Temporal Cascade Model for Analyzing Spread in Evolving Networks

APARAJITA HALDAR, University of Warwick, UK
SHUANG WANG, University of Warwick, UK
GUNDUZ VEHBI DEMIRCI*, Imagination Technologies, UK
JOE OAKLEY, University of Warwick, UK
HAKAN FERHATOSMANOGLU†, University of Warwick, UK

Current approaches for modeling propagation in networks (e.g., of diseases, computer viruses, rumors) cannot adequately capture temporal properties such as order/duration of evolving connections or dynamic likelihoods of propagation along connections. Temporal models on evolving networks are crucial in applications that need to analyze dynamic spread. For example, a disease spreading virus has varying transmissibility based on interactions between individuals occurring with different frequency, proximity, and venue population density. Similarly, propagation of information having a limited active period, such as rumors, depends on the temporal dynamics of social interactions. To capture such behaviors, we first develop the *Temporal Independent Cascade (T-IC)* model with a spread function that efficiently utilizes a hypergraph-based sampling strategy and dynamic propagation probabilities. We prove this function to be submodular, with guarantees of approximation quality. This enables scalable analysis on highly granular temporal networks where other models struggle, such as when the spread across connections exhibits arbitrary temporally evolving patterns. We then introduce the notion of 'reverse spread' using the proposed T-IC processes, and develop novel solutions to identify both sentinel/detector nodes and highly susceptible nodes. Extensive analysis on real-world datasets shows that the proposed approach significantly outperforms the alternatives in modeling both *if* and *how* spread occurs, by considering evolving network topology alongside granular contact/interaction information. Our approach has numerous applications, such as virus/rumor/influence tracking. Utilizing T-IC, we explore vital challenges of monitoring the impact of various intervention strategies over real spatio-temporal contact networks where we show our approach to be highly effective.

CCS Concepts: • **Applied computing**; • **Mathematics of computing → Graph theory**; • **Computing methodologies → Modeling and simulation**;

Additional Key Words and Phrases: temporal cascade model, reverse spread maximization, dynamic spread analysis, spatio-temporal contact networks, sentinel nodes, susceptible nodes, efficiency, scalability

ACM Reference Format:
Aparajita Haldar, Shuang Wang, Gunduz Vehbi Demirci, Joe Oakley, and Hakan Ferhatosmanoglu. 2023. Temporal Cascade Model for Analyzing Spread in Evolving Networks. *ACM Trans. Spatial Algorithms Syst.* 1, 1, Article 1 (January 2023), 29 pages. https://doi.org/10.1145/3579996

*Previously at the University of Warwick. This publication describes work performed at the University of Warwick and is not associated with Imagination Technologies.
†Also with Amazon Web Services. This publication describes work performed at the University of Warwick and is not associated with Amazon.

Authors' addresses: Aparajita Haldar, University of Warwick, Coventry, UK; Shuang Wang, University of Warwick, Coventry, UK; Gunduz Vehbi Demirci, Imagination Technologies, UK; Joe Oakley, University of Warwick, Coventry, UK; Hakan Ferhatosmanoglu, University of Warwick, Coventry, UK.

Manuscript submitted to ACM

1 INTRODUCTION

Research on modeling cascades in networks has traditionally mostly focused on identifying the influential nodes in social networks [14]. In this paper, we introduce a *Temporal Independent Cascade (T-IC)* model that can simulate spread through evolving networks to identify i) *sentinel nodes* and ii) *susceptible nodes*. The proposed setting has many potential applications, ranging from identifying targets susceptible to rumors or misinformation, to detecting disease outbreaks and analyzing intervention strategies. Within a single model, we aim to address several challenges, namely: sentinel/susceptible solution sets, evolving connectivity patterns, dynamic propagation patterns, granular data, and approximation guarantees. Existing solutions do not handle these problems together effectively.

i) **Sentinel/susceptible solution sets** - While diffusion models have predominantly been considered for influence maximization and identifying highly influential nodes, we consider how they can identify nodes that i) detect activation anywhere in the network (sentinel nodes) or ii) collect activation from anywhere in the network (susceptible nodes). A set of sentinel (or detector) nodes is one that provides the best coverage over the entire network. That is, spread processes taking place anywhere in the network are likely to reach (and be detected by) one of these sentinels. Identifying such a sentinel set involves an optimization objective that is different from that of influence maximization, and requires a 'reverse spread' process. Some previous works have explored the identification of sentinel nodes in epidemiological settings [24, 48]. However, such studies are limited to susceptible-infectious-recovered (SIR) compartmental models to obtain mathematical results for infectious diseases. A generalized diffusion model is more broadly applicable by adjusting the model parameters to simulate different real-world settings (e.g., tipping points for social collective behavior, effects of computer virus replication) [49]. Another objective is the identification of high priority susceptible nodes, i.e., those that are independently most likely to collect activation (disregarding the overall network coverage).

ii) **Evolving connectivity patterns** - The spread model must cater for both the addition/deletion of edges and varying contact frequencies. For example, connectivity patterns such as the interactions between infected individuals over time [32], or the usage of removable storage devices in a computer network affected by a computer virus [60], can govern how the spread takes place. Most cascade models that operate on such evolving connectivity patterns (and preserve approximation guarantees on the solutions) often use static snapshots at regular intervals to represent the dynamic nature of the network. Rather than using partially observed connections from an incomplete static snapshot to produce solutions that are more likely to be sub-optimal [41], there is a need for a solution set that is optimized across the entire window of time-varying contact events. Furthermore, it is desirable for the spread model to capture temporal dependencies at every step.

iii) **Dynamic propagation patterns** - To our knowledge, there is currently no comprehensive solution with provable quality guarantees to handle dynamic propagation at different rates, where the likelihood of the spread of activation along connections can exhibit arbitrary temporally evolving patterns. Such a model is important to identify necessary solution sets of nodes and design appropriate intervention strategies for various real-world settings. For example, the bounded infectious period of a disease-causing virus [32] can be captured by a dynamic propagation rate that tapers down to 0 after a defined duration of time. Similarly, the limited period of effectiveness of a rumor [18] can be modeled, as can the more long-term effects on a machine infected with a computer virus [60].

iv) **Granular data** - The cascade model should incorporate highly granular data and any available exogenous features. For example, information about individual contacts (rather than only aggregated mobility data), and about venue type and popularity are crucial for disease monitoring and the design of intervention strategies. Information diffusion networks and cascade models are well-suited for such data and applications. While there is growing interest

in generating contact networks using mobile data, network-based cascade models rarely use such fine-grained spatio-temporal data. To our knowledge, existing temporal network epidemiology solutions that do model the dynamics of contagion in time-varying networks [27, 48] are mostly based on SIR compartmental models. More general cascade models may be designed for varied application domains (e.g., misinformation campaigns in social networks, computer viruses infecting networked machines, disease epidemics in population contact networks). Similar models have erstwhile been studied mostly for information diffusion problems without the spatio-temporal element (e.g., viral marketing campaigns in social networks).

v) **Approximation guarantees** - Prior IC based models lose the approximation guarantees on their solutions when modified to support time windows or graph snapshots. To allow for scalable analysis that can efficiently support the use of fine-grained, temporally evolving graph data, rigorous approximation guarantees must be maintained in the temporal spread modeling algorithm. Such considerations enable the use of the solution for large-scale networks even with granular temporal features.

Towards addressing the above challenges, we introduce the *Temporal Independent Cascade (T-IC)* model for detection and analysis of spread, e.g., disease outbreaks, misinformation campaigns. The model includes a novel spread function that utilizes dynamic propagation probabilities for every edge in the network. We prove this to be submodular with approximation guarantees within $1-\frac{1}{e}$ of optimal, thus preserving solution quality. We introduce two distinct objectives and associated T-IC based solutions in this context: i) finding sentinel nodes, and ii) finding susceptible nodes. We illustrate the application of our approach for monitoring spread in evolving networks, to i) detect *if* there is any spread by checking a limited number of nodes, and ii) understand *how* it is likely to spread by identifying the most susceptible nodes. Our examples of application-driven propagation probability functions are derived from real disease/influence spread characteristics. We illustrate the approach on real-world location-based networks, pandemic datasets, and social networks. We further analyze a range of intervention strategies towards containing an outbreak using the T-IC model, e.g., targeted shutdown of venues (or websites) to slow down the disease spread (or cyber attack).

Our approach involves three stages. We first formalize the evolving network where edge connections may be added/deleted and the propagation probability between two nodes can vary over time. An example for this is a spatio-temporal contact network, e.g., meetings events between individuals within a population with a varying likelihood of transmission that is based on contact duration, proximity, population density, and other customizable factors determined by domain experts. A similar temporal setting can be considered for other applications, such as the spread of computer viruses through networked machines, or the spread of online misinformation between social media user accounts.

Second, the *Temporal Independent Cascade (T-IC)* model is defined over the network, which handles dynamic propagation rates and allows an active node to repeatedly try to activate its neighbors. That is, while spread may continue to take place as long as a node is active, the actual likelihood is determined by the propagation rate (e.g., it may drop to zero quickly depending on the infectious period of a virus [32], or based on the short-term nature of rumors [18]). This is an enhancement of the well-known IC model, as it takes the temporal ordering of activations into account and allows for dynamic changes in both propagation probability and network connectivity patterns.

Third, we define spread functions to address the objectives of finding sentinel nodes and finding susceptible nodes. We propose an efficient hypergraph-based sampling strategy to capture evolving connections and dynamic propagation rates in the network. We then use T-IC to generate 'random reachable sets' that can reflect the patterns of spread within this evolving network. These random reachable sets are used to develop two solutions, *Reverse Spread Maximization (RSM)* and *Expected Spread Maximization (ESM)*, that we use to identify sentinel and susceptible nodes respectively, with provable approximation guarantees for the optimality of these solution sets. We employ RSM to find sentinel nodes, i.e.,

a set of nodes where at least one is likely to be activated regardless of where the spread begins. RSM is useful to detect *if* there is spread within a population. An application is "sentinel surveillance" and the early detection of outbreaks by using sensors at the set of sentinels [3, 12, 23]. To understand *how* the spread is likely to take place, we also study ESM to identify susceptible nodes, i.e., a set of nodes that accumulates the most spread from arbitrary seeds. A susceptible set represents all individuals most liable to be part of the spread.

Finally, both approaches are applicable to a number of mitigation or intervention strategies. Our RSM and ESM solutions both rely on measuring the reverse reachability through the network, with each having a different optimization objective. Sentinels offer the maximum coverage over the network using a set of nodes of minimal size, thereby serving as detectors that are collectively likely to catch any spread. This could involve selective testing of individuals for a disease, detecting for malicious data used in attacks on communication networks, signaling wireless network failures, or similar preemptive actions. Susceptible nodes, on the other hand, are all independently most likely to catch this spread, making them all at high-risk, but may not cover the network well. This can be useful for contact tracing the spread of disease, detecting failure points in communication networks, or identifying the most sensitive targets of misinformation campaigns. It is important to note that merely ranking the nodes by importance or ability to spread activation (i.e., via influence maximization methods) are insufficient for these two distinct problems.

The main technical contributions of the paper are as follows:

- A temporal independent cascade model (T-IC) for networks with dynamic propagation rates is introduced, along with a 'reverse spread' function which is submodular. These features allow T-IC to provide formal approximation guarantees on its solutions. While active nodes can still repeatedly spread activation, a dynamic propagation rate (which can rise, fall, or drop to zero) is what dictates the extent of this spread.
- A sampling strategy based on hypergraph construction is proposed to handle evolving connections in the network, and dynamic spread patterns are reflected in 'random reachable sets' that form hyperedges. This helps to choose the optimal solution sets regardless of where the spread first begins, while also incorporating temporal dependencies into the sampling step.
- Two solutions are devised with different objectives: RSM for selecting sentinel nodes (e.g., individuals/computers to periodically test for disease/virus) to detect *if* there is any spread, and ESM for identifying susceptible nodes (e.g., individuals/computers to treat/fix and whose connections may need to be traced) to capture *how* the spread occurs.
- Evaluations using granular real-world datasets (location, contact, and social networks) confirm that the proposed solutions are effective in identifying i) a sentinel set with highest 'reverse spread' coverage and success of detecting spread, and ii) a susceptible set with highest likelihood of activated nodes included based on the 'expected spread' pattern. A further case study of possible intervention, mitigation, and categorical analysis strategies is presented for disease monitoring and social applications. Analogues of these ideas can be applied to other settings discussed in our examples, such as analysis of the spread of computer viruses or social network misinformation.

