Influence Maximization through Scheduled Seeding in a Real-World Setting

Tomer Lev[1], Irad Ben-Gal[1], Erez Shmueli[1,*]

Abstract—In this paper, we evaluate, for the first time, the potential of a scheduled seeding strategy for influence maximization in a real-world setting. We first propose methods for analyzing historical data to quantify the infection probability of a node with a given set of properties in a given time, and assess the potential of a given seeding strategy to infect nodes. Then, we examine the potential of a scheduled seeding strategy by analyzing a real-world large-scale dataset containing both the network topology as well as the nodes' infection times. Specifically, we use the proposed methods to demonstrate the existence of two important effects in our dataset: a *complex contagion* effect and a *diminishing social influence* effect. As shown in a recent study, the scheduled seeding approach is expected to benefit greatly from the existence of these two effects. Finally, we compare a number of benchmark seeding strategies to a scheduled seeding strategy that ranks nodes based on a combination of the number of infectious friends they have, as well as the time that has passed since they became infectious. Results of our analyses show that for a seeding budget of 1%, the scheduled seeding strategy yields a convergence rate that is 14% better than a seeding strategy based solely on their degrees, and 215% better than a random seeding strategy, which is often used in practice.

Index Terms—Influence Maximization; Social Networks Analysis; Scheduled Seeding

1 Introduction

Social networks offer a powerful tool for information sharing with friends, family, and colleagues. Along with their role as a major social communication channel in our lives, social networks redesigned the way individuals consume content, acquire their personal preferences, and make decisions.

One of the main mechanisms that empowers social networks is social influence [2], [36], [21], [49]. This mechanism enables individuals to diffuse their messages in the social network passively through a viral process that resembles a virus's spread. A highly studied problem that arises in this context is finding influential nodes, that if seeded (i.e., infected intentionally), may further infect a large fraction of the network through the viral contagion process. This problem is commonly referred to as the influence maximization problem.

While many solutions were suggested for the influence maximization problem (e.g., [10], [46], [48]), most of these solutions focus on selecting the set of nodes to be seeded at the initial phase of the contagion process. As such, they are often based on examining static properties of nodes, such as properties derived from the network topology.

Several recent studies [44], [47], [24], [33], [45], [26], [23], [39], [12], [35], [8], [25], [38] suggested using a scheduled/adaptive/sequential seeding approach to find not only the best set of nodes to be seeded but also the right timing to seed them. The key idea behind these methods is that by spreading the seeding budget over time, it becomes possible to consider the dynamic states of nodes in addition to their static topological properties, when choosing the nodes to be seeded. This typically leads to a substantial improvement in the overall number of infections.

To the best of our knowledge, all existing scheduled seeding works thus far, tested the potential of scheduled seeding in theory and/or by simulations. While most of these studies did examine real-world network topologies, they assumed a particular underlying mathematical contagion model, and simulated the infection process based on this model. Here, we examine for the first time, the potential of scheduled seeding, by analyzing a unique real-world dataset, provided to us by a relatively large mobile network operator. This dataset contains both the call network between a sampled set of customers (which can be used to infer the social network topology), as well as the time in which each customer joined a specific service offered by the mobile network operator (which represents the infection times).

For that purpose, we first propose a method to quantify the infection probability of a node with a given set of properties in a given timestamp, using historical data. While infection probability of nodes can be calculated in a relatively straightforward manner when node properties remain static over time, here we are interested in properties that can change over time, making such a calculation more challenging. Then, we use the proposed method to demonstrate the existence of two important effects in our dataset: a *complex contagion* effect and a *diminishing social influence* effect. Put together, these effects imply that a node has a higher probability to become infected if it has a relatively high number of infectious neighbors, and if the time passed since its neighbors became infected is relatively short. As shown in a recent study by Goldenberg et al. [16], when these two effects co-exists, the scheduled seeding approach is expected to perform considerably better than the traditional initial seeding approach.

[1] *Department of Industrial Engineering at Tel-Aviv University, Tel-Aviv, Israel.*
* *Corresponding author, shmueli@tau.ac.il.*

We then suggest a method to assess the potential of a given seeding strategy to infect nodes using historical data. The suggested method is then applied to compare between a number of benchmark seeding strategies and a scheduled seeding strategy that ranks nodes based on the number of infectious friends they have, as well as the time that has passed since they became infectious. The score of nodes used for ranking in this case are taken as their infection probabilities, obtained by applying the method described above. Results of our analyses show that for a seeding budget of 1% (i.e., seeding 1% of the examined population), the scheduled seeding strategy, yields a convergence rate that is 14% better than a seeding strategy based solely on their degrees, and 215% better than a random seeding strategy, which is often used in practice.

The contribution of this paper can be summarized along two axes:

- Methodologically: we propose methods for analyzing historical data to: 1) quantify the infection probability of a node with a given set of properties in a given time, and 2) assess the potential of a given seeding strategy to infect nodes.
- Empirically: for the first time, we examine the potential of a scheduled seeding strategy by analyzing a real-world dataset. Specifically, we demonstrate the existence of both a *complex contagion* effect and a *diminishing social influence* effect, and that the scheduled seeding strategy considerably outperforms all other benchmark seeding strategies.

The remainder of this paper is organized as follows. In section 2, we present the relevant background and related work for this paper. In section 3, we describe the proposed methods, including toy examples to demonstrate their use. In section 4, we describe the unique dataset used in our analyses. Section 5 reports the results of our analyses. Section 6 summarizes this paper and suggests directions for future research.

2 Background and Related Work

In this section, we provide the relevant background to our work.

2.1 Contagion Models

Over the years, with the emergence of globally infectious diseases, mathematical contagion models were proposed in the literature to understand the disease dynamics and predict the possible outcomes of future outbreaks [1]. Due to the great success of these models in the field of disease modeling, their usage was extended to other fields, such as information diffusion and product adoption.

Existing contagion models can be broadly classified into two categories: (1) compartmental models and (2) individual-based models.

The primary assumption underlying compartmental models is that the population is fully interconnected, where any two individuals can interact and potentially infect one another. Perhaps the most-studied compartmental contagion model is the SIR model [30]. In this model, the entire population is divided into three compartments: S - Susceptible, I - Infected, and R - Recovered. Transitions between compartments can occur in one of the following manners: (1) susceptible individuals may change their state to infected with probability β, and (2) infected individuals may change their state to recovered with a constant pace γ.

Another well-studied compartmental contagion model is the Bass diffusion model [34], which was proposed to describe the process of adopting new products. This model introduced the innovation factor, which represents individuals' ability to adopt the product regardless of their infected neighbors. Therefore, according to this model, the population can be divided into two groups: (1) innovators who adopt the product at early stages, and (2) imitators who adopt the product after interacting with adopters.

As mentioned above, the second type of contagion models is individual-based models. The main assumption behind these models is that the population is not fully connected, and that different individuals are limited in terms of other individuals that they may interact with or infect. These potential interactions are typically described using a network structure where individuals are represented as nodes, and an interaction between two individuals is represented as an edge.

The Linear Threshold model is one of the most studied individual-based models in the context of information diffusion [21], [29]. This model's main assumption is that an individual's behavior depends to a great extant on the number of neighbors that share this behavior. Therefore, if the number (or weighted sum) of infected neighbors of node v at timestamp t is greater than a predefined threshold of v (θv), at the next timestamp v will become infected and begin to infect its own neighbors.

The Independent Cascade model is another fundamental individual-based model [17], [18]. This model assumes that any infected node has a single attempt to infect its uninfected neighbors at the next timestamp after its infection. In any future timestamps, this node will not be able to further infect its neighbors.

It is important to emphasize that many other models were suggested in the literature. Most of them are merely extensions of the basic models reviewed in this subsection, tailored to more specific settings.

2.2 The Influence Maximization Problem

One of the most important problems in the information diffusion field is the search for important or influential nodes in the social network. These nodes may have the potential to considerably enhance a viral process in the network, and therefore it is important to identify such nodes. Formally, the influence maximization problem can be defined as selecting a subset of nodes from a given network G, whose seeding (intentional activation) will start a viral contagion process, which is likely to result in activation of a significant number of nodes in the network. This problem may have different objective functions. For example, to maximize the number of activated nodes in a given time frame, or with a given seeding budget, or to minimize the number of seeding attempts needed to obtain a specified number of activated nodes.

For example, modern marketing efforts use social networks for market analysis and for defining promotion

strategies. Unlike classical mass-marketing methods that address a wide market segment, social networks' promotion is often characterized by micro-segmentation, attempting to utilize detailed information about each of the involved individuals [19]. The main motivation behind such an approach, is that influencing the opinion of only a few individuals may shape the opinion of the majority, by following a viral contagion process [28].

The task of identifying influential nodes is still widely investigated, but the identification of influential nodes is not always easy. In many cases, nodes are referred to as "influential" when past evidence show that their involvement in the contagion process contributes significantly to the spread. Nonetheless, in most real-world cases, this type of information is missing, and most of the data available to the marketers is the topological structure of the social network and past adoption history.

2.3 Traditional Seeding Strategies

Kempe et al. [29] studied the influence maximization problem under the Linear Threshold and Independent Cascade models and their generalizations. They prove that finding the optimal solution to the problem is NP-hard in both settings. Consequently, they presented a greedy algorithm that obtains a $(1 - 1/e)$ approximation of the optimal solution. While the greedy algorithm ensures a reasonably good result in terms of coverage, it is still very expensive in terms of run-time when executed on large-scale data-sets.

