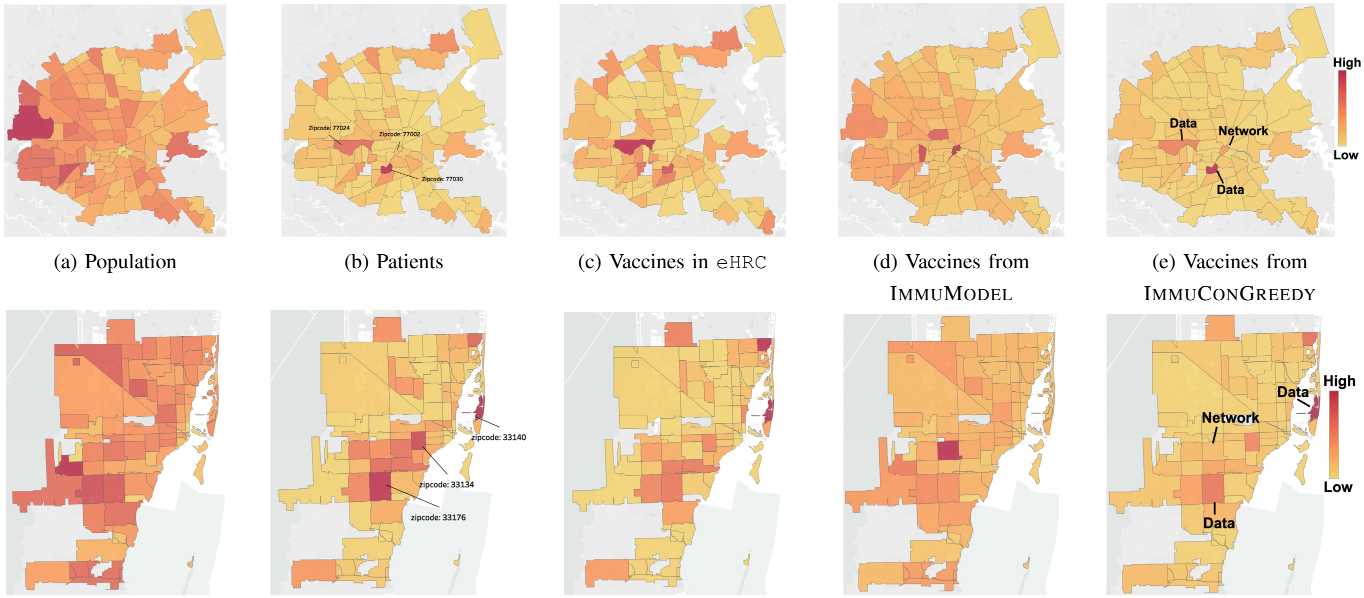
[1] J. Medlock and A. P. Galvani, "Optimizing influenza vaccine distribution," *Science*, vol. 325, 2009.

[2] M. E. Halloran, N. M. Ferguson, S. Eubank, I. M. Longini, D. A. T. Cummings, B. Lewis, S. Xu, C. Fraser, A. Vullikanti, T. C. Germann, D. Wagener, R. Beckman, K. Kadau, C. Barrett, C. A. Macken, D. S. Burke, and P. Cooley, "Modeling targeted layered containment of an influenza pandemic in the United States," in *Proceedings of the National Academy of Sciences (PNAS)*, March 10 2008, pp. 4639–4644.

[3] H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau, "On the vulnerability of large graphs," in *ICDM*, 2010.

[4] Y. Zhang, A. Adiga, A. Vullikanti, and B. A. Prakash, "Controlling propagation at group scale on networks," in *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 619–628.

[5] Y. Zhang and B. A. Prakash, "Dava: Distributing vaccines over networks under prior information," in *Proceedings of the SIAM Data Mining Conference*, ser. SDM '14, 2014.



(a) Population (b) Patients (c) Vaccines in `eHRC` (d) Vaccines from ImmuModel (e) Vaccines from ImmuConGreedy

(f) Population (g) Patients (h) Vaccines in `eHRC` (i) Vaccines from ImmuModel (j) Vaccines from ImmuConGreedy

Figure 7. **Case Studies for `Houston` and `Miami` per location. `Houston`: (a), (b), (c), (d) and (e); `Miami`: (f), (g), (h), (i) and (j). Heatmap of (a) and (f): Total population; (b) and (g): Patients in `eHRC`; (c) and (h): Number of vaccines actually taken in `eHRC`; (d) and (i): Vaccine allocations from ImmuModel; (e) and (j): Vaccine allocations from ImmuConGreedy.**

[6] L. Pellis, F. Ball, S. Bansal, K. Eames, T. House, V. Isham, and P. Trapman, "Eight challenges for network epidemic models," *Epidemics*, pp. 58–62, 2015.