2 RELATED WORK

There is extensive work in the areas of information diffusion and influence maximization, especially for social networks [9, 10, 14, 20, 35, 53, 56]. Most prior work on evolving networks typically focuses on maximizing influence [16, 21, 41, 52, 59] as opposed to the objectives that we study. To our knowledge, there is no prior work on capturing sentinel and susceptible sets in temporal cascade networks as studied in this paper. Under the widely used independent cascade and linear threshold propagation models, finding a subset of users that maximizes the expected spread is shown to be NP-Hard [28]. There are studies using time constraints within IC models, which use a constant probability of spread or

ignore repeated activations from an active node [8, 30, 36]. In contrast, our approach exposes temporal factors in the continuous activation process and allows customizable formulations of propagation rates. Allowing nodes to remain active enables us to provide approximation algorithms with quality guarantees. Simultaneously, our temporally-guided sampling policy and use of dynamic propagation rates along edges ensures that the algorithm realistically models the spread of activations. The proposed T-IC model can be applied in a variety of settings, including the identification of misinformation campaigns or rumors propagating in social networks, monitoring of disease outbreaks in contact networks, and tracing of computer viruses replicating through networked machines.

The selection of sentinel nodes is an important task in many applications (on static networks) where early detection of activation is beneficial, such as monitoring disease outbreaks [3], signaling wireless sensor network failures [13] and detecting malicious data transmissions [51]. Identification of a minimal solution set of sentinel nodes is of interest as resources are often constrained (e.g. medical tests, wireless sensor costs). The dynamics of spread over an evolving network topology introduce novel challenges in the selection of sentinels.

Susceptible nodes are also studied in the context of static networks. For example, failure points in communication networks may be triggered by the cascading failure and redistribution of data packets, and have been modeled stochastically for mitigation purposes [45]. Other studies on fake news detection find that recognizing easily influenced users is more important to control the spread, rather than the influential "persuaders" [50]. Therefore, our temporal model can be applied to solve pressing problems across a variety of domains.

Temporal networks have been studied for varied applications. For example, rumor propagation is affected by the dynamics and durations of contacts [18], while computer viruses stabilize or die out depending on dynamic interactions with removable devices [60]. All of these are domains where our T-IC model finds potential application, with a perspective of tracing 'reverse spread' to identify sentinel/susceptible nodes. That is, we focus on temporal networks where there may be an opportunity to prepare intermittent tests on sentinels, or to quickly mitigate the undesirable effects at susceptible nodes.

Location-based social network datasets that we cover in our evaluations, e.g., Foursquare [58] and SafeGraph [47] have been widely used in recent data-driven applications such as disease monitoring [5, 7, 55, 57] and urban/traffic planning [26, 40, 46]. Our approach enables fine-grained analysis on such data in any specific propagation scenario (e.g., with varying population density or varying proximity between interacting individuals). We illustrate this using data sources ranging from social networks to statistics collected for analyzing pandemic spread, e.g., the BBC Pandemic app [32] and Italy mobility studies [42]. We also utilize Foursquare trajectories and generate detailed trajectories from the SafeGraph data, in lieu of their aggregated versions, as we support a more granular approach to spread modeling.

Among various spread models, there is extensive epidemiological research based on the SIR framework [29] and its extensions, where differential equations govern the transition rates between Susceptible, Infectious, and Recovered stages. This framework has led to numerous spread models being designed that handle datasets at a low-granularity aggregated level (e.g., district-level infection case counts). One such approach deals with greedy immunization strategies restricted to local behavior such as node degrees [43]. SIR is also used to study the impacts of static and temporal network structures on outbreak size, sentinel surveillance, and vaccination objectives [23, 24].

The aforementioned equation-based compartmental models in the domain operate on aggregate data and assume homogeneous mixing within the population without considering temporally ordered meeting events. SIR models are powerful in the production of analytical and numerical results but do not properly simulate the real-world dynamics of propagation rates between specific nodes [25]. Instead, our solution aims to offer highly granular and efficient predictive models, with the goal of identifying critical subsets (i.e. sentinel and susceptible nodes) in the contact network. A

recent Hawkes-process based variation of SIR has been proposed by performing agent-based infection simulations to assign COVID-19 risk scores to individuals [44]. Our approach has orthogonal perspectives where we use our assigned transmission risk scores to determine optimal solution sets that are either sentinels or susceptible individuals.

The SIR framework has also been applied to other domains, such as computer viruses and malware. A numerical study on the spread of malware through malicious hyperlinks on the web adds a component to address the temporal nature of such malware [37]. However, such work suffers the same deficiencies of not making full use of the network topology information. In our work, we evaluate our spread model and solution sets on various temporal networks, as well as demonstrate different interventions that can help combat harmful or malicious spread. For example, T-IC is able to utilize individual-level location sequences to trace the disease spread spatially, resulting in support for more use cases such as "backward contact tracing" (using knowledge of active nodes to trace initial seeds), which has attracted attention in the context of COVID-19 [15]. The need for dynamic network analysis for forward and backward influence tracking has been highlighted in the literature [1]. Similarly, identifying susceptible nodes in social networks can help in combating fake news, because by ensuring that the users who are susceptible to fake news are also exposed to real news, they are less likely to believe the falsehoods [50].

3 PROBLEM DEFINITION

We now present the Temporal Independent Cascade (T-IC) model, and define our optimization objectives, for tracing the spread over an evolving network with dynamic propagation probabilities.

3.1 Temporal network

In a standard IC model, information flows/diffuses/propagates through the network via a series of cascades. Nodes may either be active (already influenced by the information that is propagating through the network) or inactive (either unaware of the information that is propagating but has not reached it by this point, or not influenced by the propagation that did reach it). The standard IC model assumes a static probability distribution over a static graph structure. This IC process is simulated over a graph $G=(V,E,p)$ where each edge $(u,v) \in E$ is associated with a constant probability function $p\colon E \mapsto [0,1]$, reflecting the likelihood of activation when nodes $u \in V$ and $v \in V$ have a common edge (e.g., a common meeting point in the location histories of two individuals). Propagation starts from an initial seed set in V (the only nodes active at step 0). Propagation takes place in discrete steps with each active node u during step i being given a single chance to activate its currently inactive neighbor v with some probability $p(u,v)$. That is, at every step $i \geq 1$, any node made active in step $i-1$ has a single chance to activate any one of its inactive neighbors. The process continues with nodes remaining active once activated, until no further propagation is possible. Therefore, this is a stochastic process that requires a large number of simulations to accurately determine the spread of information.

Since the standard IC model uses a static probability distribution over a static network, it is insufficient to handle evolving graphs with changing propagation rates. Both the structure of G and the propagation rates can change dynamically. For example, the spread of rumors depends on the dynamics as well as the duration of contacts between individuals, and is guided by the specific short-lived nature of gossip or fake news. Similarly, a disease spread model needs to consider the order and duration of interactions within a population, and the varying infectivity of a virus over its lifetime. Therefore, we define a new temporal IC model for a temporal network.

Definition 1 (Temporal Network). *Given a discrete time domain $T=\{1,2,...,n\}$, a temporal network is a graph $G = (V,E,P(t))$ where for each time interval $t \in T$, edge set E is associated with a different propagation probability distribution $P(t)\colon E \mapsto [0,1]$. That is, each edge $(u,v) \in E$ has n probabilities $p(u,v,t)$, one for each $t \in T$.*

Since u and v may be linked multiple times in T, the corresponding $p(u, v, t)$ for every interval t needs to be separately maintained. An evolving graph is represented by adding all edges and assigning $p(u, v, t) = 0$ when there is no (u, v) connection in interval t. Moreover, a rigorous formulation for $p(u, v, t)$ is needed to describe more complicated cases of spread, which we discuss in Section 5.1.2.

3.2 Temporal Independent Cascade model

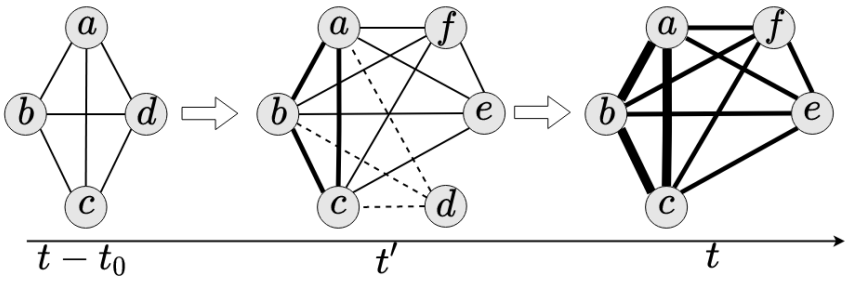
To model the spread of activation in the temporal graph G, we introduce the Temporal Independent Cascade (T-IC) model as an enhancement of the popularly used standard IC model for our setting. Given time intervals $i, j \in T$ such that $i < j$, the T-IC model proceeds from i to j as follows: Let A_t denote the set of initially active nodes at the beginning of time interval t. Within each interval $t \in [i, j]$, the standard IC model is executed once under probability distribution $P(t)$ on edges. That is, the active nodes for that particular interval activate their neighbors based on the propagation rates associated with the edges for that time interval only. This proceeds until no further spread is possible. Note that this $p(u, v, t)$ varies, and can fall to 0 to indicate that spread does not take place at the given t. The set of all activated nodes at the end of interval t is A_{t+1}, which thus also represents the active nodes at the beginning of the next time interval $t+1$ (active nodes remain active in subsequent intervals). This process is executed for each interval $i, i+1, \ldots, j$ and the final set of active nodes A_{j+1} is obtained after interval j.

In other words, we do not modify the standard IC process, but instead run it to completion independently within each discrete time interval $t \in [i, j]$ under the corresponding probability distribution defined over edges. For the entire chosen time window ($[i, j] \in T$), stacking several of these IC processes, and treating activated nodes as the seeds for the next run, allows us to mimic the spread over an evolving graph without sacrificing the approximation guarantees. Furthermore, if a node is activated in a specific time interval, it can continue to activate its neighbors in subsequent time intervals, but subject to the propagation probability distribution. For example, a person infected with a virus may continue to spread infection during interactions with people for as long as they are contagious (e.g., around 14 days for COVID-19 [4]). This continuous activation during the T-IC process, along with the propagation probability formulations, makes it possible to reflect real-world spreading phenomena (e.g., for disease monitoring) with our model.

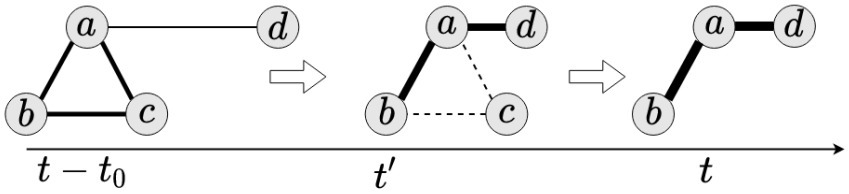
Figure 1 illustrates the impact of temporal order and dynamic connections in the spread model. Suppose node c is initially active. In the first case (Figure 1(a)), nodes a and b have higher likelihood of activation than nodes that arrive later, such as e or f, or nodes that leave, such as d. The greater chance of activation due to the prolonged duration of contact with the active node c, as well as the increased risk of activation due to the high density of nodes, is reflected in the thicker edge connections to a and b. In the second case (Figure 1(b)), a and b are again more susceptible due to their proximity to c, because c leaves before d approaches closer. The risk of activation for d then increases as the node approaches closer to the other nodes that may have been activated by c. We develop a propagation probability assignment that can capture all these factors for different application settings, which we describe in Section 5.1.2.

3.3 Optimization objectives

Our first objective is to identify the sentinel nodes, i.e., the set of nodes that maximizes the probability of detecting any activations in a network where the set of initially active nodes is not known. A practical application is to detect an outbreak using minimal resources (e.g., medical tests, computer virus checks, social media reviews). Testing a targeted set of individuals can be an efficient way to detect the onset of spread within a population before it is widespread, akin to the concept of "sentinel surveillance" [23]. Therefore, our optimization objectives focus on finding k-element subsets of sentinel nodes. The proposed solution can also be run until termination over an indefinite time window, but we note



(a) Propagation probability increases with population density and duration



(b) Propagation probability increases with proximity and duration

Fig. 1. Dynamic propagation probabilities in the network reflect temporal characteristics. Edge connection indicates proximity, with thicker edge for higher probability (based on population density, proximity, and duration of contact) and dotted edge for latent probability after deletion.

that when there is a temporal constraint within which to detect outbreaks, a truncated solution set (choosing only k nodes) can sufficiently cover the entire relevant spread.