The complexity of the problem and the non-scalability of the greedy approximation algorithm opened the chase after scalable seed selection heuristics. One of the most popular heuristics is identifying influential nodes, based only on the network structure. This solution can be addressed via graph-based metrics, such as centrality measures [6].

One way to measure a node's centrality is by counting the number of its connections (known as the node's degree). While calculating the degree of a node is a relatively trivial task, such an approach is limited since it considers only the first-order effect, without considering higher-order effects. Other frequently used centrality measures that take into account high-order effects include the PageRank [41], the Betweenness Centrality [7] and the Eigenvector Centrality [5]. Each of these measures has its own attributes and represents a different type of importance that characterizes a node. For a good source on centrality measures, the reader is referred to [6] and [37].

With respect to influence maximization, several works investigated the efficiency of seeding central nodes. The work by Hinz et al. [22], for example, investigated four seeding strategies: Hubs (Degree/EigenVector Centrality), Bridges (Betweenness Centrality), Fringes (Edge Nodes) and Random. The authors conducted three experimental studies of adoption using a small controlled network; a real social network of selected students; and a large-scale cellular network. The study found that targeting Hubs is the most effective strategy in terms of influence maximization, with the Bridges strategy right afterwards, both with a big gap above the Random strategy (150-200%) and a huge gap above the Fringes strategy. Similar results were obtained by Banerjee et al. [3], where the authors investigated empirically the spread of financial loan systems within a social network of Indian villagers. The authors found that villagers with high Eigenvector Centrality scores are more likely to influence others in their surroundings, in comparison to the other measures of centrality.

Another notable group of seeding heuristics are the *CELF* [31] and *CELF++* [20] algorithms, which are based on a "lazy-forward" optimization scheme for selecting the seeds. Their underlying idea is based on bounding the marginal contribution of a node in a future iteration, with its marginal contribution in a previous iteration due to monotonicity and sub-modularity properties of the influence maximization problem. These heuristics provide an efficient variation of the greedy approximation algorithm by improving the order of evaluating nodes to be added to the "seed set". Empirical evaluation showed that the proposed heuristics outperform (in terms of influence maximization) and run faster than the greedy algorithm, while still guaranteeing a constant factor approximation of the optimal solution. Similarly, [15], [14] show the advantages of using a marginal gain of influence to form the seed set that will provide better propagation throughout the network.

Chen et al. suggested a different group of seeding heuristics [10], [9], [27], [11]. In [10] they presented an improved greedy algorithm for seeding outcome evaluation by reducing the search space per each evaluation, and showed a 700-times faster performance on the independent cascade model. In [9] they suggested the Maximum Influence Path (PMIA) algorithm. Using this method under the independent cascade model, the authors suggested to locate the nodes whose seeding will result in a long chain of cascades with the highest probability. In [27] they proposed the Influence Rank Influence Estimation (IRIE) algorithm, which performs an estimation of the influence function for any given seed set, using precomputed influence estimated values for iterative seed set ranking. Empirical simulations have shown that the IRIE heuristic performance is similar to that of the Greedy, PMIA and Pagerank influence heuristics, while its memory consumption provides a significant improvement over that of the other heuristics.

For up-to-date surveys on traditional seeding strategies for influence maximization, the reader is referred to [42], [32], [4].

2.4 Adaptive Seeding Strategies

The majority of existing works that dealt with the influence maximization problem, focused on selecting a subset of network nodes, that if seeded simultaneously at the beginning of the process, would maximize the adoption rate at the end of the process. Recently, numerous works presented a new adaptive approach, which spreads the seeding actions over time, and therefore allows to reassess the contribution of the seeds' selection in each timestamp, in order to improve the overall adoption rate.

For example, Seeman et al. [43] presented a two-stage framework for influence maximization. The underlying assumption of this model is that besides of the "non-active" (susceptible) and "active" (infective) states there is an intermediate state referred to as "available": a node v is considered available for seeding only if one of its neighbors

$w \in N(v)$ is active. Given an initial set of available nodes $X \subseteq V$, the goal of the first stage is to select a seeding set $S \subseteq X$ in order to extend the set of available nodes, so that the seeding actions in the second stage will maximize the expected influence. The idea behind it relies on the known fact that selecting a neighbor of a random node v is likely to have a higher degree than v itself and thus one would like to include those higher-degree nodes in the set of available nodes for seeding.

In another study, Tong et al. [47] suggested an adaptive seeding strategy for a variant of the Independent Cascade model. In this variant, referred to as "Dynamic Independent Cascade" model, the authors assume that the activation of a node v by seeding occurs with a probability p_v. Therefore, in contrast to the models surveyed above, a seeding action may fail, keeping the node in a non-active state. Under this setting, the authors suggest an adaptive seeding approach, in which the selection of nodes to be seeded at each timestamp, is performed while taking into account the realization of the previous seeding attempts.

Jankowski et al. [24], [25], [26] suggested an adaptive seeding approach to the influence maximization problem under the Independent Cascade model. The authors show that, regardless of the chosen strategy for selecting influential nodes, spreading the seeding actions along different timestamps of the diffusion process can improve the overall adoption rate and these results are further supported by Iyer et al. [23]. Moreover, they present an inherent trade-off between the obtained adoption rate and the duration of the diffusion process.

In another study by Ni [39], the author proposed a Markov decision process optimization within an "Incremental Chance" diffusion framework. According to the contagion model, the probability of a node to get activated is proportional to the fraction of its infected neighbors, and once a node becomes active, it remains infective. The goal in this case is to minimize the time taken to reach a complete influence by selecting the seeding set, under the constraint that only a portion of the budget is available at each timestamp. In more recent research by Ni et al. [38], the authors proposed an improved entropy-based centrality measure which takes into account the weight of connections and a confidence level. They show the superiority of their method for sequential seeding compared to other state-of-the-art centrality measures in terms of diffusion speed and influence convergence.

Chierichetti et al. [12] introduced a different diffusion model in which there are two competing ideas, each aiming at maximizing its spread over a social network. The goal of the marketer in this setting is to determine the best order to address the individuals in order to maximize the amount of adopters. The authors also provide an efficient greedy algorithm that ensures the best achievable solution to the problem.

Lin et al. [33] suggested the "Push-Driven Cascade" model in which the probability that a node will become active after a seeding action is determined by the activation state of its neighbors including the node's bias towards the adoption. The marketer's role in this setting is to choose a single node to seed at each timestamp to maximize the overall adoption in the network.

Sela et al. [45] proposed a diffusion model, named Active Viral Marketing (AVM), which better fits real-world marketing scenarios. According to this model, adoption of products relies on continuous active promotion efforts by the marketer, and the success of a marketing attempt to infect a potential customer (uninfected node), depends on the number of adopting friends (infected neighbors) of this customer. Specifically, a customer is more likely to adopt a product if more of his/her friends have already adopted it, while taking into account that social influence diminishes over time due to a memory-loss effect. The authors also proposed a set of heuristics to schedule the marketing attempts. The main idea behind these heuristics is to consider both the information on the dynamic adoption-states of neighbor nodes, as well as the static topology of the social network, when choosing the next node to seed.

It is important to emphasize that in the three latter models, each node has an accumulated influence in favor of the product, but only the seeding act itself is considered to be the trigger for activation, where the viral spread serves only as a positive effect on the activation probability. This contradicts classical diffusion models where nodes could become active as a result of a viral infection without any external intervening operation.

Goldenberg et al. [16] identified three different properties of existing contagion models that can be utilized by a scheduled seeding approach to improve the total number of activated nodes: (1) stochastic dynamics, (2) *complex contagion* and *diminishing social influence* effects, and (3) state dependent seeding. By analyzing each of these properties separately, they demonstrate the advantages of the scheduled seeding approach over the traditional initial seeding approach, both in theory and by empirical evaluation.

Several studies considered the use of sequential studies in networks with more complex structures. For example, Michalski et al. [35] presented the advantage of using sequential seeding in temporal networks. As another example, Bródka et al. [8] suggested to use sequential seeding in multi-layer networks.

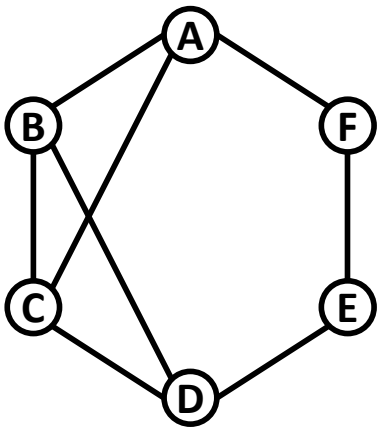
To the best of our knowledge, all existing scheduled seeding works tested the potential of scheduled seeding in theory and/or by simulations (with or without real-world network topologies).

3 METHODOLOGY

This work examines the potential of scheduled seeding by analyzing a real-world large-scale dataset, containing both the network topology as well as the infection times of nodes in that network.

In particular, we first propose a method to quantify the infection probability of a node with a given set of properties in a given timestamp, using historical data (subsection 3.1). Then, we suggest a method to assess the potential of a given seeding strategy to infect nodes by using historical data (subsection 3.2).