[7] A. Ramanathan, L. L. Pullum, T. C. Hobson, C. A. Steed, S. P. Quinn, C. S. Chennubhotla, and S. Valkova, "Orbit: Oak ridge biosurveillance toolkit for public health dynamics," *BMC bioinformatics*, vol. 16, no. 17, p. S4, 2015.

[8] O. Ozmen, L. L. Pullum, A. Ramanathan, and J. J. Nutaro, "Augmenting epidemiological models with point-of-care diagnostics data," *PLOS ONE*, vol. 11, no. 4, pp. 1–13, 04 2016.

[9] C. L. Barrett, R. J. Beckman, M. Khan, V. S. Anil Kumar, M. V. Marathe, P. E. Stretz, T. Dutta, and B. Lewis, "Generation and analysis of large synthetic social contact networks," in *Winter Simulation Conference*. Winter Simulation Conference, 2009, pp. 1003–1014.

[10] S. Eubank, H. Guclu, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, "Modelling disease outbreaks in realistic urban social networks," *Nature*, vol. 429, no. 6988, pp. 180–184, May 2004.

[11] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos, "Threshold conditions for arbitrary cascade models on arbitrary networks," *Knowledge and Information Systems*, 2012.

[12] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, "Gelling, and melting, large graphs by edge manipulation," in *Proc. of CIKM*, 2012.

[13] R. M. Anderson and R. M. May, *Infectious Diseases of Humans*. Oxford University Press, 1991.

[14] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[15] C. R. Palmer, P. B. Gibbons, and C. Faloutsos, "Anf: A fast and scalable tool for data mining in massive graphs," ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 81–90.

[16] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," *Journal of computer and system sciences*, vol. 31, no. 2, pp. 182–209, 1985.

[17] A. F. McDaid, B. Murphy, N. Friel, and N. Hurley, "Clustering in networks with the collapsed stochastic block model," *Arxiv preprint arXiv:1203.3083*, Mar. 2012.

[18] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "On the bursty evolution of blogspace," in *WWW '03*, 2003, pp. 568–576.

[19] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *KDD '03*, 2003.

[20] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "A data-based approach to social influence maximization," *Proceedings of the VLDB Endowment*, vol. 5, no. 1, pp. 73–84, 2011.

[21] H. W. Hethcote, "The mathematics of infectious diseases," *SIAM Review*, vol. 42, 2000.

[22] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. of INFOCOM*, 2005.

[23] R. Cohen, S. Havlin, and D. ben Avraham, "Efficient immunization strategies for computer networks and populations," *Physical Review Letters*, vol. 91, no. 24, Dec. 2003.

[24] J. Aspnes, K. Chang, and A. Yampolskiy, "Inoculation strategies for victims of viruses and the sum-of-squares partition problem," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, ser. SODA '05, 2005, pp. 43–52.

[25] P. Van Mieghem, D. Stevanović, F. Kuipers, C. Li, R. Van De Bovenkamp, D. Liu, and H. Wang, "Decreasing the spectral radius of a graph by link removals," *Physical Review E*, vol. 84, no. 1, p. 016101, 2011.

[26] B. A. Prakash, L. A. Adamic, T. J. Iwashyna, H. Tong, and C. Faloutsos, "Fractional immunization in networks," in *Proc. of SDM*, 2013, pp. 659–667.

[27] E. Shim, "Optimal strategies of social distancing and vaccination against seasonal influenza," *Mathematical biosciences and engineering*, vol. 10(5), 2013.

[28] E. B. Khalil, B. Dilkina, and L. Song, "Scalable diffusion-aware optimization of network topology," in *KDD 2014*. ACM, 2014, pp. 1226–1235.

[29] A. Patwardhan and R. Bilkovski, "Comparison: Flu prescription sales data from a retail pharmacy in the us with google flu trends and us ilinet (cdc) data as flu activity indicator," *PloS one*, vol. 7, no. 8, p. e43611, 2012.

[30] J. R. Gog, S. Ballesteros, C. Viboud, L. Simonsen, O. N. Bjornstad, J. Shaman, D. L. Chao, F. Khan, and B. T. Grenfell, "Spatial transmission of 2009 pandemic influenza in the us," *PLoS Comput Biol*, vol. 10, no. 6, p. e1003635, 2014.

[31] K. Malhotra, T. C. Hobson, S. Valkova, L. L. Pullum, and A. Ramanathan, "Sequential pattern mining of electronic healthcare reimbursement claims: Experiences and challenges in uncovering how patients are treated by physicians," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2670–2679.

[32] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functionsi," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.