The objective is to maximize the probability that at least one node in a k-element solution set S becomes active after a random T-IC process (i.e., one that starts with randomly selected seeds) within a given time window $[i, j]$. We note that the 'temporal reverse spread maximization' objective, as defined below, corresponds to this goal of identifying a k-element set of sentinel nodes. Optimizing the success rate of detection of spread anywhere in the network by using sentinels can be achieved by maximizing the expected amount of 'reverse spread' $\phi(\cdot)$. Expected reverse spread can be defined as the expected number of nodes that can spread activation to the nodes in S. Therefore the expected reverse spread of set S on G within $[i, j]$, denoted by $\phi_{ij}(G, S)$, is the expected number of nodes that can activate at least one node in set S during a random T-IC process in $[i, j]$. We discuss how to compute $\phi(\cdot)$ in Section 4. The problem of maximizing the expected reverse spread $\phi_{ij}(G, S)$ can be formally defined as follows:

Definition 2 (Temporal Reverse Spread Maximization). *Find the k-element subset of nodes $S^* \subset V$ such that*

$$S^* = \underset{S \subset V, |S|=k}{\arg\max}\ \phi_{ij}(G, S) \tag{1}$$

Definition 2 addresses the problem of detecting *if* there is any spread. To understand *how* the spread takes place within the network, we need to identify high priority nodes that collect activation, i.e., nodes that are the most susceptible

to activation. For example, in disease, computer virus, or misinformation monitoring applications, the objective is to identify the subset S that are all highly likely to be infected. These offer important candidates to immunize, disconnect, or re-educate in order to mitigate the spread.

Hence, we next aim to identify the k-element subset S containing the maximum expected number of active nodes after T-IC process in $[i, j]$. Similarly to the temporal reverse spread maximization objective, the 'temporal expected spread maximization' objective defined below corresponds to the goal of identifying a k-element set of susceptible nodes. This second problem is formally defined as follows:

Definition 3 (Temporal Expected Spread Maximization). *Let $I(s)$ be an indicator random variable for node $s \in S$ such that*

$$I(s) = \begin{cases} 1 & \text{if } s \text{ is activated} \\ 0 & \text{if } s \text{ is not activated} \end{cases} \tag{2}$$

after executing random T-IC process. Find a k-element subset of nodes $S^* \subset V$ that maximizes the expected number of active nodes in S^*

$$S^* = \underset{S \subset V, |S|=k}{\arg\max} \; \mathbb{E}\left[\sum_{s \in S} I(s)\right] \tag{3}$$

In the next section, we describe our solutions to address the above objectives, namely *Reverse Spread Maximization* (RSM) and *Expected Spread Maximization* (ESM).

4 REACHABLE SET SAMPLING BASED ALGORITHM

We introduce an extension to the well-defined sampling approach for hypergraph construction that allows us to preserve optimality guarantees in our dynamic setting. We show that the defined reverse spread function $\phi(\cdot)$ is submodular in Theorem 4.1. It follows that the standard hill-climbing greedy algorithm achieves $1-\frac{1}{e}$-approximation guarantee, i.e., a solution that uses this function can be approximated to within $1-\frac{1}{e}$ of optimal [28].

Theorem 4.1. *Under the T-IC model, function $\phi_{ij}(\cdot)$ is submodular.*

Proof. Let $G = (V, E, p)$ be a directed graph where each edge $(u, v) \in E$ is associated with a weight $p(u, v)$ denoting the probability that spread occurs from u to v. Kempe et al. [28] showed that the IC model induces a distribution over graph G, such that a directed graph $g = (V, E')$ can be generated from G by independently realizing each edge $(u, v) \in E$ with probability $p(u, v)$ in E'. In a realized graph $g \sim G$, nodes reachable by a directed path from a node u are its reachable set $R(u, g)$, and correspond to the nodes activated in one instance of the IC process with u as the initially active seed node. They proved that for $S \subset V$, the spread function $\sigma(S, g) = |\cup_{u \in S} R(u, g)|$ is submodular.

Similarly, the T-IC model induces a distribution over $G = (V, E, P(t))$, where the IC model is executed independently in each discrete time interval $t \in [i, j]$ under the corresponding probability distribution defined over edges. Additionally, an activated node remains active in subsequent intervals, getting multiple chances to activate its neighbors. So, directed graph $g_{ij} = (V, E')$ can be generated as follows: For intervals $t = i, i+1, \ldots, j$, each edge $(u, v) \in E$ is realized in E' with probability $p(u, v, t)$, only if node u is active at the beginning of interval t. Hence, the reachable set $R(s, g_{ij})$ corresponding to a node s on the generated graph g_{ij} consists of all the nodes that are reachable and activated by time interval j by the seed s that was initially active in time interval i.

Let g_{ij}^T denote the transpose of g_{ij}, obtained by reversing all its directed edges. Reachable set $R(r, g_{ij}^T)$ corresponds to all seed nodes that, if active in interval i, would have the ability to activate the receiving node r by time interval j. Given

a set of nodes S, let the reverse spread $\phi(S, g_{ij})$ denote the number of nodes that can reach some node in S. That is, $\phi(S, g_{ij}) = |\bigcup_{u \in S} R(u, g_{ij}^T)|$. Since $\phi(S, g_{ij}) = \sigma(S, g_{ij}^T)$, the submodularity of $\phi(S, g_{ij})$ follows. Therefore, the expected reverse spread $\phi_{ij}(G, S) = E(\phi(S, g_{ij}))$ is submodular, being a linear combination of submodular functions. □

Borgs et al. [6] employ a sampling strategy to build a hypergraph representation and estimate the spread of activation. We enhance this technique to handle dynamic propagation rates and identify solution sets for both our defined tasks (i.e., identifying sentinel nodes and susceptible nodes). Our algorithm and sampling strategy use a novel process of generating the hypergraph to encode the reverse spread of any given subset of nodes via its nets. A hypergraph is a generalization of a graph in which two/more nodes (pins) may be connected by a hyperedge (net). The two-step sampling strategy is as follows: i) we execute random T-IC processes (that start with random active seeds) on the temporal network, and ii) for each execution of a T-IC process, we construct a net whose pins are the nodes that are activated during the process.

As shown in Theorem 4.1, g_{ij} can be drawn from the distribution induced by a T-IC model on G. The edges in the graph g_{ij} are tried to be realized by traversing only live edges (i.e., edges where the starting node is already active). Due to this constraint of only considering live edges, this enforces a dynamic nature to the spread as it takes place by respecting the temporal ordering of connections. If a node v is reachable from many different nodes in g_{ij}, then it is more likely that this node will be activated by time interval j. Since any random seed in time interval i is equally likely to start the spread, the existence of more paths that lead to the node v results in a higher likelihood of its activation. This means that the reachable set of nodes $R(u, g)$ (i.e., all the nodes from the realized graph g_{ij} that are reachable by a directed path of edges from the node u), which depends on the random seed node u, is one among many possible sets of activated nodes at the end of a random T-IC process on the randomly sampled g_{ij}. Note that since traversal over live edges helps to capture temporal dependencies in the spread model, the identification of sentinel/susceptible nodes is a non-trivial and orthogonal problem that cannot be achieved through traditional influence maximization.

Overall, the solution depends on two levels of randomness that are encountered during the hypergraph construction: i) the sampling strategy for $g_{ij} \sim G$, and ii) the computation of $R(u, g_{ij})$ given a random seed u. The former depends on the probability distribution induced by the T-IC model over G, while the latter depends on the seed node u. We refer to such a reachable set that is generated by two levels of randomness as a 'random reachable set' $RR(u, g_{ij})$. Outbreaks are typically thought to start from a single source [22, 31]. Therefore, we consider one random seed node in our simulations on all datasets.

The main sampling step is repeatedly performed to build a hypergraph $H = (V, N)$ where each net $n_u \in N$ is independently generated by executing a random T-IC process from seed u. The hypergraph corresponds to a random reachable set $RR(u, g_{ij})$, i.e., $\text{pins}(n_u) = RR(u, g_{ij})$. The solution quality and concentration bounds thus depend on the number of nets generated to build the hypergraph [6].

Note that H and G are composed of the same set of nodes V. For a solution set S, the number of nodes sharing a net with at least one node in set S (which we refer to as $\deg(S)$ henceforth) corresponds to the number of times a node in S gets activated during the random T-IC processes executed to compute the random reachable sets. To select S as a collection of sentinel nodes, higher $\deg(S)$ will be more likely to detect spread in the network, which can be understood as follows: The degree of a node in the hypergraph is the sum of $|N|$ Bernoulli random variables [6]. This is because the inclusion of a node v in a random reachable set $RR(u, g_{ij})$ and in $\text{pins}(n_u)$ can be considered as a Bernoulli trial with success probability p_v, where p_v denotes the probability that v gets activated in a random T-IC process. That is, the hypergraph node degrees are binomially distributed with an expected value of $p_v \times |N|$. This implies that

$p_v = \mathbb{E}[deg(v)/|N|]$. Therefore, this node degree corresponds to the estimation of reverse spread of node v, since the reverse spread can be written as $\phi_{ij}(v) = |V| \times p_v$. Similar to the node degrees in hypergraph H, the expected value $\mathbb{E}[deg(S)/|N|]$ corresponds to the probability that *at least one* node in S gets activated during a random T-IC process. Therefore, the degree of a set S of nodes in hypergraph H, corresponds to the reverse spread $\phi_{ij}(S) = |V| \times \mathbb{E}[deg(S)/|N|]$, which can be estimated well if a sufficient number of nets are built.

We next describe two algorithms, RSM and ESM, to efficiently compute solution sets for the two tasks corresponding to Definitions 2 and 3 respectively.

4.1 Reverse Spread Maximization solution

In the hypergraph $H = (V, N)$, if a node connects many nets (i.e., its degree is high), then that node has a high probability of being activated during a random T-IC process. Similarly, if a set S of nodes covers many of the nets (random reachable sets), then its expected reverse spread $\phi_{ij}(G, S)$ is likely to be higher. In other words, there is a larger set of nodes that all have a chance to activate at least one node of S within the time window $[i, j]$.

As in the maximum coverage problem, we want to cover the maximum number of nets (elements) in the hypergraph H by choosing a solution set S of k nodes (subsets). This step is therefore equivalent to the well-known NP-Hard maximum coverage problem [54]. Borgs et al. [6] show that the maximum set cover computed by the greedy algorithm on the hypergraph yields $(1 - \frac{1}{e} - \epsilon)$-approximation guarantee for the influence maximization problem. Here, the parameter ϵ relates the approximation guarantee to the running time of the algorithm, and the solution quality improves with the increasing number of nets in the hypergraph.

Algorithm 1: RSM solution

```
   input: G = (V, E, P(t)), i, j, K, |N|
1  H = (V, N = ∅); S = ∅
2  for n = 1 to |N| do
3      Select source node s ∈ V uniformly at random; A = {s}
4      for t = i to j do
5          BFS_Q = A
6          while BFS_Q ≠ ∅ do
7              u = dequeue(BFS_Q)
8              foreach (u, v) ∈ E do
9                  Draw p ∈ [0, 1] uniformly at random
10                 if p ≤ p(u, v, t) and v ∉ A then
11                     A = A ∪ {v}; enqueue(BFS_Q, v)
12     N = N ∪ {A}
13 for k = 1 to K do
14     v_k = arg max_v deg_H(v); S = S ∪ {v_k}
15     Remove v_k and all of its incident nets from H
16 return S
```

Algorithm 1 displays the overall execution of the proposed solution. It generates a number of random reachable sets by first drawing a graph g_{ij} from the distribution induced by T-IC model on the input graph G and then performing a breadth-first search (BFS) starting from a randomly selected node u. This randomized BFS through time intervals proceeds such that the set of source nodes at each interval are the activated nodes in the preceding interval (lines

4–11). Thus each edge $(u, v) \in E$ is searched with probability $p(u, v, t)$ in time interval t. All nodes activated during a random BFS form a random reachable set and are connected by a net in hypergraph H (line 12). After generating the hypergraph H with $|N|$ nets, the algorithm repeatedly chooses the highest degree node at each iteration, adds it to the solution set, and subtracts this node together with all incident nets from the hypergraph. This is done repeatedly until a k-element subset of nodes, which is the resulting solution set S, is computed (lines 13–15). This algorithm generates a solution of sentinel nodes for Definition 2.

4.2 Expected Spread Maximization solution

In order to maximize the expected number of active nodes in S, all the nodes having the highest probability of being activated should be included in the solution set, since the expected number can be given as $\mathbb{E}\left[\sum_{s \in S} I(s)\right] = \sum_{s \in S} p_s$. Hence, the problem in Definition 3 can be solved by Algorithm 2. The final step (line 13) now selects the solution set as the k-element subset of nodes having the most incident nets (i.e. largest degrees in H).