For the illustrative examples discussed in this section, we will consider the network depicted in Fig. 1, which comprises of six nodes and eight undirected edges, and the infection times of these nodes.



v	I
A	1
B	3
C	4
D	2
F	2

Fig. 1. An illustrative example of a network with six nodes and eight edges. The infection times of these nodes are presented in the table on the right, where each entry represents a node and its infection time (in case it became infected).

3.1 Quantifying Infection Probabilities

We are interested in quantifying the infection probability of a node with a given set of properties in a given timestamp, using historical data. The infection probability of nodes can be calculated in a relatively straightforward manner when node properties remain static over time (e.g., the number of neighbors a given node has). This can be obtained, for example, by calculating the number of infected nodes with that property, divided by the total number of nodes with that property (i.e., not necessarily infected), at the end of the diffusion process. In contrast, here we are interested in properties that can change over time, such as the number of infectious neighbors a given node has, making such a calculation more challenging.

To illustrate this point, consider the network and infection times from Fig. 2. Simply calculating the number of infected nodes with m infectious neighbors, divided by the total number of nodes with m infectious neighbors at the end of the diffusion process, would provide a false representation of reality. Specifically, node B became infected at timestamp $t = 3$, and had 0 infected neighbors at timestamp $t = 0$, 1 infected neighbor at timestamp $t = 1$, 2 infected neighbors at timestamps $t = 2$ and $t = 3$, and 3 infected neighbors at timestamp $t = 4$. According to the simple calculation, the infection of node B would be associated with the property of 3 infectious neighbors (which is the number of infectious friends B had at the end of the diffusion process), while clearly node B became infected at timestamp $t = 3$ when it had less than 3 infectious neighbors.

To cope with this challenge, the method we propose here (QIP) creates a series of snapshots, where each snapshot represents a static point in time. Each such snapshot contains static information about the properties of nodes, and therefore can be analyzed easily.

QIP receives as input the network topology ($G(V, E)$), a hash table indicating the infection timestamp of nodes (I), the set of node attributes of interest (F) - e.g. the number of infectious neighbors a node has, the start time and end time of the time period we plan to analyze (t_s, t_e), and the timestamp interval which determines the length of each snapshot ($step$).

Algorithm 1 Quantify Infection Probability (QIP)

Input: $G(V, E)$ - The network topology
I - The hash table indicating for each node its timestamp of infection
F - The set of attributes
t_s - The start time
t_e - The end time
$step$ - The timestamp interval
Output: A_F - a hash table indicating for each combination of attribute values, its corresponding infection probability

1: $n \leftarrow \{\}$
2: $n_I \leftarrow \{\}$
3: **for** $(t = t_s; t < t_e; t = t + step)$ **do**
4: $\quad n^t \leftarrow \{\}$
5: $\quad n_I^t \leftarrow \{\}$
6: $\quad$ **for** $v \in V$ **do**
7: $\quad\quad$ **if** $v \notin I \vee I[v] \geq t$ **then**
8: $\quad\quad\quad f \leftarrow extract\_attribute\_value(G, v, t, F)$
9: $\quad\quad\quad$ **if** $f \notin n^t$ **then**
10: $\quad\quad\quad\quad n^t[f] \leftarrow 0$
11: $\quad\quad\quad\quad n_I^t[f] \leftarrow 0$
12: $\quad\quad\quad$ **end if**
13: $\quad\quad\quad n^t[f] \leftarrow n^t[f] + 1$
14: $\quad\quad\quad$ **if** $I[v] \in [t, t + step)$ **then**
15: $\quad\quad\quad\quad n_I^t[f] \leftarrow n_I^t[f] + 1$
16: $\quad\quad\quad$ **end if**
17: $\quad\quad$ **end if**
18: $\quad$ **end for**
19: $\quad$ **for** $f \in n^t$ **do**
20: $\quad\quad$ **if** $f \notin n$ **then**
21: $\quad\quad\quad n[f] \leftarrow 0$
22: $\quad\quad\quad n_I[f] \leftarrow 0$
23: $\quad\quad$ **end if**
24: $\quad\quad n[f] \leftarrow n[f] + n^t[f]$
25: $\quad\quad n_I[f] \leftarrow n_I[f] + n_I^t[f]$
26: $\quad$ **end for**
27: **end for**
28: $A_F \leftarrow \{\}$
29: **for** $f \in n_I$ **do**
30: $\quad A_F[f] \leftarrow n_I[f]/n[f]$
31: **end for**
32: **return** A_F

Then, the algorithm initializes two hash tables (lines 1 and 2). The first, n, stores for each possible combination of values for F (recall that F is a set of attributes), the total number of nodes that were observed with that combination of values. The second, n_I, stores for each possible combination of values for F, the total number of infected nodes that were observed with that combination of values.

Next, in lines 3-27, the algorithm iterates over the various snapshots (starting at timestamp $t = t_s$ to timestamp $t = t_e$ with jumps of size $step$). For each such snapshot, the algorithm initializes two hash tables (lines 4 and 5).

The first, n^t, stores for each possible combination of values for F, the number of nodes that were observed with that combination of values, during this snapshot. The second, n_I^t, stores for each possible combination of values for F, the total number of non-infected nodes that were observed with this combination of values and will become infected in the next timestamp. Then, in lines 6-18, the algorithm iterates over the different nodes, retrieves the combination of values for F for that node (f) using the $extract\_attribute\_value$ function in line 8, increments the corresponding entry in n^t by 1 (line 13) and similarly increments the corresponding entry in n_I^t by 1 if the node became infected during the next timestamp (lines 14-15). The iteration ends in lines 19-26, where the values in n^t and n_I^t that were accumulated during this snapshot, are added to n and n_I which contain information about the entire diffusion process.

Finally, when done iterating over all snapshots, the algorithm calculates the ratio for each observed combination of values for F (lines 28-31), and returns the result (line 32).

Fig. 2 illustrates the operation of QIP. For that purpose, we use the same network topology (G) and the same infection times (I) from Fig. 1. The (single) attribute we are examining in this example (F) is the number of infectious neighbors. Finally, $t_s = 0$, $t_e = 3$, and $step = 1$.

The figure contains four subfigures, each representing a single snapshot. In each subfigure, nodes who are going to be infected in the next snapshot are marked in yellow, nodes who are already infected in the current snapshot are marked in red and susceptible nodes are marked in white. The table at the right-top corner of each subfigure represents n^t and n_I^t, and the table at the right-bottom corner represents the aggregated values for n and n_I over all snapshots so far (including the snapshot at timestamp t).

At timestamp $t = 0$, all nodes have 0 infectious neighbors, and the only node that is going to become infected in the next timestamp is A. Therefore, for $f = 0$ (where f represents a concrete combination of values for the attributes in F) we have $n^0 = 6$ and $n_I^0 = 1$.

At timestamp $t = 1$, nodes D and E have 0 infectious neighbors, and nodes B, C and F have 1 infectious neighbor. Therefore, for $f = 0$ we have $n^1 = 2$ and because node D will get infected in the next timestamp, we have $n_I^1 = 1$. Similarly, for $f = 1$ we have $n^1 = 3$ and because node F will get infected in the next timestamp we have $n_I^1 = 1$.

At timestamp $t = 2$, nodes B, C and E have 2 infectious neighbors. Therefore, for $f = 2$ we have $n^2 = 3$ and because node B will get infected in the next timestamp we have $n_I^2 = 1$.

At timestamp $t = 3$, node E has 2 infectious neighbors, and node C has 3 infectious neighbors. Therefore, for $f = 2$ we have $n^3 = 1$ and because node E will not get infected in the next timestamp we have $n_I^3 = 0$. For $f = 3$ we have $n^3 = 1$, and because node C will get infected in the next timestamp, we have $n_I^3 = 1$.

Finally, for $f = 0$ we have $n = 8$ and $n_I = 2$; for $f = 1$ we have $n = 3$ and $n_I = 1$; for $f = 2$ we have $n = 4$ and $n_I = 1$; and for $f = 3$ we have $n = 1$ and $n_I = 1$.

Now, based on all snapshots, we can estimate the conditional probability of a node to become infected given the number of infectious neighbors it has: for $f = 0$ we have $A_F = 2/8 = 0.25$; for $f = 1$ we have $A_F = 1/3 = 0.33$; for $f = 2$ we have $A_F = 1/4 = 0.25$; and for $f = 3$ we have $A_F = 1/1 = 1.0$.

Another way, perhaps more intuitive, to characterize $A_F[f]$ is the following. We denote the set of all nodes in V that had an attribute value of f at timestamp t and did not get infected before timestamp t by:

$$S_{f,t} = \{v \in V | extract\_attribue\_value(G, v, t, F) = f \wedge (v \notin I \vee I[v] \geq t)\}$$

Similarly, we denote the set of all nodes in V that had an attribute value of f at timestamp t and got infected between timestamp t and timestamp $t + step$ by:

$$S_{f,t}^I = \{v \in V | extract\_attribue\_value(G, v, t, F) = f \wedge v \in I \wedge I[v] \in [t, t + step)\}$$

Therefore, $A_F[f]$ is simply:

$$A_F[f] = \frac{\sum_{t=t_s}^{t_e} |S_{f,t}^I|}{\sum_{t=t_s}^{t_e} |S_{f,t}|}$$

3.2 Assessing the Potential of a Seeding Strategy

QIP allows us to quantify the infection probability of a node with a given set of properties in a given timestamp, using historical data. In principle, such probabilities can be used to rank nodes to be seeded, where nodes with higher infection probabilities are ranked higher. However, it is important to note that the highest infection probability obtained by QIP by itself is insufficient to determine the performance of such a seeding strategy.

