Temporal sequence of retweets help to detect influential nodes in social networks

Ayan Kumar Bhowmick*, Martin Gueuning†‡, Jean-Charles Delvenne†, Renaud Lambiotte‡§ and Bivas Mitra*
*Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India
†ICTEAM, Universite Catholique de Louvain, Louvain-la-Neuve, Belgium
‡naXys, University of Namur, Namur, Belgium §Mathematical Institute, University of Oxford, UK
Email: ayankb@iitkgp.ac.in, martin.gueuning@uclouvain.be, jean-charles.delvenne@uclouvain.be,
renaud.lambiotte@unamur.be, bivas@cse.iitkgp.ernet.in

***Abstract*—Identification of influential users in online social networks allows to facilitate efficient information diffusion to a large part of the network, thus benefiting diverse applications including viral marketing, disease control and news dissemination. Existing methods have mainly relied on the network structure only for the detection of influential users. In this paper, we enrich this approach by proposing a fast, efficient and unsupervised algorithm *SmartInf* to detect a set of influential users by identifying *anchor nodes* from temporal sequence of retweets in Twitter cascades. Such *anchor nodes* provide important signatures of tweet diffusion across multiple diffusion localities and hence act as precursors for detection of influential nodes[1]. The set of influential nodes identified by *SmartInf* have the capacity to expose the tweet to a large and diverse population, when targeted as seeds thereby maximizing the influence spread. Experimental evaluation on empirical datasets from Twitter show the superiority of *SmartInf* over state-of-the-art baselines in terms of infecting larger population; further, our evaluation shows that *SmartInf* is scalable to large-scale networks and is robust to missing data. Finally, we investigate the key factors behind the improved performance of *SmartInf* by testing our algorithm on a synthetic network using synthetic cascades simulated on this network. Our results reveal the effectiveness of *SmartInf* in identifying a diverse set of influential users that facilitate faster diffusion of tweets to a larger population.**

***Index Terms*—*SmartInf*, inter-retweet time intervals, *anchor nodes*, cascades**

I. Introduction

Micro-blogging platforms such as Twitter and Sina Weibo have become widely popular for the creation and dissemination of real-time information in the form of breaking news, personal updates and spontaneous ideas [1], [2]. For instance, Twitter serves as an effective medium for real-time posts about earthquakes, epidemic outbreaks or spreading awareness in many situations, like the Arab-Spring movement in 2011 [3] or the U.S. presidential elections in 2016 [4]. In recent years, interest of the research community has increasingly focused on whether diffusion can be maximized by seeding a piece of information with certain special individuals, often called *influential users* [5]–[7]. These influential users may play a significant role in the diffusion process, favoring a larger spread of the information. Such users should have an indirect impact in shaping the behavior or actions of a large number of other users due to their activity as well as position in the network. Once identified, these users, who typically represent a very small fraction of the network, may be targeted directly for efficient information spread.

Existing literature on identifying influential users in a social network have mainly concentrated on using the knowledge of the underlying network topology [8]–[11]. Few attempts have been made to quantify the influence of users in Twitter by proposing different influence measures as well as using various centrality measures [8], [12]–[15]. Further, some recent state-of-the-art methods have combined diverse structural information by performing k-core or K-truss decomposition of the graph for identifying influential nodes with better spreading behavior [16]–[18]. In addition, some works have relied on computing pairwise influence among users to rank influential nodes [19], [20] as well as detecting structural holes playing key influential roles in diffusing information over the network [12], [21], [22]. A brief review of the state-of-the-art has been provided in Sec. II.

However, to the best of our knowledge, the state-of-the-art literature mostly overlooked the broad heterogeneity of users in terms of their temporal patterns, while identifying the influential nodes. A user that occupies a highly central position in the network may remain (i) inactive (passive) in posting retweets or (ii) mostly participates in non-popular cascades. Hence this user, despite of her favorable structural position, may not be a highly influential node. On the other hand, overall activity (say (re)tweet) rate, even for a structurally central user, does not reflect her true role in diffusing tweets in popular cascades. Hence, the methods relying purely on the network structure & mean user activity are blind to these aspects, thereby may fail in their purpose by highly ranking some ineffective users who are very unlikely to reinforce the spreading process. Moreover, state-of-the-art methods [13], [14], [23] relying on network topology need to directly compute the influence of each node to identify & rank the influential users; this requires complete information of the network topology, which is typically difficult and expensive to obtain for large-scale social networks. Notably, the dependency on the complete follower network significantly affects the scalability of the aforementioned algorithms.

In Twitter, after each retweet event, a new set of users gets exposed to the content. Notably, this diffusion process in the

[1] We use the terms ‘influential nodes’ and ‘influential users’ interchangeably.

underlying follower network is not uniform, rather the cascade evolves in bursts. In this context, we introduce the concept of *diffusion locality* associated to a cascade as a connected set of exposed users which slowly grows with each retweet activity. Precisely, after each retweet event, if the set of newly exposed users is not too large, it is absorbed by the current diffusion locality. However, if the tweet gets exposed to a large new population, the cascade propagates to a new diffusion locality associated to it. The transition of a cascade across multiple localities is facilitated by retweets caused by the *anchor nodes*.