Algorithm 2: ESM solution

```
   input: G = (V, E, P(t)), i, j, K, |N|
1  H = (V, N = ∅); S = ∅
2  for n = 1 to |N| do
3  |  Select source node s ∈ V uniformly at random; A = {s}
4  |  for t = i to j do
5  |  |  BFS_Q = A
6  |  |  while BFS_Q ≠ ∅ do
7  |  |  |  u = dequeue(BFS_Q)
8  |  |  |  foreach (u, v) ∈ E do
9  |  |  |  |  Draw p ∈ [0, 1] uniformly at random
10 |  |  |  |  if p ≤ p(u, v, t) and v ∉ A then
11 |  |  |  |  |  A = A ∪ {v}; enqueue(BFS_Q, v)
12 |  N = N ∪ {A}
13 S = arg max_{V' ⊂ V, |V'| = k}  Σ_{v ∈ V'} deg_H(v)
14 return S
```

As illustrated in Figure 3, we apply our sampling strategy on g_{ij}, allowing solutions for our novel objectives of finding sentinel/susceptible nodes. Clearly, the sampling step is dependent on temporal dynamics which change upon transposing g_{ij}, therefore solutions for our novel objectives are distinct from influence maximization approaches in literature. Specifically, the RSM approach can detect any outbreak efficiently, e.g., sentinel nodes b and e (or f) collectively correspond to greater coverage of the network than that offered by the ESM solution (nodes b and a). The ESM solution appears in RR-sets more frequently, and are nodes that are all highly likely to be activated. Such sentinel nodes are relevant in settings such as disease monitoring (for contact tracing efforts), or computer virus tracking (to identify and restore infected machines quickly). Identifying susceptible nodes in a social network also has interesting use cases such as combating rumor spread and misinformation campaigns more effectively.

5 PERFORMANCE EVALUATION

For each real dataset, we build a temporal network on which we execute the T-IC process, and evaluate the solutions in terms of identifying sentinel nodes and susceptible nodes as defined in Section 3. The algorithms are executed on an

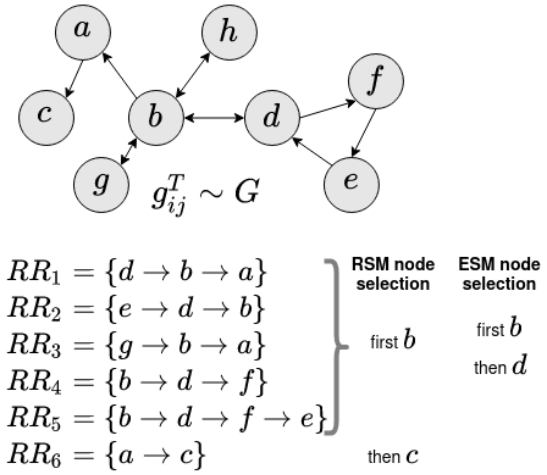


Fig. 2. RSM vs ESM on g_{ij}^T.

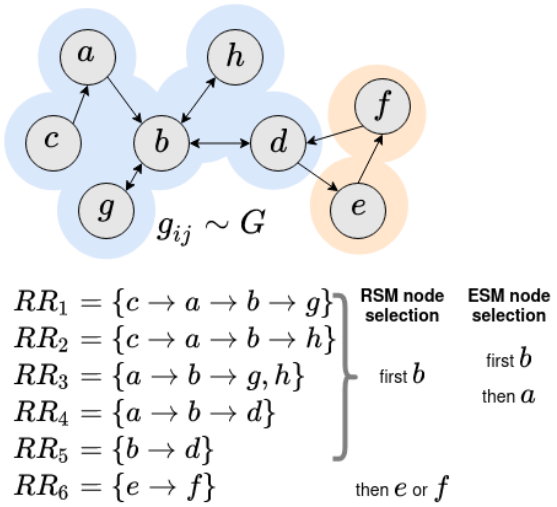


Fig. 3. RSM vs ESM on g_{ij}.

Ubuntu 20.04 machine with 16 Intel 3.90 GHz CPUs and 503 GB RAM. The code and data used are publicly available at: https://github.com/publiccoderepo/T-IC-model

5.1 Setup

5.1.1 Datasets:

We used eight real datasets to build temporal networks. Our T-IC model can be used to analyze a variety of application settings, such as monitoring disease or malware spread, countering misinformation campaigns, and tracking social

influence. For an example use case of disease monitoring, we use six location-based networks. The first two are spatio-temporal network datasets that we construct using the locations (check-ins) of Foursquare users, in line with other research studies on disease monitoring applications [5, 57]. We refer to them as **NYC** and **Tokyo** datasets. The NYC and Tokyo datasets [58] record check-in times, (anonymized) user IDs, venue IDs, venue locations, and venue categories. The temporal and spatial information from these are used to build edge connections in the network of users, selecting 25 consecutive days' data. To alleviate sparsity, the nodes for all users visiting the same venue in the same day are connected bidirectionally. Similarly, the third dataset **SP-office** [17] taken from SocioPatterns[1] contains a temporal network of contacts between individuals in an office building, where active contacts are recorded at 20-second intervals. We consider a small interval of 6 hours to determine whether interactions take place between individuals, based on whether their active locations are the same. Since all the location data is collected from 12 departments within the same workplace, using a larger interval would result in a fully connected contact network. Information about departments is provided which is similar to venue categories in our first two datasets. We consider the first 8 consecutive days to construct the temporal network.

Our fourth location dataset is based on SafeGraph [47], which has been used to analyze mobility patterns for COVID-19 mitigation [7]. SafeGraph contains POIs, category information, opening times, as well as aggregate mobility patterns such as the number of visits by day/hour and duration of visits. Using these mobility patterns, we generate synthetic trajectories for $2K$ individuals visiting 100 unique POIs in the NYC area over 25 days. To build an individual's trajectory, for each day of the week for three consecutive weeks, we select and assign sequential visit locations to appropriate timestamps (based on travel time and visit duration) as follows: i) each individual receives a random start timestamp for travel, a random start POI location, and a random trajectory length that determines the number of POIs to visit, ii) SafeGraph dwell time estimates and a random distance-based travel time are used to determine the timestamp for reaching the next location, iii) depending on this timestamp, POIs are filtered out from the candidate list (based on opening time, category information, and distance from current location) to ensure that the trajectory sequences generated are feasible and realistic, and iv) the next location POI is selected from among the remaining candidates, and the process (steps ii–iv) is repeated until the full length trajectory is complete (where no candidates exist, the trajectory is truncated). We then construct the corresponding contact network by connecting (bidirectionally) nodes that appear in the same location at the same time, considering 5 minute intervals to determine this overlap. We call this semi-synthetic network **SafeGraph-traj**.

Analysis of spread on such location-based networks has been widely studied and allows us to compare relevant baselines on popular available datasets. Nonetheless, adjustment of the propagation parameters and model activation state types can reflect other use-cases for appropriate datasets that provide granular temporal information.

We also conduct studies on two social network datasets: **wiki-Vote** [33] and **cit-HepPh** [34]. wiki-Vote consists of user discussions on Wikipedia, with edges between users representing votes. cit-HepPh encodes citation connections between research papers. These datasets reflect the typical network structure for problems such as influence maximization, and we assign propagation probabilities from related literature as described later.

Finally, for examining intervention strategies in more detail, we use the above two social networks as well as two location datasets developed for studying pandemics: **Haslemere** and **Italy**. The first records meetings between users of the BBC Pandemic Haslemere app over time, including pairwise distances with 5 minute intervals [32]. The second reports temporal aggregated mobility metrics for each day's movement of population between Italian provinces based

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

on smartphone user locations before and during the COVID-19 outbreak over 90 days [42]. The `Italy` dataset also provides transition probabilities between provinces, which we directly use as the propagation probabilities for our T-IC process.

The statistics of the constructed networks are in Table 1. Here, we report the total number of temporal edges constructed, and the maximum degree across the entire time domain of the temporal edges.

Table 1. Dataset properties

Dataset	#Nodes	#Edges	Max degree
`NYC`	876	18270	147
`Tokyo`	765	102018	311
`SP-office`	232	78249	131
`SafeGraph-traj`	2000	57530	56
`Haslemere`	469	205662	1506
`Italy`	111	235190	6808
`wiki-Vote`	8297	103689	1167
`cit-HepPh`	34546	841798	846

5.1.2 Propagation probability setting:

Each directed edge (u, v) is assigned a corresponding probability $p(u, v, t)$ of propagation from node u to node v at time t, defined based on the needs of the specific application dataset. For example, tracking disease spread may require this propagation to be based on contact duration and physical proximity, which are not relevant to malware in cyberspace. Factors like the period of transmissibility for diseases versus rumors/misinformation have different thresholds that are determined by experts. T-IC supports a variety of settings simply by adjusting the propagation rate formulation as needed. We present a few of these possibilities below:

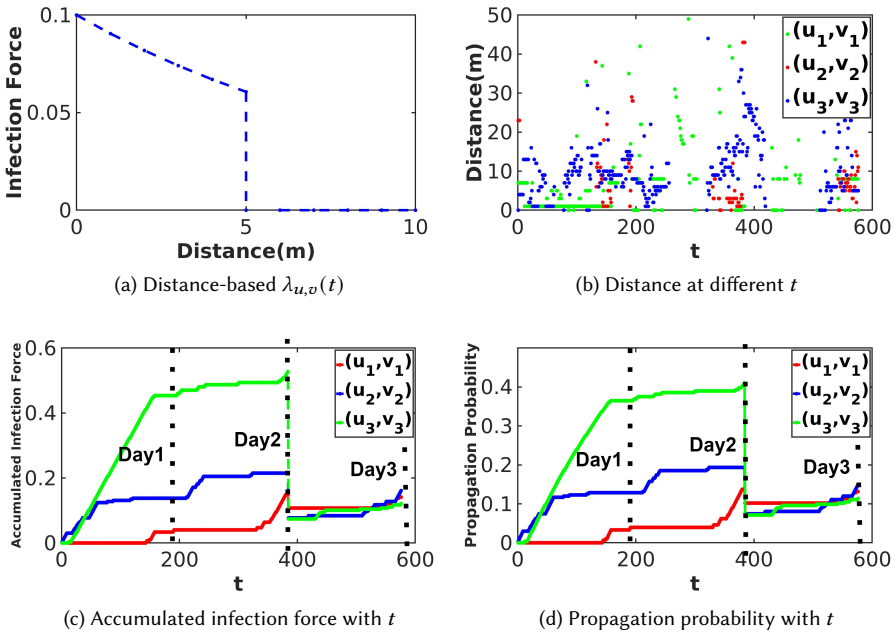
• *Sampled from a distribution:* For social network datasets, the propagation probability is assigned following the common practice in influence modeling studies [11] of using a uniform distribution. We randomly assign the edges of the network to a discrete time interval in $[0, T]$, and sample $p(u, v, t) \in [0, 0.3]$ for each edge (u, v).

• *Provided by the data:* The `Italy` dataset directly uses the provided transmission probability between connected nodes. In this dataset, it reflects the meta-population migration patterns between different regions (nodes). Note that this network is of a lower granularity than our individual-level trajectories in the location-based datasets.

• *Obtained from domain experts:* For location-based contact networks and `Haslemere`, we utilize a domain-informed probability assignment. Recent epidemiological studies quantify how transmission rates are related with the distance between the individuals as well as the overall population density at the location [2, 19, 32, 38]. Based on these, we calculate the propagation probability p of a connection from node u to node v at time interval t using Equation 4, to incorporate knowledge of virus spreading characteristics:

$$p(u, v, t) = 1 - exp(-\sum_{t'} \lambda(u, v, t')) \tag{4}$$

where $\lambda(u, v, t)$ denotes the "force of infection" (the larger the value of $\lambda(u, v, t)$, the greater is the transmission probability between u and v at time t) [32], and $t' \in (t - t_0, t]$ indicates the relevant duration of time up to the current time interval t. The latter is governed by t_0, the duration for which historic infection force is considered, since the transmission



(a) Distance-based $\lambda_{u,v}(t)$

(b) Distance at different t

(c) Accumulated infection force with t

(d) Propagation probability with t

Fig. 4. Propagation probability example for Haslemere

probability is decided by the accumulated infection force over t'. Note that $p(u, v, t)$ can drop to zero to denote no spread, such as once the individual is no longer infectious.

Therefore, a minimal expression for $\lambda(u, v, t)$ must consider the distance from u to v and the population density at the venue to determine risk, and is formulated based on the literature as:

$$\lambda(u, v, t) = ae^{-d_{u,v,t}\rho_1} + be^{-m^{-1}\rho_2} \tag{5}$$

where $d_{u,v,t}$ is the distance between u and v at time interval t (based on their location data), m is the number of people located at the same venue, and a, b, ρ_1 and ρ_2 are hyper-parameters. In line with [32], to realistically simulate spread, we use default values of $\rho_1 = \rho_2 = 0.1$, $a = 0.05$, and choose $b = 0.05$ (for `NYC`, `SafeGraph-traj`, and `SP-office` datasets) or $b = 0.01$ (for `Tokyo` dataset due to its dense connectivity). When $d_{i,j} > l$ (distance threshold) or when the dataset has no such proximity information, the contribution to infection force is assumed to be zero (i.e., $a = 0$).