To illustrate this point, consider again the example from Fig. 2. Since the infection probability of a node with 3 infectious neighbors is 100%, one may infer that using a seeding strategy which ranks nodes based on these infection probabilities, can lead to a 100% infection rate. However, this inference is wrong. For example, examining Fig. 3, we see that at timestamp $t = 0$, none of the nodes have 3 infectious neighbors. In fact, all nodes have exactly 0 infectious neighbors, and therefore, no matter which node is selected to be seeded, its infection probability is considerably lower than 100%. Consequently, the infection rate at the level of the entire strategy will also be lower than 100%.

To cope with this challenge, the method we propose here ($APSS$) takes into account both the seeding budget as well as the the entire set of susceptible nodes and their scores at each snapshot.

$APSS$ receives as input the network topology ($G(V, E)$), a hash table indicating the infection timestamp of nodes (I), the set of node attributes of interest (F), the start time and end time of the time period we plan to analyze (t_s, t_e), the timestamp interval which determines the length of each snapshot ($step$), and the seeding budget (B).

Then, the algorithm initializes the set of nodes that were seeded thus far, S, to an empty set (line 1). In line 2, the algorithm determines the seeding budget B^t for each snapshot, by distributing the overall seeding budget B evenly over the various snapshots (line 2).

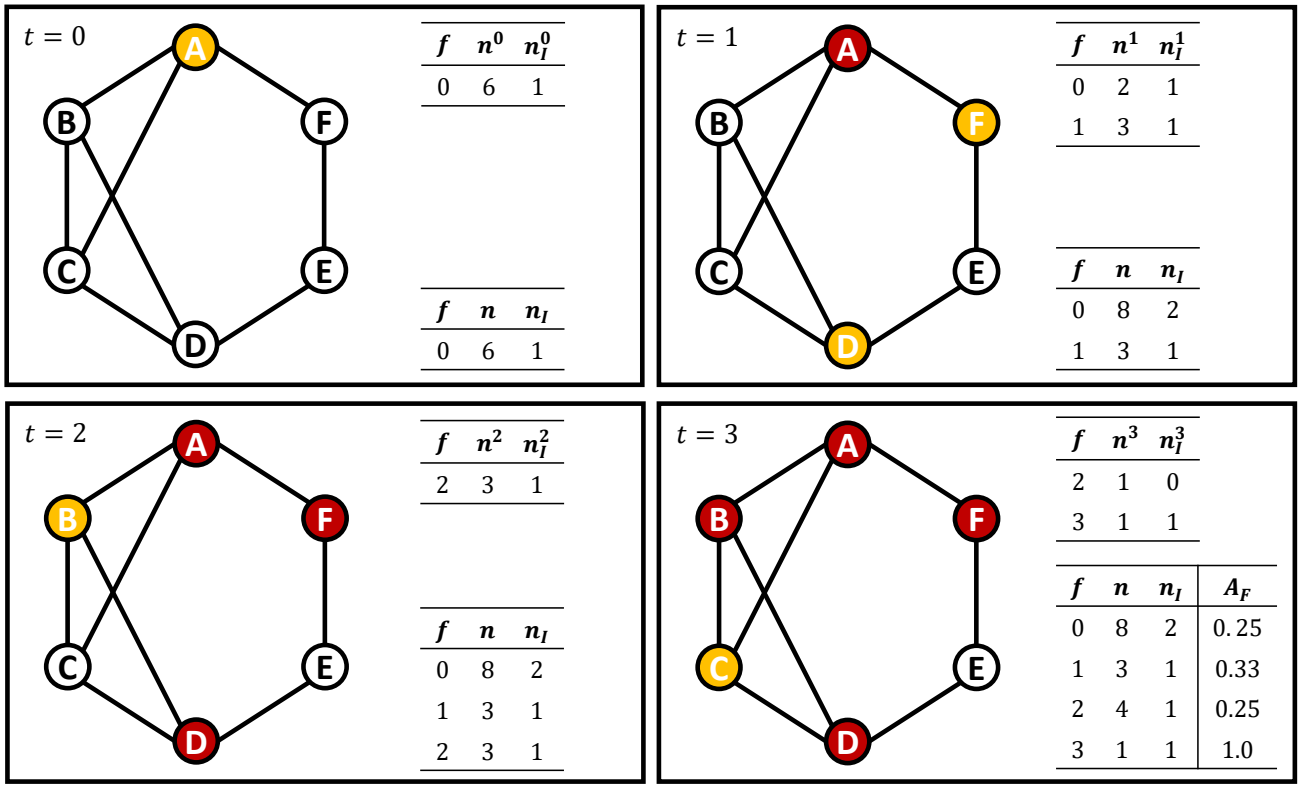


Fig. 2. Illustration of the operation of QIP.

Next, in lines 3-20, the algorithm iterates over the various snapshots (starting at timestamp $t = t_s$ to timestamp $t = t_e$ with jumps of size $step$). For each such snapshot, the algorithm initializes the set of nodes to be seeded during this snapshot (S^t), and a hash table that stores the score of each node ($score$) to be empty (lines 4 and 5). In lines 6-12, the algorithm iterates over all nodes that were not seeded and have not become infected thus far (line 7). For these nodes, the algorithm retrieves the combination of values for F for that node (f) using the $extract\_attribute\_value$ function (line 10), applies the $calc\_attribute\_score(f)$ function to obtain a score for that combination of values, and stores the obtained score in the $score$ hash table (line 11). In line 13, the algorithm sorts the nodes by their scores in descending order using the $sort\_keys\_by\_values(score)$ function. Then, in lines 14-18, the algorithm chooses nodes to be seeded and adds them to S^t, until reaching the seeding budget for this snapshot. After reaching the seeding budget of this snapshot, all nodes in S^t are added to S (line 19).

Finally, when it finishes the iteration over all snapshots, the algorithm counts how many of the nodes that were chosen to be seeded, actually became infected (lines 21-26) and returns the result in line 27.

Fig. 3 illustrates the operation of $APSS$. For that purpose, we use the same network topology (G) and the same infection times (I) from Fig. 1. The attribute we are examining in this example (F) is the number of infectious neighbors. Finally, $t_s = 0$, $t_e = 3$, and $step = 1$.

The figure contains four subfigures, each representing a single snapshot. In each subfigure, nodes who were chosen to be seeded at that snapshot are marked with green dashed outline, nodes who were chosen to be seeded in the past are marked with green solid outline, infected nodes are marked in red and susceptible nodes are marked in white. The table at the right-top corner of each subfigure represents the sorted list of nodes by their scores, and the set at the right-bottom corner represents the set of nodes who were chosen for seeding in all snapshots thus far. The table in the last snapshot shows the nodes that were selected to be seeded (over all snapshots) and whether they became infected.

In this example, we use the infection probabilities from Fig. 2 as the scores for nodes. Furthermore, we assume that the seeding budget for each snapshot is $B^t = 1$.

At timestamp $t = 0$, all nodes have $f = 0$ infectious neighbors, and therefore they all have $score = 0.25$. After sorting the nodes by their scores, node E was (arbitrarily) chosen to be seeded and was added to S^t, as well as to to S.

At timestamp $t = 1$, node D has $f = 0$ infectious neighbors, and nodes B, C and F have $f = 1$ infectious neighbor. Therefore, node D has $score = 0.25$, and nodes B, C and F have $score = 0.33$. Note that node A is not considered since it became infected, and node E is not considered since it was already seeded previously. After sorting the nodes by their scores, node F was chosen to be seeded and was added to S^t, as well as to to S.

At timestamp $t = 2$, nodes B and C have $f = 2$ infectious neighbors, and therefore they both have $score = 0.25$. After sorting the nodes by their scores, node C was chosen to be seeded and was added to S^t, as well as to S.

At the end of this process we can see that the set of nodes that were chosen to be seeded is $S = \{E, F, C\}$. Nodes F

Algorithm 2 Assessing the Potential of a Seeding Strategy ($APSS$)

Input: $G(V, E)$ - The network topology
I - The hash table indicating for each node its timestamp of infection
F - The set of attributes
t_s - The start time
t_e - The end time
$step$ - The timestamp interval
B - The seeding budget
Output: R - The total number of seeded nodes that became infected

1: $S \leftarrow \{\}$
2: $B^t \leftarrow B/[(t_e - t_s)/step]$
3: **for** $(t = t_s; t < t_e; t = t + step)$ **do**
4: $S^t \leftarrow \{\}$
5: $score \leftarrow \{\}$
6: **for** $v \in V$ **do**
7: **if** $v \in S \vee (v \in I \wedge I[v] < t)$ **then**
8: continue
9: **end if**
10: $f \leftarrow extract\_attribute\_value(G, v, t, F)$
11: $score[v] \leftarrow calc\_attribute\_score(f)$
12: **end for**
13: $ranked \leftarrow sort\_keys\_by\_values(score)$
14: **for** $v \in ranked$ **do**
15: **if** $|S^t| < B^t$ **then**
16: $S^t \leftarrow S^t \cup \{v\}$
17: **end if**
18: **end for**
19: $S \leftarrow S \cup S^t$
20: **end for**
21: $R \leftarrow 0$
22: **for** $v \in S$ **do**
23: **if** $v \in I$ **then**
24: $R \leftarrow R + 1$
25: **end if**
26: **end for**
27: **return** R

and C became infected after their seeding, while node E did not become infected at all. Therefore, out of 3 seeding attempts, only 2 were successful, and the algorithm returns $R = 2$.

3.3 Runtime Complexity Analysis

The runtime complexity of the QIP algorithm is given by:

$$O(\frac{t_e - t_s}{step} \cdot |V| \cdot O_{eav})$$

where O_{eav} is the runtime complexity of the $extract\_attribute\_value$ function. More specifically, the loop in lines 3-27 has $\frac{t_e - t_s}{step}$ iterations, the loop in lines 6-18 has V iterations, the loop in lines 19-26 has $|n^t| \leq |V|$ iterations, and the loop in lines 29-31 has $|n_I| \leq |V|$ iterations. All other operations, except $extract\_attribute\_value$ have a runtime complexity of $O(1)$. The $extract\_attribute\_value$ function may have an arbitrary runtime complexity which we denote as O_{eav}.

Similarly, the runtime complexity of the $APSS$ algorithm is given by:

$$O(\frac{t_e - t_s}{step} \cdot |V| \cdot log(|V|) \cdot O_{eav})$$

More specifically, the loop in lines 3-20 has $\frac{t_e - t_s}{step}$ iterations, the loop in lines 6-12 has V iterations, the loop in lines 14-18 has $|ranked| \leq |V|$ iterations, and the loop in lines 22-26 has $|S| \leq |V|$ iterations. All other operations, except $extract\_attribute\_value$ and $sort\_keys\_by\_values$ have a runtime complexity of $O(1)$. The runtime complexity of $extract\_attribute\_value$ was denoted by O_{eav}, and $sort\_keys\_by\_values$ has a runtime complexity of $O(|V| \cdot log(|V|))$.

4 The Dataset

The dataset used in this study was provided to us by a relatively large mobile network operator. Data originating from mobile network operators was found to be very appealing for the purpose of this study since such companies hold information about the social network of their customers that can be derived from the communication network (i.e., calls and text messages), as well as information on the adoption time of products or services that these companies offer to their customers[1].

With regard to the adoption information, we focus on a specific service that the mobile network operator offers to its customers, namely, the repair service, which can be seen as an insurance plan for customers' mobile devices. We chose this service over other products and services that the company offers since the company does not actively advertise this service as part of its ongoing marketing campaigns. This allowed us to neutralize various effects that might have interfered with this study's objective and isolate the effect of the viral marketing process over the network.

In the following subsections we describe the data that we received in details (subsection 4.1) and explain the adjustments made to the data to fit this study (subsection 4.2).

4.1 Data Description

The received dataset contains the metadata for calls and text messages (in-going and out-going records) of 9,549 sampled customers between September 1, 2018 and November 26, 2018 (we denote this group of sampled customers by W). Customers in W were sampled randomly over all the customers who satisfied the following criteria: customers who had a successful repair in the given period, thus, used the repair service between September 1, 2018 to November 26,

1. We would like to emphasize that in general, it is very difficult to obtain combined information about both the structure of the social network as well as on the adoption of a product over this network. Clearly, companies such as Facebook hold similar data, but such data is typically not publicly available. When it comes to publicly available data, various samples of networks (including Facebook) are available. However, while these samples typically include the network structure, they do not include information about products' adoption. Nevertheless, it should also be noted that we believe our methods can be generalized to other products and other settings such as that of Facebook.

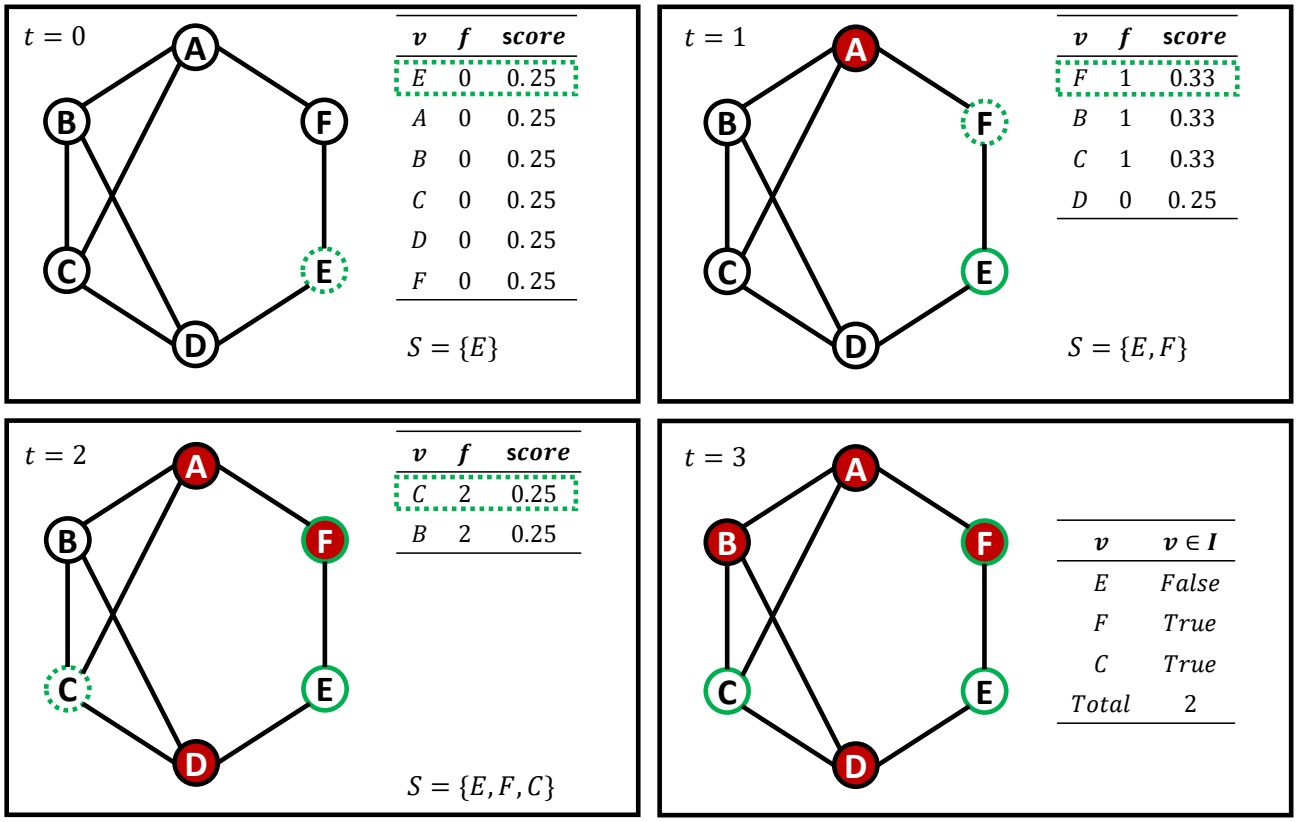


Fig. 3. Illustration of the operation of $APSS$.

2018, did not use the repair service again in the following week, and did not pay additional charges for their repair.

For each customer in W, we received all of its interactions (calls or text messages) with other customers of the company during the mentioned period. Overall, we received 6,283,938 interactions, where customers in W interacted with 259,106 unique customers (we denote this group of unique customers by U).

In addition, for each customer in W we obtained the date of its most recent successful repair, and for each customer in U we received the date in which they joined the repair service (if at all).

To summarize, the dataset provided to us included 6,283,938 records, where each record (a single interaction between $w \in W$ and $u \in U$) in the dataset contained the following attributes:

- W's customer encrypted id
- U's customer encrypted id
- Date and time of the interaction
- Interaction type (call or text message)
- Duration of the call (0 for text messages)
- Incoming or outgoing record
- The last repair date for W
- Date and time of U joining the repair service (or $null$ if U did not join the repair service at all)

It is important to note that the data was provided to us under strict privacy guidelines, which included among the rest: (1) using pseudo-identifiers instead of real customer identifiers, (2) limiting the sample size (group W) up to 10,000 customers only, (3) limiting the call network to one hop from the sampled population (i.e., only to the circle of customers that interacted with the sampled population, and not with additional circles), and (4) personal information about the customers such as gender, age, address, etc. was not shared with us at all.

4.2 Data Adjustments

Recall that the proposed algorithms in section 3 expect to receive two data structures, $G(V, E)$ which represent the social network topology and I which contains information about the time in which each node $w \in W$ got infected.

In order to derive the social network from this dataset, we followed two approaches. The first was to extract an interaction network which included a node for each customer in $W \cup U$ and an edge (v_1, v_2) between the two nodes v_1 and v_2 was created if there was at least one mobile interaction (i.e., call or text message) between the customers v_1 and v_2 in the dataset. The second approach was based on a friendship network that was built in a very similar manner to the interaction network, except that an edge (v_1, v_2), representing a friendship relationship between the two nodes v_1 and v_2, was created only if there was a reciprocal interaction between the customers v_1 and v_2 (i.e., at least one interaction that v_1 initiated and at least one interaction that v_2 initiated) in the dataset. This is a common approach used in network science literature to transform interactions into friendship relationships [13], [40]. It should be noted that we also experimented another approach to extract friendship relationship, by which an edge (v_1, v_2), is created only if

the average call duration between v_1 and v_2 is higher than the median call duration in the entire dataset. The main results of this paper remained consistent under this altered definition of friendship.