Figure 4 shows an example of our dynamic probability assignment for the `Haslemere` network, demonstrating the accumulation of infection force and the changing propagation probability as distances vary with time during the interactions between three node-pairs (i.e., (u_1, v_1), (u_2, v_2), and (u_3, v_3) in Figures 4b–4d). Here, we choose a distance threshold of $l = 5$ meters as shown in Figure 4a, so there is no infection force once the distance between a node-pair is greater than 5 meters. Figure 4b shows the varying distance between contact node-pairs every 5 minutes over three days. The `Haslemere` data covers 16 hours of each day, and for simplicity of illustration we ignore the remaining hours of each day on the x-axis of Figures 4b–4d. We select $t_0 = 1$ day, i.e., the transmission probability at time t is decided by the past 1 day's interactions. Figure 4c is the corresponding accumulated infection force of Figure 4b, which is computed as $\sum_{t'} \lambda(u, v, t')$ using Equation 5 to calculate λ. The trend of the propagation probability in Figure 4d is the same as in

Figure 4c because the propagation probability is proportional to the accumulated infection force, as shown in Equation 4. A more detailed exploration of the influence of the various hyper-parameter settings for Equation 5 can be found in Section 5.2.3. While we experiment with various hyper-parameter settings for disease modeling applications, these may be customized to incorporate the domain findings on transmission risks, e.g., [4, 39] (based on contact duration, venue size and occupancy rates, activity type, ventilation, and other factors) as an orthogonal scope of work.

5.1.3 Baselines:

To our knowledge, there is no work examining sentinel and susceptible nodes on temporal networks. We thus look for comparable alternatives to our RSM and ESM solution sets. We select baselines in three groupings. The first consists of IC model-based methods (**Greedy-IM** [28], **DIA** [41]), to compare the performance of T-IC for analyzing spread on evolving networks. The second is a virus propagation method for finding the critical k nodes to immunize to prevent an epidemic (**T-Immu** [43]). The final grouping covers simple heuristic-based methods (**Max-Deg**, **Random**).

Greedy-IM obtains the top-k influential nodes using a greedy hill-climbing algorithm over $|T|$ time windows. Since it is infeasible for larger datasets, we run it only on the smallest `Haslemere` and `Italy` datasets, and apply Greedy-IM for each time window separately to calculate the corresponding spread over T and average the results to select the best node. DIA (Dynamic Influence Analysis) designs a dynamic index data structure to perform influence analysis over evolving networks. The updating index structure only shows the graph connection at the latest timestamp. We select top-k influential nodes at each time window using DIA, and report average results over the $|T|$ time windows. T-Immu formulates a non-linear dynamic system to remove a small set of nodes to prevent an epidemic. An epidemic threshold is also derived for evolving graphs. Both DIA and T-Immu handle temporal and topological information in contact networks, therefore we run DIA and T-Immu on the location datasets that include such information. The Max-Deg algorithm selects the top-k nodes in decreasing degree order. The Random algorithm selects k nodes uniformly at random in a given graph, with average results presented after 20 simulations.

We compare our solution sets with the influential sets from the alternatives with respect to the following performance measures: (i) Reverse spread from the solution set (ii) Average number of activated nodes (expected spread) in the solution set (iii) Binary success rate of detecting spread. The reverse spread $\phi(\cdot)$ is computed as defined in Section 4. The binary success rate is the average number of times that there is at least one active node in the solution set during random T-IC processes. Reverse spread is expected to be correlated with binary success, as both relate to the effectiveness of the solution set (sentinel nodes) in covering/detecting spread in the network. The expected spread, computed as the average number of activated nodes in the solution set, represents its susceptibility. Specifically, we simulate 1000 random T-IC processes to activate nodes in the network within time window T.

5.2 Evaluation of results

5.2.1 Performance with different solution set sizes k:

Tables 2 and 3 summarize the comparative results on the large-scale location datasets and social datasets. We consider solution sets (S) of sizes $k = 10, 20, \ldots, 50$ with a time window of length $|T| = 25$ days. All results are normalized for ease of comparison, i.e., the range of values between the minimum and maximum is mapped to $[0, 10]$ to produce normalized value $x_n = \frac{x - x_{min}}{x_{max} - x_{min}}$, where x is the original value, and x_{min} and x_{max} are the minimum and maximum values across all the methods on the same measure. For example, consider the normalized reverse spread on the `NYC` dataset shown in Table 2 . The value 0 for DIA in this section at $k = 10$ means that the reverse spread of DIA with $k = 10$ is minimum

Table 2. Normalized performance (reverse spread and binary success rate) at $|T|=25$ with different sizes of solution set $|S|=k$

Dataset	Method	Reverse Spread					Binary Success Rate				
k		10	20	30	40	50	10	20	30	40	50
NYC	RSM	**7.9**	**8.5**	**9.0**	**9.5**	**10**	**8.6**	**7.6**	**9.1**	**8.4**	**10**
	T-Immu	5.0	8.2	8.3	8.7	9.2	7.4	7.1	7.5	6.9	8.6
	DIA	0.0	1.0	2.4	3.2	4.1	0.0	1.1	3.1	4.2	4.8
	Max-Deg	7.5	8.0	8.4	8.8	9.2	7.5	6.7	7.4	6.8	8.3
	Random	4.0	6.3	7.9	8.7	9.2	2.1	6.0	6.6	5.7	6.6
Tokyo	RSM	**8.4**	**8.9**	**9.3**	**9.7**	**10**	**8.4**	**9.1**	**9.0**	**9.2**	**10**
	T-Immu	8.3	8.8	9.1	9.3	9.7	7.9	8.4	8.2	8.5	8.9
	DIA	0.0	0.9	2.0	3.0	4.0	0.0	0.7	1.4	2.7	4.0
	Max-Deg	7.9	8.5	9.0	9.3	9.7	7.6	8.1	7.8	8.4	9.0
	Random	6.6	8.1	8.7	9.3	9.7	5.5	6.9	6.5	7.3	7.8
SafeGraph-traj	RSM	**3.1**	**5.4**	**7.3**	**8.6**	**10**	**3.1**	**5.3**	**7.0**	**8.2**	**10**
	T-Immu	1.7	3.2	4.8	6.3	7.3	1.7	2.7	3.8	4.0	5.6
	DIA	0.0	0.1	0.1	0.2	0.2	0.1	0.1	0.2	0.2	0.2
	Max-Deg	2.0	3.3	5.1	6.3	7.5	2.1	2.6	4.2	4.2	6.6
	Random	1.7	3.4	4.9	6.5	7.8	0.0	0.1	0.6	0.7	1.4
wiki-Vote	RSM	**6.4**	**7.9**	**8.8**	**9.5**	**10**	**6.6**	**8.5**	**8.6**	**8.8**	**10**
	Max-Deg	0.0	0.8	1.9	2.9	4.4	0.6	1.8	2.8	3.4	4.1
	Random	0.1	1.1	1.8	2.5	3.0	0.0	0.6	1.3	1.7	1.9
cit-HepPh	RSM	**3.5**	**5.7**	**7.5**	**8.8**	**10**	**3.9**	**4.7**	**7.0**	**8.0**	**10**
	Max-Deg	0.1	0.2	0.3	0.6	0.9	0.0	0.2	0.4	0.6	1.0
	Random	0.0	0.0	0.0	0.4	0.5	0.0	0.0	0.0	0.0	0.0

among all methods from $k=10$ to $k=50$, while the value of 10 for RSM at $k=50$ denotes that its reverse spread in this configuration is maximum across among all methods and solution set sizes.

The set returned by RSM collectively achieves the highest reverse spread coverage in all cases, which increases with increasing k (solution set size). Without prior information about the initial seeds from where activation begins to spread, distributing limited resources (e.g., scarce/expensive wireless sensors or medical tests) to these sentinel nodes (i.e., the nodes selected in S) increases the probability of detecting the spread at an early stage.

By contrast, ESM selects all nodes having the highest probabilities of being activated during a random T-IC process, and thus best captures the largest expected spread out of all methods. ESM outperforms Max-Deg, which is often enforced in reality, by up to 82% on NYC. ESM is thus an effective targeted strategy for identifying the most susceptible nodes (e.g., for contact tracing or treatment).

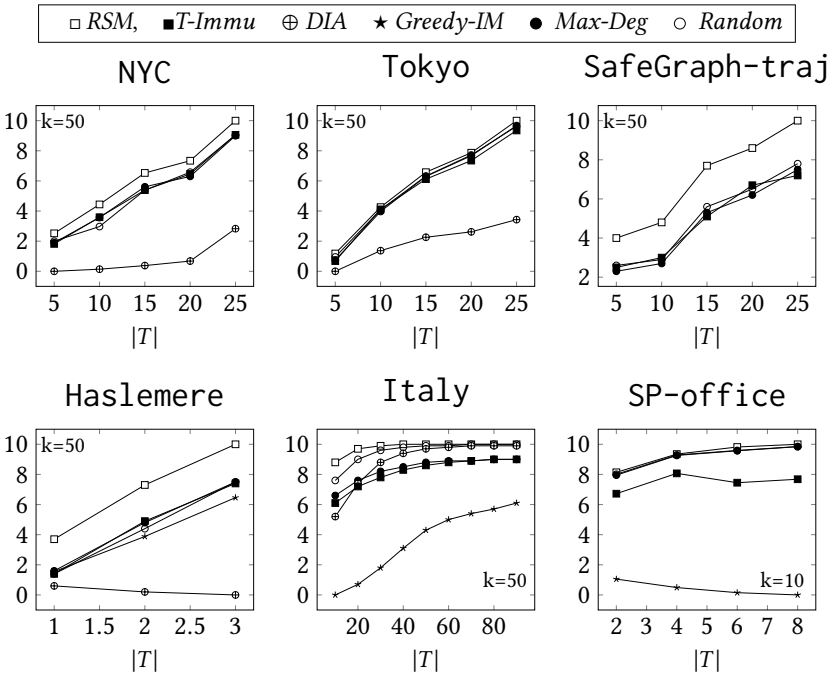
The binary success rate using RSM is the best for all datasets and k. Comparisons with T-Immu and DIA show that considering temporal properties while also preserving the overall graph structure is vital to select the ideal solution sets. RSM consistently outperforms T-Immu (nearly 2x better on SafeGraph-traj) despite having related objectives, since T-Immu cannot capture time-varying transmission probabilities. RSM also drastically outperforms DIA, which is worse than Random, because DIA selects nodes of the evolving network based on an updating index which only remembers the latest probability assignment and fails to capture the globally optimal solution set over T. The improvements (at least 10% higher success rates in the worst case) over Max-Deg confirm that the dynamic topology of the network (captured by the RSM and ESM solutions with T-IC process) plays a much more significant role compared to the local connectivity (node degrees) when modeling the spread.

Table 3. Normalized performance (expected spread) at $|T|=25$ with different sizes of solution set $|S|=k$

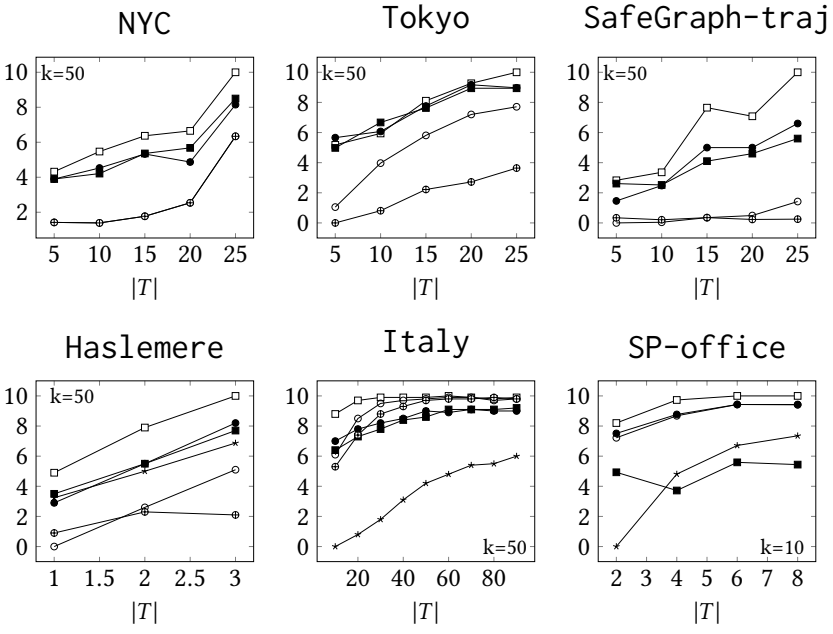
Dataset	Method	Expected Spread				
k		10	20	30	40	50
NYC	ESM	**2.3**	**3.8**	**6.1**	**7.1**	**10**
	T-Immu	0.4	0.7	1.7	2.0	2.3
	DIA	0.0	0.0	0.1	0.2	0.3
	Max-Deg	0.4	0.5	0.9	1.1	1.8
	Random	0.1	0.4	0.7	0.9	1.2
Tokyo	ESM	**2.2**	**4.5**	**5.9**	**7.8**	**10**
	T-Immu	1.1	2.0	3.5	5.0	6.8
	DIA	0.0	0.0	0.1	0.2	0.3
	Max-Deg	0.9	1.7	1.9	2.9	3.6
	Random	0.3	0.9	1.2	1.7	2.3
SafeGraph-traj	ESM	**2.1**	**3.9**	**5.9**	**6.9**	**10**
	T-Immu	0.9	1.4	2.2	2.3	3.4
	DIA	0.0	0.1	0.1	0.1	0.1
	Max-Deg	1.0	1.3	2.3	2.5	3.8
	Random	0.0	0.0	0.3	0.4	0.7
wiki-Vote	ESM	**2.8**	**6.7**	**8.4**	**9.6**	**10**
	Max-Deg	0.1	0.3	0.5	0.8	1.4
	Random	0.0	0.1	0.2	0.3	0.3
cit-HepPh	ESM	**3.5**	**5.1**	**6.7**	**8.4**	**10**
	Max-Deg	0.0	0.2	0.3	0.5	0.7
	Random	0.0	0.0	0.0	0.0	0.0

5.2.2 Performance with different time window lengths $|T|$:

We evaluate the effect of varying the time window T while keeping a constant solution set size $k=50$ on NYC, Tokyo, and SafeGraph-traj over one-day intervals up to $|T| = 25$. All three datasets provide spatio-temporal trajectories where the evolution of spread over time is relevant. We also plot the results for SP-office, Haslemere, and Italy. The time window used is different to accommodate these latter datasets. The propagation probability for Italy is the real transmission probability of people moving between any two provinces over $|T| = 90$ days, while for Haslemere it captures infections over $|T| = 3$ days. For SP-office, we consider $|T| = 8$ days, and set $k=10$ nodes. Due to the small and densely connected nature of SP-office, the baseline algorithms can identify only up to a small number of sentinel nodes. Additional nodes quickly become redundant after already covering the entire contact network. Figures 5 and 6 show that reverse spread, expected spread, and binary success all increase with $|T|$, as it allows more activations to take place. As expected, RSM has the best performance with respect to reverse spread and binary success rate in Figure 5, and ESM outperforms other methods in terms of having the best expected spread in Figure 6. Only considering the node degrees is ineffective, particularly as propagation becomes more complex, e.g., on a large network and elongated time windows. DIA is jeopardized especially with smaller time windows as the overall optimality of the solution set is not guaranteed by the most recent snapshot of the graph. Specifically, DIA selects the nodes heavily depending on the topology connections. Hence, when the connections are dense across a smaller number of venues (departments), even selecting very few nodes (2 nodes in SP-office case) immediately terminates the algorithm. In comparison, this issue can be mitigated in RSM because the propagation process is formulated with a finer granular level. Haslemere and



(a) Reverse Spread



(b) Binary Success Rate

Fig. 5. *Normalized performance (reverse spread, binary success rate) with different lengths of time window* $|T|$

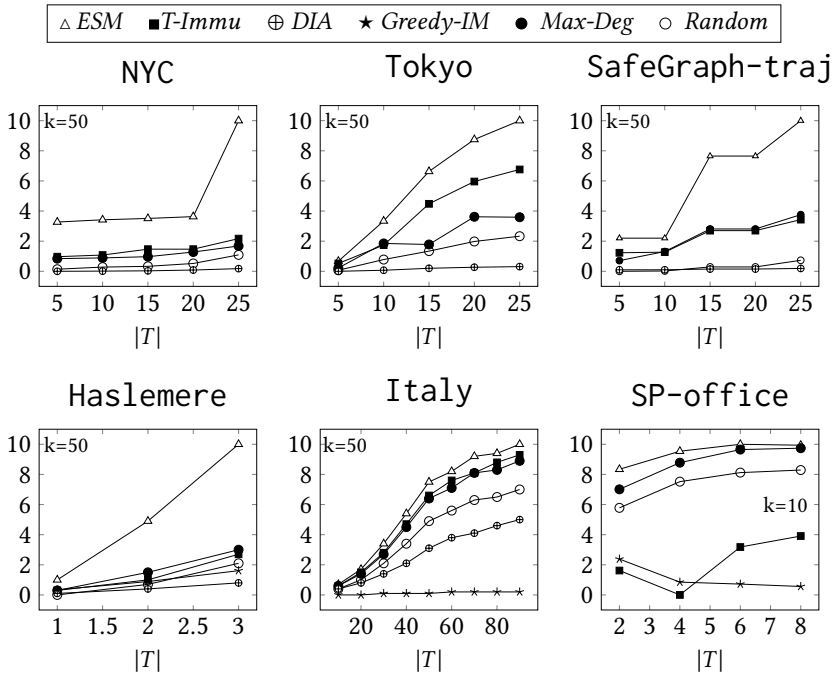
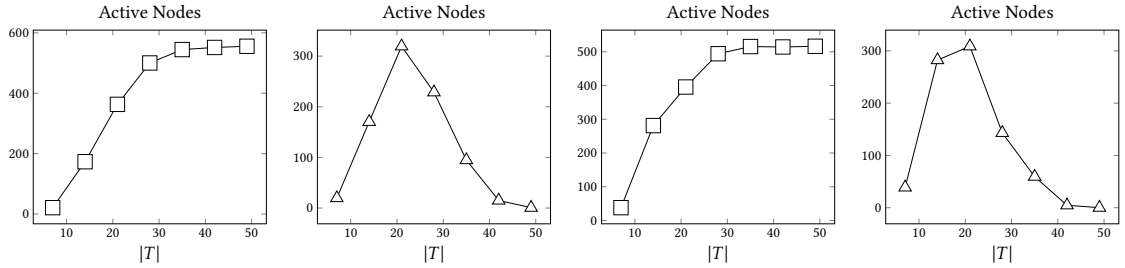


Fig. 6. *Normalized expected spread with different lengths of time window $|T|$*



(a) Total (NYC) (b) Currently active (NYC) (c) Total (Tokyo) (d) Currently active (Tokyo)

Fig. 7. *Number of active nodes with different lengths of time window $|T|$*

Italy datasets in Figures 5–6 also highlight how Greedy-IM cannot effectively capture the optimal global solution over multiple time windows.

We also measure the size of the active set of nodes as $|T|$ increases, for the NYC and Tokyo datasets. Despite there being only $T = 25$ days of contact information for these datasets, we plot results for $T = 50$ days as shown in Figure 7. This is because we set $t_0 = 14$ days as the duration of infectious force (considering a disease setting), meaning that a node once activated remains infectious for 14 days before the propagation rate drops to 0. Therefore, activations may continue to spread in the network beyond the 25th day. The total number of activated nodes is simulated by sampling 10k hypergraphs. As shown in Figures 7 (a) and (c), when $|T| > 25$ the increase in the number of activations gets slower. The active set gradually becomes stable as there are no new contacts between nodes. Since the propagation rates also

Table 4. The default values of hyper-parameters for different datasets

Dataset	Hyper-parameters	p
NYC, SafeGraph-traj	a=0, b=0.05, ρ_2=0.1	Using Equation 4
Tokyo	a=0, b=0.01 , ρ_2=0.1	Using Equation 4
SP-office	a=0, b=0.01, ρ_1=0.1	Using Equation 4
Haslemere	a=0.05, b=0, ρ_1=0.1, l=5	Using Equation 4
Italy	×	Using provided p values from data
wiki-Vote, cit-HepPh	×	Using uniform distribution

taper down for nodes that have been previously activated, there is no further significant increase in overall spread. However, this does not accurately depict how many nodes are currently activated at any time. To measure this, we introduce a new "recovery" state by reverting active nodes to inactive state after the t_0 period. In Figure (b) and Figure (d), this simulation is found to closely match the behavior of SIR framework models (e.g., the SEIR model used by Wang et al. [55]). Once an active node "recovers" after $t_0 = 14$ days, we observe that the increase in the number of active nodes is slower. The active set eventually reduces in size with $|T|$ as more nodes recover.

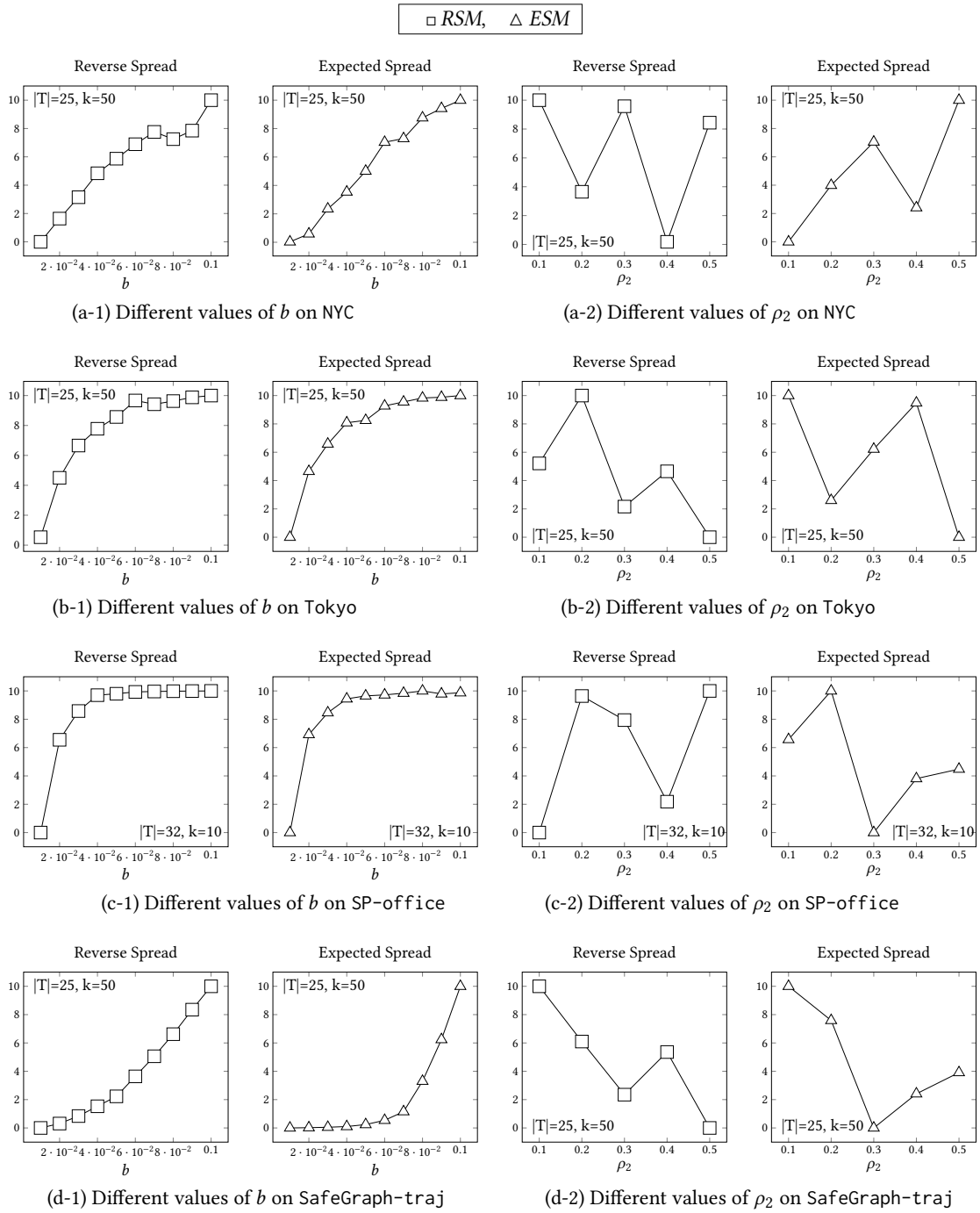
5.2.3 Performance with different hyper-parameters:

The propagation probability in Equation 4 is proportional to hyper-parameters a, b, and l, while it is inversely proportional to ρ_1 and ρ_2. Furthermore, the propagation rates are sensitive to small changes in ρ_1 and ρ_2 since they directly influence the threshold of importance of proximity and population density respectively. The default values of hyper-parameters for all datasets are summarized in Table 4. We evaluate different hyper-parameter settings on NYC, Tokyo, SP-office, SafeGraph-traj, and Haslemere datasets. These hyper-parameters are not necessary for Italy and the social network datasets (wiki-Vote and cit-HepPh), since the computation of p does not involve them.