The calculation of I is straightforward: it contains a single entry for each node $u \in U$ which joined the repair service and include u as well as the time in which u joined the repair service. That is, I does not contain entries for nodes $u \in U$ which did not joined the repair service.

It is important to emphasize that unlike other studies in this field, we make a clear distinction between an infected node, which in our case represents a customer that joined the repair service, and an infectious node, which in our case represents a customer who had a successful repair in the given time period. Accordingly, we also use a third dataset, J which contains a single entry for each node $w \in W$. Each such entry contains w as well as the time of w's latest successful repair. The use of J will become clearer in section 5.1.

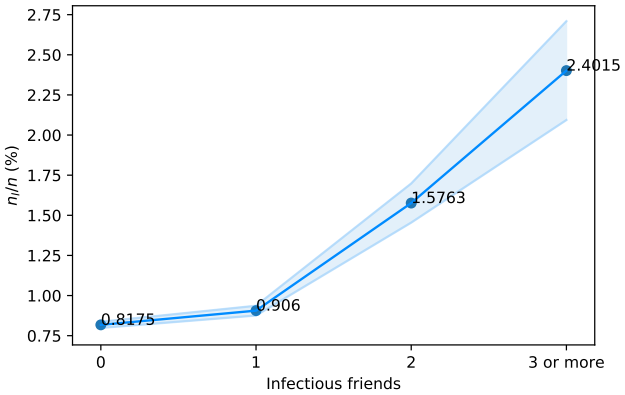


Fig. 4. Infection likelihood of customers as a function of the number of infectious friends they have. The X-axis represents the number of infectious friends a customer has, and the Y-axis represents the likelihood of a customer with that given "number of infectious friends" to become infected (join the repair service), as calculated by QIP. The colored area represents the 95% confidence interval of the proportion.

5 Results

5.1 Quantifying Infection Probabilities

In this subsection, we demonstrate the use of QIP on our dataset.

In particular, we focus on three attributes of interest. The first attribute is the number of infectious friends a node has at any given point in time. Applying QIP with this dynamic attribute in mind, allow us to evaluate whether the *complex contagion* effect is present in the dataset. The second attribute is the mean time that has passed since the friends of a given node became infectious. Applying QIP with this dynamic attribute in mind, allow us to evaluate whether a *diminishing social influence* effect is present in our dataset. The third set of attributes is a combination of these two dynamic attributes, which would allow us to evaluate whether a joint effect is present in our dataset.

For that purpose, friendship relationships were calculated as described in subsection 4.2; a customer was considered infectious in a given time if they had a successful repair earlier in time (i.e., as indicated by J, see section 4.2); and a customer is considered infected in a given time, if they joined the repair service at that time (i.e., as indicated in I, see section 4.2). Finally, we note that when we applied the QIP algorithm, we considered a time resolution of two weeks (i.e., 14 days). That is, in order to have enough data in each snapshot, we divided the time frame into consecutive periods, each of two weeks, and extracted a single snapshot of the network for each such period. We also considered other time resolutions (e.g., 7 days) and the results were consistent.

5.1.1 Number of Infectious Friends Attribute

Fig. 4 shows the results of applying QIP on the dataset, while considering the number of infectious friends dynamic node attribute. As can be seen, the likelihood of a customer to become infected grows with the number of infectious friends it has. Specifically, the likelihood of a customer with 3 or more infectious friends to become infected is almost 3 times higher than that of a customer with 0 infectious friends (2.4% vs. 0.82% respectively). This result implies the existence of the *complex contagion* effect.

5.1.2 Mean Time from Infectiousness Attribute

Fig. 5 shows the results of applying QIP on our dataset, when considering the mean time from infectiousness dynamic node attribute. As can be seen, the likelihood of nodes to become infected generally decreases with the mean time that has passed since its friends became infectious. Specifically, in the case of 2 time periods that have passed (i.e., 1 month), the likelihood of a customer to become infected is 60% higher than in the case of 4 time periods or more (i.e., 2 months or more), 1.22% vs. 0.76% respectively. This result implies the existence of the *diminishing social influence* effect.

An interesting exception to the decrease in infection likelihood is the case of a single time period (i.e., 2 weeks), for which the infection likelihood is 14% lower than that of 2 time periods (i.e., 1 month): 1.22% vs. 1.05% respectively. We conjecture that this happens since the infection operation itself takes some time. Specifically, in our setting, customers that had a successful repair, might prefer to wait a bit before recommending the service to their friends (e.g., to make sure the repair is indeed successful), and their friends might take a while to join the service (e.g., since it requires a certain amount of effort, like calling the service provider).

5.1.3 Number of Infectious Friends and Mean Time from Infectiousness Attributes

Fig. 6 shows the results of applying QIP on our dataset, when considering both the number of infectious friends and the mean time from infectiousness dynamic node attributes. As can be seen, the highest likelihood for a customer to become infected is obtained in the case of 3 or more infectious friends and 2 time periods that have passed on average since these friends became infectious (3.24%). The infection likelihood in this case is almost 4 times higher than in the case of 0 infectious friends (3.24% vs. 0.82%). This result strengthen the existence of the *complex contagion* and *diminishing social influence* combined effect. As shown by Goldenberg et al. [16], the scheduled seeding approach is expected to benefit greatly from such a combined effect.

Fig. 6 highlights another important property which motivated the development of $APSS$: while the case of 3

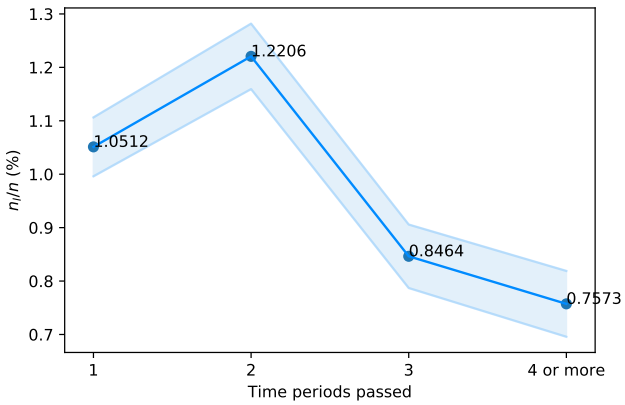


Fig. 5. Infection likelihood of customers as a function of the mean time from infectiousness. The X-axis represents the mean time from infectiousness (in time periods of 2 weeks each), and the Y-axis represents the likelihood of a customer with that given "mean time from infectiousness" time periods to become infected (join the repair service), as calculated by QIP. The colored area represents the 95% confidence interval of the proportion.

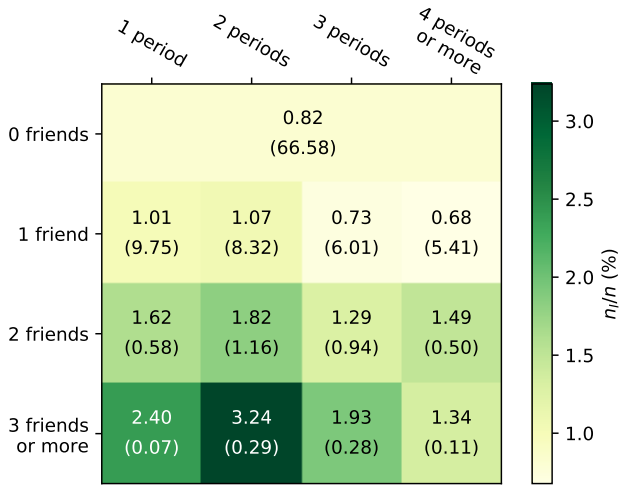


Fig. 6. Infection likelihood of customers as a function of the number of infectious friends and the mean time from infectiousness. The rows represents the number of infectious friends a customer has, and the columns represents the mean time that has passed since they became infectious (in time periods of 2 weeks each). Each entry represents the likelihood of a customer with that given "number of infectious friends" and "mean time from infectiousness" values to become infected (join the repair service), as calculated by QIP. In parenthesis, we present the percentage of cases we observed with such values.

or more infectious friends and 2 time periods presents an infection likelihood which is 4 times higher than that of 0 infectious friends, it appears only in 0.29% of the observed cases, whereas in the vast majority of the observed cases are associated with 0 infectious friends.

5.2 Assessing the Potential of Scheduled Seeding

In this subsection, we demonstrate the use of $APSS$ on our dataset, with the goal of evaluating the potential of the scheduled seeding approach. To that end, we considered 7 different seeding strategies. The first three strategies, which are commonly used as benchmark seeding strategies in the literature, use static node attributes, and include:

- *Random* - Nodes to be seeded are selected randomly.
- *Degree (Interactions)* - Nodes to be seeded are selected according to their degree in the interaction network (see section 4.2).
- *Degree (Friends)* - Nodes to be seeded are selected according to their degree in the friendship network (see section 4.2).

The remaining four strategies use dynamic node attributes, and rely on QIP to obtain a score for a given node in a given timestamp based on its attributes values at that timestamp. Three of these strategies consider the dynamic node attributes that were described in section 5.1:

- *Number of Infectious Friends (NIF)* - Nodes to be seeded are selected according to the infection likelihood that QIP calculates for the number of infectious friends they have (see subsection 5.1.1 for more details).
- *Mean Time from Infectiousness (MTI)* - Nodes to be seeded are selected according to the infection likelihood that QIP calculates for the mean time that has passed since their friends became infectious (see subsection 5.1.2 for more details).
- *Number of Infectious Friends and Mean Time from Infectiousness (NIF&MTI)* - Nodes to be seeded are selected according to the infection likelihood that QIP calculates for the combination of number of infectious friends they have and the mean time that has passed since they became infectious (see subsection 5.1.3 for more details).