We experiment with $\rho_1, \rho_2 \in [0.1, 0.5]$, $a, b \in [0.01, 0.1]$, and $l \in [5, 20]$. We fix one of the parameters as the default value, then experiment with different values of the others, e.g., setting $b = 0.05$ for SafeGraph-traj, and changing ρ_2 from 0.1 to 0.5. Specifically, we modify b and ρ_2 for NYC, Tokyo, SP-office, and SafeGraph-traj, where the population density at POIs is a relevant factor for the risk of propagation of disease. Meanwhile a, ρ_1, and l are relevant for determining propagation risk in Haslemere, and we modify them for this dataset in order to reflect different distance thresholds between infected and potential susceptible individuals.

Figure 8 shows the spread resulting from RSM and ESM with different b and ρ_2 on the contact-based datasets NYC, Tokyo, SP-office, and SafeGraph-traj. The performance fluctuates with the increase of ρ_2, while the performance gets large strictly with the increase of b. As shown in Equation 5, ρ_2 is the factor that dictates the importance of the number of people m in determining transmission risk. Hence, in the real-life application, it is important to select an appropriate value of ρ_2 to reflect how the density of population at a POI contributes to the spread. It is important to note that we do not select our hyper-parameter values in a way that maximizes the spread and provides any undue advantage for our method. Rather, we choose values that are reasonable reflections of real-world scenarios.

We present the spread resulting from RSM and ESM solutions with different a, ρ_1, and l on Haslemere in Figure 9. The reverse spread and expected spread increase overall with increase in a and l while decrease with increase in ρ_1. This is intuitive, since the likelihood of propagation increases with larger values of a, and the longer distance threshold l implies a high probability that the infection will successfully spread when individuals are in proximity to each other even though they are separated by some distance. Meanwhile, a large choice of ρ_1 will decrease the factor of the infection force that is brought about by the proximate contact.



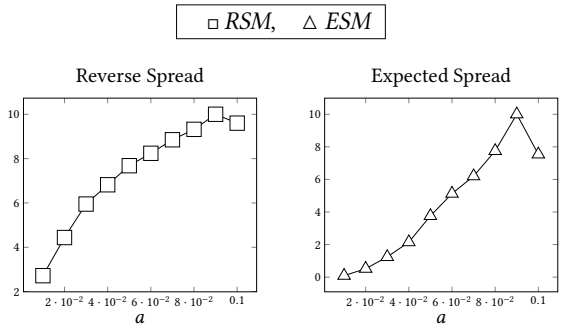
(a-1) Different values of b on NYC (a-2) Different values of ρ_2 on NYC

(b-1) Different values of b on Tokyo (b-2) Different values of ρ_2 on Tokyo

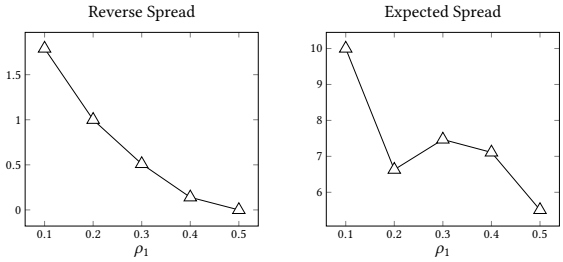
(c-1) Different values of b on SP-office (c-2) Different values of ρ_2 on SP-office

(d-1) Different values of b on SafeGraph-traj (d-2) Different values of ρ_2 on SafeGraph-traj

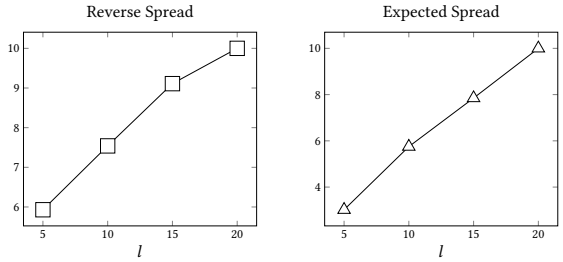
Fig. 8. *Spread resulting from different values of b and ρ_2*



(a) Different values of a on Haslemere



(b) Different values of ρ_1 on Haslemere



(c) Different values of l on Haslemere

Fig. 9. *Spread resulting from different values of a, ρ_1, l (|T|=3,k=50)*

5.2.4 *Running time efficiency:*

While the solution quality improves with higher number of hypergraph nets generated (up to a certain point), there is an efficiency trade-off. We measure the running time of generating hypergraph nets by varying the desired number of nets $|N|$ and the number of time windows $|T|$ under consideration, as shown in Table 5 and Table 6, respectively. Specifically, with $|N|$ increasing from $20K$ to $100K$ (for a fixed $|T|=5$) in Table 5, there is a slight increase in running time (from 0.43 to 2.18 seconds for Tokyo) demonstrating the efficient and scalable nature of our reachable set sampling based algorithm. The hypergraph construction time also increases with $|T|$ (for a fixed $|N|=20K$) in Table 6 due to a prolonged propagation process, which is especially evident in dense networks (e.g., Tokyo). Despite the increase in computation time and memory requirements when increasing $|N|$, we find that stable solution sets of sufficiently high quality are produced without the need for more than $20K$ hypergraphs.

Table 5. Running time (in seconds) with different number of nets $|N|$ ($|T| = 5$)

Dataset \ |N|	20000	40000	60000	80000	100000
NYC	0.99	1.95	2.96	4.00	4.99
Tokyo	0.43	0.86	1.31	1.75	2.18
SP-office	0.83	1.71	2.62	3.51	4.43
SafeGraph-traj	0.04	0.07	0.11	0.15	0.19
wiki-Vote	53.40	108.76	158.52	216.58	271.99
cit-HepPh	1.27	2.69	3.98	5.35	6.45

Table 6. Running time (in seconds) with different number of time windows $|T|$ ($|N| = 20K$)

Dataset \ |T|	5	10	15	20	25
NYC	0.98	1.80	3.34	5.51	11.32
Tokyo	0.42	5.40	22.65	55.94	106.0
SP-office	0.83	2.64	6.04	12.21	18.35
SafeGraph-traj	0.04	0.06	0.10	0.12	0.16
wiki-Vote	53.40	149.49	320.87	549.95	917.51
cit-HepPh	1.27	7.46	25.68	63.73	130.65

Table 7. Running time (in seconds) with different sizes of solution set |S|=k on Haslemere

Method \ k	10	20	30	40	50
RSM	19.15	19.16	19.20	19.23	19.26
ESM	18.34	18.34	18.35	18.35	18.36
Greedy-IM	878.98	2647.99	6729.71	11780.86	18948.14

T-IC can model large-scale individual-level contacts efficiently, whereas other solutions [23, 41, 43] are only feasible on small graphs. For example, T-Immu repeats computations of the eigenvalue of the dynamic contact network structure which makes it not feasible for large-scale datasets (e.g., cit-HepPh), and DIA can only consider the latest snapshot but not the global structure efficiently.

We compare RSM with the commonly used IC model based method, Greedy-IM, in terms of the running time for selecting different size of solution set S from $k=10$ to $k=50$ in Table 7. The running time of Greedy-IM grows quickly and gets infeasible with large dataset and long time windows, whereas our RSM and ESM running times grow much more slowly. Hence, for running time comparisons, we only evaluate on the small dataset Haslemere. We observe the running time of Greedy-IM increases sharply from 879s to 18948s with increase in k at $T=3$, which makes it not applicable for large evolving networks.

5.3 Intervention Strategies

We now analyze the effect of several intervention strategies for reducing spread in temporal networks. Several of these techniques have applicability to the disease monitoring context. There are analogues to reduce spread in other settings such as with misinformation/rumors propagating in social networks. We study targeted connection shutdown, backward contact tracing, and node-level categorical analysis.

5.3.1 *Targeted Connection Shutdown:*

We first consider the reduction of edge connections in our network construction and analyze how the spread changes as a result of these dropped edges. For the a disease spread setting, this simulates (partial) lockdown strategies.

We randomly select a seed set of 10 nodes from which to simulate the T-IC process. We use two intervention strategies to select the edge connections to drop (we drop 30% of the total edges). The first is to randomly drop edges. The second is based on the priority of the nodes, i.e., delete connections for the node categories visited by more people. That is, the number of edges dropped is proportional to the number of connections to the nodes. We perform 20 simulations to get the average decrease in spread resulting from each of the two strategies. For `NYC`, random deletion reduces 78% spread while node category prioritization (on the top-50 busiest venues) achieves 83% spread reduction. Similarly, for `SafeGraph-traj`, random deletion reduces 26% spread while an additional 4% spread reduction is achieved by venue prioritizing the top-50 busiest venues. For the less granular `Italy` dataset, which does not have venue information, we prioritize the deletion of the top-50 densely connected provinces. Spread reduction is 49% when using prioritization, while random deletion reduces 38% spread. For `SP-office` dataset, random deletion can only reduce 5% spread while 22.8% spread is reduced by prioritizing the top-5 most visited categories (departments). Therefore, a targeted approach to connection shutdown at specific nodes shows superior performance over random occupancy/usage restrictions across all nodes.

The same holds true for social datasets where the propagation rates were sampled from a uniform distribution. For social datasets, there is no venue information provided therefore all nodes are of the same type, so we prioritize the most connected nodes from which we drop a number of edge connections. Random deletion of 5% edge connections on `wiki-Vote` can reduce 26.75% spread while there is a 54.78% spread reduction when prioritizing the 100 most connected nodes. For `cit-HepPh`, 22.45% spread is reduced when dropping 30% connections on the 1000 most connected nodes, while the random deletion can only reduce 6.78% spread.

5.3.2 *Backward Contact Tracing:*

We next calculate the contribution of backward traced nodes to the activations in the selected ESM solution set, in Table 8. Backward contact tracing has gained popularity in the epidemiological domain. A similar idea could be applied in a social network setting to track content back from the followers of users sharing misinformation. First, we select different sizes of solution set S from $k=10$ to $k=50$. Considering the reverse reachable set of nodes from a given solution set for backward tracing, we identify the top spread contributors as the nodes that participate most frequently in activations. We find that this backward traced set of superspreader nodes account for 67.9% to 95.0% of the activations in S on the `Haslemere` dataset. For `Tokyo`, they contribute 77.8% to 96.1%. The solution set of `SP-office` can contribute 39.2% to 70.7%. Similar patterns can also be observed on the two social networks. This skewed over-dispersion further points to the importance of backward contact tracing and need for suppressing superspreader events to tackle unwanted spread in different settings.

5.3.3 *Node-Level Categorical Analysis:*

The `Tokyo` and `NYC` datasets also include node-level categories (venue) which provide further insights for designing effective intervention strategies. For $|T|=25$, we observe that only 26 to 83 venues in NYC are visited by persons in solution sets selected by RSM when increasing k from 10 to 50, while ESM, Max-Deg, and Random cover up to 3x as many venues. For `Tokyo` and `NYC`, an analysis of the categories of venues visited reveals transportation hubs (including

Table 8. Contribution (% activations among nodes of S) of backward traced superspreaders with different sizes of solution set $|S|=k$

Dataset \ k	10	20	30	40	50
NYC	39.9	53.6	63.3	71.1	77.3
Tokyo	77.8	91.0	94.3	95.5	96.1
SafeGraph-traj	32.0	37.2	41.5	45.3	48.9
SP-office	39.2	50.9	59.1	65.4	70.7
Haslemere	67.9	79.1	86.0	91.1	95.0
Italy	16.7	31.4	44.5	56.1	66.6
wiki-Vote	8.90	15.7	21.5	26.6	31.4
cit-HepPh	18.6	28.6	36.2	42.4	47.3

airport, subway, and train station), restaurants, bars, and coffee shops as superspreaders in the solutions sets, with transportation hubs in particular having an out-sized impact when increasing the set size k of infected individuals.