The fourth strategy, *Number of Infectious Interactions (NII)*, is considered for completeness purposes. This strategy selects nodes to be seeded according to the infection likelihood that QIP calculates for the number of infectious interactions (i.e., not necessarily friends) they have.

Fig. 7 shows the results of executing $APSS$ with the seven strategies described above and for varying seeding budgets. The X-axis represent the seeding budget as a percentage of the network size in a logarithmic scale. The Y-axis represents the percentage of successful seeding attempts (i.e., the percentage of nodes, out of those who were seeded, that became infected). The vertical dashed line represents a seeding budget of 1%.

As can be seen, for smaller seeding budget percentages, the *NIF&MTI* strategy outperforms all the other considered strategies. For example, for a seeding budget of 1%, the *NIF&MTI* strategy obtains 7.74% successful seedings attempts, compared with 6.79% obtained by the *Degree (Friends)* strategy, and only 2.45% obtained by the *Random* strategy, which is often used in practice. The following best performing strategy is *NIF*, which is only slightly worse than *NIF&MTI*. This implies that *NIF* can be used as a simplified variation of *NIF&MTI* while obtaining similar results.

In addition, we also observe the advantage of using a friendship network rather than an interaction network. For example, for a budget of 1%, the *NIF* strategy obtains 7.51% successfully seedings attempts compared with 6.81% obtained by the *NII* strategy, while the *Degree (Friends)* obtains 6.79% successfully seedings attempts compared with 5.28%

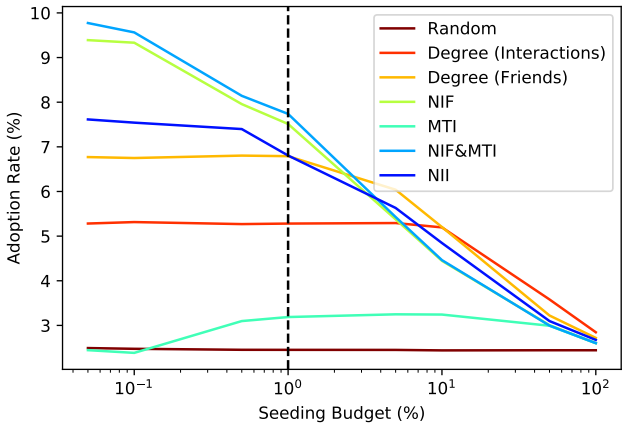


Fig. 7. Percentages of successful seeding attempts for the seven examined strategies.

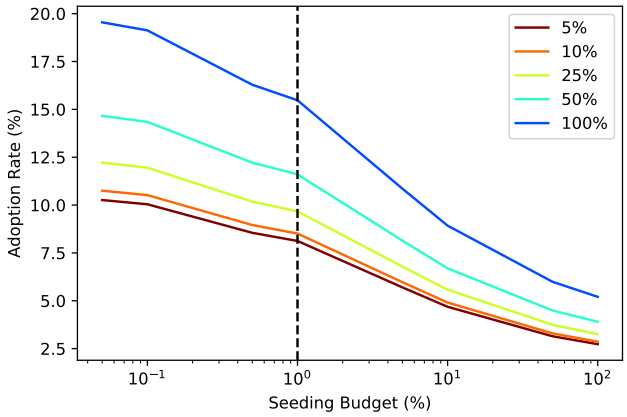


Fig. 8. Percentage of successful seeding attempts for various values of ϵ (the increase in the likelihood of a node to become infected due to seeding).

obtained by the *Degree (Interactions)* strategy. This implies that the friendship definition we presented in subsection 4.2 does manage to capture some real aspects of friendship relationships.

Finally, we notice that for most strategies, the percentage of successful seeding attempts declines when increasing the seeding budget[2], until it roughly converges with that of the *Random* strategy for a seeding budget of 100%. This is quite expected since for a seeding budget of 100%, for example, *all* nodes are selected to be seeded, regardless of their likelihood to be seeded successfully, thereby making all seeding strategies equivalent.

Fig. 7 demonstrates the ability of $APSS$ to identify nodes that have a high likelihood to become infected.

The next step in the analysis is focused on showing what would happen if such nodes are indeed seeded. Ideally, this can be achieved by performing a live experiment to compare different seeding strategies. Here, we try to achieve a similar goal via simulations based on real-data. To that end, we assume that seeding a node (e.g., calling a customer, sending them a text message, or giving them a discount) increases its likelihood to become infected by ϵ%. Such an assumption can be used to simulate the percentage of successful seeding operations for different seeding strategies.

Fig. 8 shows the results of executing $APSS$, for varying seeding budgets. This time we focus on the *NIF&MTI* strategy which was found to be the best performing strategy, and we assume that seeding operation increases the likelihood of a node to become infected by ϵ%, for $\epsilon \in \{5, 10, 25, 50, 100\}$. The X-axis represent the seeding budget as a percentage from the network size in a logarithmic scale. The Y-axis represents the percentage of successful seeding attempts (i.e., the percentage of nodes, out of those who were seeded, tha became infected, where their likelihood to become infected is increased by ϵ %). The vertical dashed line represents a seeding budget of 1%.

Clearly, the higher the value of ϵ is, the higher is the percentage of successful seeding attempts. Specifically, for a seeding budget of 1% and $\epsilon = 100\%$, 15.47% of the seeding attempts are successful.

2. Note again that the Y-axis represents the *percentage*, rather than the *number*, of successful seeding attempts.

Interestingly, under the assumption that a seeding operation increases the likelihood of a node to become infected by ϵ%, if we want to assess how good the seeding strategy is, we can ignore the value of ϵ, by calculating the performance of a strategy relatively to another strategy (e.g., *Random*). Fig. 9 shows the results of executing $APSS$ with the six strategies described above (excluding the *Random* strategy) and for varying seeding budgets. The X-axis represent the seeding budget as a percentage of the network size and is presented in in a logarithmic scale. The Y-axis represents the percentage of successful seeding attempts obtained by each strategy, divided by that of the *Random* strategy. The vertical dashed line represents a seeding budget of 1%.

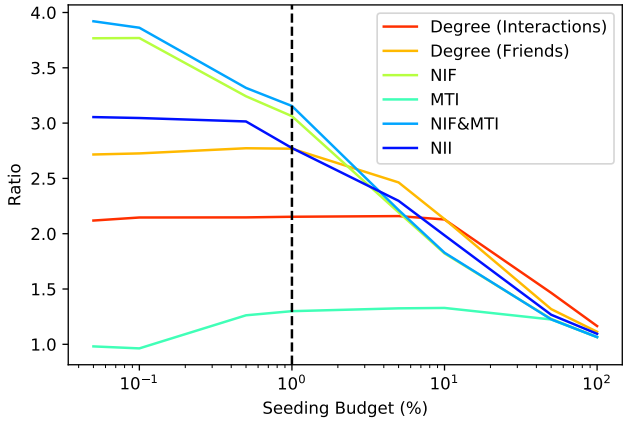


Fig. 9. Ratio of successful seeding attempts for for the 6 examined strategies divided by *Random*.

As can be seen, for a seeding budget of 1%, the proposed *NIF&MTI* strategy performs 1.14 times better (i.e., 14% better) than the *Degree (Interactions)* strategy, and 3.15 times better (i.e., 215% better) than the *Random* strategy, which is often used in practice. These results hold regardless of the value of ϵ.

6 Summary and Conclusion

This work examined the potential of scheduled seeding by analyzing a real-world large-scale dataset, containing both

the network topology as well as the nodes' (customers') infection times. We first proposed a method to quantify the infection probability of a node with a given set of properties in a given timestamp, by analyzing historical data. Then, we used the proposed method to demonstrate the existence of both a *complex contagion* effect and a *diminishing social influence* effect in the considered real-world example. Finally, we suggested a method to assess the potential of a given seeding strategy to infect nodes by using historical data, and compared a scheduled seeding strategy that ranks nodes based on a combination of the number of infectious friends they have, as well as the time that has passed since they became infectious, to a number of benchmark seeding strategies. Results of our analyses showed that this scheduled seeding strategy considerably outperform the other benchmark seeding strategies.

It is important to note that the methodology proposed in this paper considers a setting in which data is available on both the social network topology as well as the infection time of nodes during the diffusion process. Since in this work, our experiments were based on a single proprietary dataset, an important research direction would be to repeat our experiments on additional datasets. However, in many settings, such combined data is not available or is only partially available. Therefore, a closely related research direction would be to adjust the proposed methodology and evaluate it in cases of partial and/or noisy data.

The dataset used in this study was limited to a sample of roughly 10,000 customers (and the customers they interacted with) for a period of roughly three months, and contained information about the adoption of a specific service. Future research effort should be devoted to obtain a larger sample for a longer period of time. Such effort should also be devoted to examine other types products or services.

While this study can be seen as an important milestone in understanding the potential of scheduled seeding strategies, the entire evaluation made in this study was based on historical data. Another important research direction would be to compare scheduled seeding strategies and traditional initial strategies in live experiments relying on A/B testing that was unavailable in the current study.

Acknowledgements

This work was partly funded by the Kamin grant of the Israeli Chief Scientist (file number 58073).

References

[1] Roy M Anderson, Robert M May, and B Anderson. *Infectious diseases of humans: dynamics and control*, volume 28. Wiley Online Library, 1992.

[2] Solomon E Asch. Effects of group pressure upon the modification and distortion of judgments. *Groups, leadership, and men. S*, pages 222–236, 1951.

[3] Abhijit Banerjee, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. The diffusion of microfinance. *Science*, 341(6144):1236498, 2013.

[4] Suman Banerjee, Mamata Jenamani, and Dilip Kumar Pratihar. A survey on influence maximization in a social network. *Knowledge and Information Systems*, 62(9):3417–3455, 2020.

[5] Phillip Bonacich. Some unique properties of eigenvector centrality. *Social networks*, 29(4):555–564, 2007.

[6] Stephen P Borgatti. Centrality and network flow. *Social networks*, 27(1):55–71, 2005.

[7] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001.

[8] Piotr Bródka, Jarosław Jankowski, and Radosław Michalski. Sequential seeding in multilayer networks. *arXiv preprint arXiv:2009.05335*, 2020.

[9] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010.

[10] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[11] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 88–97. IEEE, 2010.

[12] Flavio Chierichetti, Jon Kleinberg, and Alessandro Panconesi. How to schedule a cascade in an arbitrary graph. *SIAM Journal on Computing*, 43(6):1906–1920, 2014.

[13] Koustuv Dasgupta, Rahul Singh, Balaji Viswanathan, Dipanjan Chakraborty, Sougata Mukherjea, Amit A Nanavati, and Anupam Joshi. Social ties and their relevance to churn in mobile telecom networks. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, pages 668–677, 2008.

[14] Xiaoheng Deng, Fang Long, Bo Li, Dejuan Cao, and Yan Pan. An influence model based on heterogeneous online social network for influence maximization. *IEEE Transactions on Network Science and Engineering*, 7(2):737–749, 2019.

[15] Xiaoheng Deng, Yan Pan, Hailan Shen, and Jingsong Gui. Credit distribution for influence maximization in online social networks with node features 1. *Journal of Intelligent & Fuzzy Systems*, 31(2):979–990, 2016.

[16] Dmitri Goldenberg, Alon Sela, and Erez Shmueli. Timing matters: Influence maximization in social networks through scheduled seeding. *IEEE Transactions on Computational Social Systems*, 5(3):621–638, 2018.

[17] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[18] Jacob Goldenberg, Barak Libai, and Eitan Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 2001:1, 2001.

[19] Avi Goldfarb and Catherine Tucker. Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 30(3):389–404, 2011.

[20] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pages 47–48. ACM, 2011.

[21] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.

[22] Oliver Hinz, Bernd Skiera, Christian Barrot, and Jan U Becker. Seeding strategies for viral marketing: An empirical comparison. *Journal of Marketing*, 75(6):55–71, 2011.

[23] Shankar Iyer and Lada A Adamic. The costs of overambitious seeding of social products. In *International Conference on Complex Networks and their Applications*, pages 273–286. Springer, 2018.

[24] Jarosław Jankowski, Piotr Bródka, Przemysław Kazienko, Boleslaw K Szymanski, Radosław Michalski, and Tomasz Kajdanowicz. Balancing speed and coverage by sequential seeding in complex networks. *Scientific reports*, 7(1):891, 2017.

[25] Jarosław Jankowski, Piotr Bródka, Radosław Michalski, and Przemysław Kazienko. Seeds buffering for information spreading processes. In *International Conference on Social Informatics*, pages 628–641. Springer, 2017.

[26] Jarosław Jankowski, Boleslaw K Szymanski, Przemysław Kazienko, Radosław Michalski, and Piotr Bródka. Probing limits of information spread with sequential seeding. *Scientific reports*, 8(1):1–9, 2018.

[27] Kyomin Jung, Wooram Heo, and Wei Chen. Irie: Scalable and robust influence maximization in social networks. In *Data Mining*

(ICDM), 2012 IEEE 12th International Conference on, pages 918–923. IEEE, 2012.
[28] Elihu Katz and Paul Lazarsfeld. Personal influence, 1955.
[29] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.
[30] William O Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. In *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*, volume 115, pages 700–721. The Royal Society, 1927.
[31] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.
[32] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1852–1872, 2018.
[33] Shuyang Lin, Qingbo Hu, Fengjiao Wang, and S Yu Philip. Steering information diffusion dynamically against user attention limitation. In *2014 IEEE International Conference on Data Mining*, pages 330–339. IEEE, 2014.
[34] Vijay Mahajan, Eitan Muller, and Frank M Bass. New product diffusion models in marketing: A review and directions for research. In *Diffusion of technologies and social behavior*, pages 125–177. Springer, 1991.
[35] Radosław Michalski, Jarosław Jankowski, and Piotr Bródka. Effective influence spreading in temporal networks with sequential seeding. *IEEE Access*, 8:151208–151218, 2020.
[36] Stanley Milgram. Behavioral study of obedience. *The Journal of abnormal and social psychology*, 67(4):371, 1963.
[37] Mark Newman. Networks: an introduction. *United Slates: Oxford University Press Inc., New York*, pages 1–2, 2010.
[38] Chengzhang Ni, Jun Yang, and Demei Kong. Sequential seeding strategy for social influence diffusion with improved entropy-based centrality. *Physica A: Statistical Mechanics and its Applications*, 545:123659, 2020.
[39] Yaodong Ni. Sequential seeding to optimize influence diffusion in a social network. *Applied Soft Computing*, 56:730–737, 2017.
[40] J-P Onnela, Jari Saramäki, Jorkki Hyvönen, György Szabó, David Lazer, Kimmo Kaski, János Kertész, and A-L Barabási. Structure and tie strengths in mobile communication networks. *Proceedings of the national academy of sciences*, 104(18):7332–7336, 2007.
[41] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
[42] Sancheng Peng, Yongmei Zhou, Lihong Cao, Shui Yu, Jianwei Niu, and Weijia Jia. Influence analysis in social networks: A survey. *Journal of Network and Computer Applications*, 106:17–32, 2018.
[43] Lior Seeman and Yaron Singer. Adaptive seeding in social networks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 459–468. IEEE, 2013.
[44] Alon Sela, Irad Ben-Gal, Alex Sandy Pentland, and Erez Shmueli. Improving information spread through a scheduled seeding approach. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 629–632. ACM, 2015.
[45] Alon Sela, Dmitri Goldenberg, Irad Ben-Gal, and Erez Shmueli. Active viral marketing: Incorporating continuous active seeding efforts into the diffusion model. *Expert Systems with Applications*, 107:45–60, 2018.
[46] Paulo Shakarian, Sean Eyre, and Damon Paulo. A scalable heuristic for viral marketing under the tipping model. *Social Network Analysis and Mining*, 3(4):1225–1248, 2013.
[47] Guangmo Tong, Weili Wu, Shaojie Tang, and Ding-Zhu Du. Adaptive influence maximization in dynamic social networks. *IEEE/ACM Transactions on Networking (TON)*, 25(1):112–125, 2017.
[48] Thomas W Valente. Network interventions. *Science*, 337(6090):49–53, 2012.
[49] Philip Zimbardo. The lucifer effect-understanding how good people turn evil, 2007.

Tomer Lev received a B.Sc. in 2019 and an M.Sc. in 2020 (under the supervision of Dr. Erez Shmueli), both in Industrial Engineering from Tel Aviv University. His professional experience in recent years includes being a teacher assistant, a data analyst, and most recently, a data scientist. His research interests focus on social networks analysis and data science.

Prof. Irad Ben-Gal is the head of LAMBDA: The Laboratory of AI & Machine Learning Business and Data Analytics at Tel Aviv University. Prof. Ben-Gal wrote four books, published more than 150 scientific papers and patents, supervised dozens of graduate students and received numerous awards for his work. He held a visiting professor position at Stanford University, teaching graduate courses in analytics and is currently co-heading the TAU/Stanford "Digital Living 2030" research initiative. Irad is a world-renowned expert in machine learning, data science and predictive analytics with more than 25 years of experience in the field, including close R&D collaborations with companies such as Oracle, Intel, GM, AT&T, Procter & Gamble, Applied Materials and Nokia. Irad is a co-founder and the chairman of CB4 ("See Before"), a startup backed by Sequoia Capital that provides AI solutions to retail organizations. He is an advisory board member in several startup companies that focus on AI applications, helping them to develop new AI solutions.

Dr. Erez Shmueli is a senior lecturer and the head of the Big Data Lab at the department of Industrial Engineering at Tel-Aviv University and a research affiliate at the MIT Media Lab. He received his BA degree (with honors) in Computer Science from the Open University of Israel and MSc and PhD degrees in Information Systems Engineering from Ben-Gurion University of the Negev and spent two years as a post-doctoral associate at the MIT Media Lab. His research interests include Big Data, Complex Networks, Computational Social Science, Machine Learning, Recommender Systems, Database Systems, Information Security and Privacy. His professional experience includes being a programmer and a team leader in the Israeli Air-Force and, a project manager in Deutsche Telekom Laboratories at Ben-Gurion University of the Negev, a co-founder of two startups (Babator and SafeMode), and a consultant (among the rest to Microsoft and the municipality of Ashdod).