6 CONCLUSION

In this work, we introduce the *Temporal Independent Cascade (T-IC)* model for the tasks of *Reverse Spread Maximization (RSM)* and *Expected Spread Maximization (ESM)*, and illustrate its application in various settings. We show that reverse spread under the T-IC model is submodular, and propose efficient algorithms to produce solution sets for RSM and ESM. These algorithms are able to maintain the approximation guarantees of IC models in temporal networks, enabling them to handle large-scale and highly granular data. Our objectives are to identify i) a minimal set of sentinel nodes (i.e., nodes which minimally cover the network, for the purpose of spread detection), and ii) a set of highly susceptible nodes (e.g., for prioritizing tracing, intervention, and treatment measures across various use cases). Through extensive quantitative analysis performed on eight real-world datasets across multiple settings, we show that RSM significantly outperforms alternative methods for the former task, while the ESM solution sets capture significantly more susceptible individuals for the latter. We observe that the dynamic topology captured by our model plays a more significant role than local connectivity, which is evident in the sentinel nodes identified by RSM having significantly higher success rates of detecting spread compared to T-Immu and DIA. We also find that temporal characteristics alongside the global graph structure are needed for optimal solutions, which can be seen as ESM significantly outperforms Max-Deg as a superior targeted strategy for identifying susceptible nodes. Finally, we consider several targeted intervention strategies (targeted connection shutdown, backward contact tracing, and node-level categorical analysis), and show that the T-IC model can be effectively leveraged for these purposes in a variety of application settings.

ACKNOWLEDGMENTS

This research is supported in part by The Alan Turing Institute under the EPSRC grant EP/N510129/1. Aparajita and Joe are supported by the Feuer International Scholarship in Artificial Intelligence.

REFERENCES

[1] Charu C Aggarwal, Shuyang Lin, and Philip S Yu. 2012. On influential node discovery in dynamic social networks. In *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 636–647.

[2] Chrysovalantis Anastasiou, Constantinos Costa, Panos K Chrysanthis, Cyrus Shahabi, and Demetrios Zeinalipour-Yazti. 2021. ASTRO: Reducing COVID-19 Exposure through Contact Prediction and Avoidance. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 8, 2 (2021), 1–31.

[3] Paolo Bajardi, Alain Barrat, Lara Savini, and Vittoria Colizza. 2012. Optimizing surveillance for livestock disease spreading through animal movements. *Journal of the Royal Society Interface* 9, 76 (2012), 2814–2825.

[4] Martin Bazant and John Bush. 2020. A Guideline to Limit Indoor Airborne Transmission of COVID-19. *Bulletin of the American Physical Society* (2020).

[5] Seth G Benzell, Avinash Collis, and Christos Nicolaides. 2020. Rationing social contact during the COVID-19 pandemic: Transmission risk and social benefits of US locations. *Proceedings of the National Academy of Sciences* 117, 26 (2020), 14642–14644.

[6] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing social influence in nearly optimal time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 946–957.

[7] Serina Chang, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, and Jure Leskovec. 2021. Mobility network models of COVID-19 explain inequities and inform reopening. *Nature* 589, 7840 (2021), 82–87.

[8] Wei Chen, Wei Lu, and Ning Zhang. 2012. Time-critical Influence Maximization in Social Networks with Time-delayed Diffusion Process. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence* (Toronto, Ontario, Canada) *(AAAI'12)*. AAAI Press, 592–598.

[9] Wei Chen, Chi Wang, and Yajun Wang. 2010. 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* (Washington, DC, USA) *(KDD '10)*. ACM, New York, NY, USA, 1029–1038.

[10] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient Influence Maximization in Social Networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Paris, France) *(KDD '09)*. ACM, New York, NY, USA, 199–208.

[11] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE international conference on data mining*. IEEE, 88–97.

[12] Nicholas A Christakis and James H Fowler. 2010. Social network sensors for early detection of contagious outbreaks. *PloS one* 5, 9 (2010), e12948.

[13] Gianluca Dini, Marco Pelagatti, and Ida Maria Savino. 2008. An algorithm for reconnecting wireless sensor network partitions. In *European Conference on Wireless Sensor Networks*. Springer, 253–267.

[14] Pedro Domingos and Matt Richardson. 2001. Mining the Network Value of Customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California) *(KDD '01)*. ACM, New York, NY, USA, 57–66.

[15] Akira Endo, Quentin J Leclerc, Gwenan M Knight, Graham F Medley, Katherine E Atkins, Sebastian Funk, Adam J Kucharski, et al. 2020. Implication of backward contact tracing in the presence of overdispersed transmission in COVID-19 outbreak. *medRxiv* (2020).

[16] Nathalie TH Gayraud, Evaggelia Pitoura, and Panayiotis Tsaparas. 2015. Diffusion maximization in evolving social networks. In *Proceedings of the 2015 ACM Conference on Online Social Networks*. 125–135.

[17] Mathieu Génois and Alain Barrat. 2018. Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Science* 7, 1 (2018), 1–18.

[18] Andrzej Grabowski and Andrzej Jarynowski. 2016. Rumor propagation in temporal contact network from polish polls. In *2016 Third European Network Intelligence Conference (ENIC)*. IEEE, 85–89.

[19] Bnaya Gross, Zhiguo Zheng, Shiyan Liu, Xiaoqi Chen, Alon Sela, Jianxin Li, Daqing Li, and Shlomo Havlin. 2020. Spatio-temporal propagation of COVID-19 pandemics. *EPL (Europhysics Letters)* 131, 5 (2020), 58003.

[20] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. 2013. Information diffusion in online social networks: A survey. *ACM Sigmod Record* 42, 2 (2013), 17–28.

[21] Kai Han, Keke Huang, Xiaokui Xiao, Jing Tang, Aixin Sun, and Xueyan Tang. 2018. Efficient algorithms for adaptive influence maximization. *Proceedings of the VLDB Endowment* 11, 9 (2018), 1029–1040.

[22] Herbert W Hethcote. 2000. The mathematics of infectious diseases. *SIAM review* 42, 4 (2000), 599–653.

[23] Petter Holme. 2017. Three faces of node importance in network epidemiology: Exact results for small graphs. *Physical Review E* 96, 6 (2017), 062305.

[24] Petter Holme. 2018. Objective measures for sentinel surveillance in network epidemiology. *Physical Review E* 98, 2 (2018), 022313.

[25] Ting Jiang, Yang Zhang, Minhao Zhang, Ting Yu, Yizheng Chen, Chenhao Lu, Ji Zhang, Zhao Li, Jun Gao, and Shuigeng Zhou. 2022. A survey on contact tracing: the latest advancements and challenges. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 8, 2 (2022), 1–35.

[26] Levente Juhász and Hartwig H Hochmair. 2020. Studying spatial and temporal visitation patterns of points of interest using SafeGraph data in Florida. (2020).

[27] Márton Karsai and Nicola Perra. 2017. Control strategies of contagion processes in time-varying networks. In *Temporal Network Epidemiology*. Springer, 179–197.

[28] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the Spread of Influence Through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, D.C.) *(KDD '03)*. ACM, New York, NY, USA, 137–146. https://doi.org/10.1145/956750.956769

[29] William Ogilvy Kermack and Anderson G McKendrick. 1927. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 115, 772 (1927), 700–721.

[30] Jinha Kim, Wonyeol Lee, and Hwanjo Yu. 2014. CT-IC: Continuously activated and time-restricted independent cascade model for viral marketing. *Knowledge-Based Systems* 62 (2014), 57–68.

[31] István Z Kiss, Joel C Miller, Péter L Simon, et al. 2017. Mathematics of epidemics on networks. *Cham: Springer* 598 (2017).

[32] Stephen M Kissler, Petra Klepac, Maria Tang, Andrew JK Conlan, and Julia R Gog. 2020. Sparking" The BBC Four Pandemic": Leveraging citizen science and mobile phones to model the spread of disease. *bioRxiv* (2020), 479154.

[33] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*. 641–650.

[34] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. 177–187.

[35] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. 2007. Cost-effective Outbreak Detection in Networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Jose, California, USA) *(KDD '07)*. ACM, New York, NY, USA, 420–429.

[36] Bo Liu, Gao Cong, Dong Xu, and Yifeng Zeng. 2012. Time Constrained Influence Maximization in Social Networks. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM '12)*. IEEE Computer Society, Washington, DC, USA, 439–448.

[37] Wanping Liu and Shouming Zhong. 2017. Web malware spread modelling and optimal control strategies. *Scientific reports* 7, 1 (2017), 1–19.

[38] Rajat Mittal, Charles Meneveau, and Wen Wu. 2020. A mathematical framework for estimating risk of airborne transmission of COVID-19 with application to face mask use and social distancing. *Physics of Fluids* 32, 10 (2020), 101903.

[39] Freja Nordsiek, Eberhard Bodenschatz, and Gholamhossein Bagheri. 2021. Risk assessment for airborne disease transmission by poly-pathogen aerosols. *Plos one* 16, 4 (2021), e0248004.

[40] Anastasios Noulas, Cecilia Mascolo, and Enrique Frias-Martinez. 2013. Exploiting foursquare and cellular data to infer user activity in urban environments. In *2013 IEEE 14th international conference on mobile data management*, Vol. 1. IEEE, 167–176.

[41] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2016. Dynamic influence analysis in evolving networks. *Proceedings of the VLDB Endowment* 9, 12 (2016), 1077–1088.

[42] Emanuele Pepe, Paolo Bajardi, Laetitia Gauvin, Filippo Privitera, Brennan Lake, Ciro Cattuto, and Michele Tizzoni. 2020. COVID-19 outbreak response, a dataset to assess mobility changes in Italy following national lockdown. *Scientific data* 7, 1 (2020), 1–7.

[43] B Aditya Prakash, Hanghang Tong, Nicholas Valler, Michalis Faloutsos, and Christos Faloutsos. 2010. Virus propagation on time-varying networks: Theory and immunization algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 99–114.

[44] Sirisha Rambhatla, Sepanta Zeighami, Kameron Shahabi, Cyrus Shahabi, and Yan Liu. 2022. Toward Accurate Spatiotemporal COVID-19 Risk Scores Using High-Resolution Real-World Mobility Data. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 8, 2 (2022), 1–30.

[45] Wendi Ren, Jiajing Wu, Xi Zhang, Rong Lai, and Liang Chen. 2018. A stochastic model of cascading failure dynamics in communication networks. *IEEE Transactions on Circuits and Systems II: Express Briefs* 65, 5 (2018), 632–636.

[46] Anna Izabel João Tostes Ribeiro, Thiago Henrique Silva, Fátima Duarte-Figueiredo, and Antonio AF Loureiro. 2014. Studying traffic conditions by analyzing foursquare and instagram data. In *Proceedings of the 11th ACM symposium on Performance evaluation of wireless ad hoc, sensor, & ubiquitous networks*. 17–24.

[47] Safegraph. 2020. *Safegraph Places Schema*. Retrieved April 2, 2021 from https://docs.safegraph.com/v4.0/docs/

[48] Frederik Schirdewahn, Vittoria Colizza, Hartmut HK Lentz, Andreas Koher, Vitaly Belik, and Philipp Hövel. 2017. Surveillance for outbreak detection in livestock-trade networks. In *Temporal Network Epidemiology*. Springer, 215–240.

[49] Paulo Shakarian, Abhinav Bhatnagar, Ashkan Aleali, Elham Shaabani, and Ruocheng Guo. 2015. The independent cascade and linear threshold models. In *Diffusion in Social Networks*. Springer, 35–48.

[50] Kai Shu, H Russell Bernard, and Huan Liu. 2019. Studying fake news via network analysis: detection and mitigation. In *Emerging research challenges and opportunities in computational social network analysis and mining*. Springer, 43–65.

[51] Anshoo Tandon, Teng Joon Lim, and Utku Tefek. 2019. Sentinel based malicious relay detection in wireless IoT networks. *Journal of Communications and Networks* 21, 5 (2019), 458–468.

[52] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *Proceedings of the 2018 International Conference on Management of Data*. 991–1005.

[53] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. 1539–1554.

[54] Vijay V Vazirani. 2013. *Approximation algorithms*. Springer Science & Business Media.

[55] Haotian Wang, Abhirup Ghosh, Jiaxin Ding, Rik Sarkar, and Jie Gao. 2021. Heterogeneous interventions reduce the spread of COVID-19 in simulations on real mobility data. *Scientific Reports* 11, 1 (2021), 1–12.

[56] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. 2010. Community-based Greedy Algorithm for Mining top-K Influential Nodes in Mobile Social Networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, DC, USA) *(KDD '10)*. ACM, New York, NY, USA, 1039–1048.

[57] Yingzi Wang, Xiao Zhou, Cecilia Mascolo, Anastasios Noulas, Xing Xie, and Qi Liu. 2018. Predicting the Spatio-Temporal Evolution of Chronic Diseases in Population with Human Mobility Data. IJCAI.

[58] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2014), 129–142.

[59] Yu Yang, Zhefeng Wang, Jian Pei, and Enhong Chen. 2017. Tracking influential individuals in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2615–2628.

[60] Qingyi Zhu, Xiaofan Yang, and Jianguo Ren. 2012. Modeling and analysis of the spread of computer virus. *Communications in Nonlinear Science and Numerical Simulation* 17, 12 (2012), 5117–5124.