Our empirical study reveals that inter-retweet time intervals exhibit a signature of tweet diffusion from one locality to another. Once posted, a tweet initially gains popularity & gets frequently retweeted within its locality; this results in low inter-retweet intervals. However, saturation of a tweet within a diffusion locality makes its content redundant for the population in that locality and subsequently slows down the tweet diffusion process; this results in a surge in the inter-retweet intervals. In this scenario, a retweet caused by the *anchor node* may diffuse the post to a new population (locality), where this tweet again gains popularity, which gets reflected by retweets in quick succession inside the new locality; this results in a drop in inter-retweet intervals. We empirically show the co-occurrence between first peaks in the inter-retweet intervals and migration of the cascade to a new diffusion locality. Hence, identifying the anchor nodes, who play a key role in diffusing the post from one locality to another, may be an important step to locate the influential nodes. Additionally, influential nodes with overlapping followers fail to diffuse the cascade to a large new population. Hence, utmost care needs to be taken to recommend those influential nodes which have capacity to reach a diverse population. Our paper takes an important step towards this direction. The objective of this work is to propose an algorithm leveraging on both the temporal retweet sequence of cascades and on the local structural information to identify the set of influential users in the network. This approach has the benefits of not requiring the knowledge of the full topological structure of the network on one hand while taking into account the heteregeneous activity among users on the other hand in a non-trivial way.

In this paper, after a brief review of the relevant literature (Sec. II), we introduce the studied Twitter datasets. We represent the retweet sequence of a cascade as inter-retweet time intervals from the retweet timestamps associated with each cascade (Sec. III). In Sec. IV, we propose a scalable unsupervised algorithm *SmartInf* to detect the influential nodes in a social network. The proposed methodology relies on the peaks observed from the inter-retweet intervals of the cascades, which designate the diffusion of the post from one locality to another, thus capturing the broad heterogeneity across users. Our methodology consists of two major phases. First, we establish the ranked list of influential nodes *only* from the temporal retweet sequence of cascades, which is easy to obtain & process. This phase makes our algorithm scalable. Next, we refine this list to diversify the population exposed through these influential nodes. In Sec. V, we present the experimental setup to evaluate the proposed algorithm *SmartInf*. First we present the competing algorithms for identifying the influential users in a social network. For performance evaluation, (a) we introduce the quality metrics of the influential nodes, and (b) we develop an epidemic simulation setup to measure the volume of population exposed to a tweet originated from the influential nodes. In Sec. VI, we show that on both the yardsticks, the quality of influential nodes detected by the proposed *SmartInf* method outperforms the state-of-the-art baseline algorithms. We also demonstrate the scalability as well as robustness of *SmartInf* to missing data compared to the baseline algorithms. Further, we dissect *SmartInf* to highlight that the detected influential nodes exposes the tweet to a new and diverse population. In Sec. VII, we delve deep to investigate the key factors behind this performance improvement. First we develop a simple simulation setup to generate synthetic cascades on the follower network. We execute the proposed *SmartInf* algorithm to demonstrate its efficiency in identifying anchor nodes with high retweet rate and facilitating quick diffusion of tweets in a diverse locality.

II. Related works

There has been a significant body of work in the state-of-the-art literature that have addressed the problem of identifying influential nodes on a social network. First, we provide a brief survey on these endeavors. Next, we discuss few attempts that have been made in the gamut of modeling and predicting the popularity of Twitter cascades in terms of retweet count. Finally, we survey works that rely on inter-retweet time intervals of retweet cascades exhibiting distinct temporal patterns to classify users and cascades.

A. Identification of influential users

Identification of influential users has been an important problem in the domain of social networks that has attracted widespread research interest over the years. There have been several endeavors to identify influential users in the network from the underlying topological structure. For instance, few works have quantified user influence in Twitter by proposing different influence measures [13] or using various centrality measures [14]. Some works have identified influential nodes based on their network roles [8], [12]. In addition, Huang et al. [23] have identified influencers in a temporal social network taking into account the network dynamics while Xia et al. [9] have proposed an approach to find influential users with a specific set of characteristics based on advertisers' preference. Few works have quantified influence between user pairs using available propagation traces from historical data or association rules learning to find the most influential users [24].

Another line of research focuses on combining several influence measures capturing diverse structural information to rank influential nodes. For instance, Malliaros et al. [17] have proposed the K-truss decomposition of a graph based on triangle-based extension of k-core decomposition method [16] to identify influential users while Jianqiang et al. [25] have identified influential users in a network by combining user influence based on contribution of tweets as well as their network position based on centrality measures. Further, Madoto et al. [10] have identified super spreader nodes by ranking users

TABLE I: Details of *schematic* datasets

Dataset	#Tweets	#Retweets	#Cascades	#Users	Maximum cascade size
IPL 2018	3884	197210	3884	2.5×10^4	99
15-M	2626	5649951	2626	8.7×10^4	119424
Lady Gaga	1238135	6329596	1238135	1.6×10^5	14130

TABLE II: Details of *comprehensive* datasets

Dataset	#Tweets	#Retweets	#Cascades	#Users	Maximum cascade size
Algeria	65268	17269	5730	1.6×10^4	980
Egypt	671417	188090	67539	7.6×10^6	432
Nepal	26424	521938	26424	2.9×10^5	23864

through combining eight different centrality measures using modified Borda count aggregation. Finally, Sheikhahmadi et al. [18] have combined degree, core number of a node as well as weighted diversity in the core number of friends to identify influential nodes.

There have also been a few works in literature that focus on detecting structural holes [21], [22] acting as bridges that connect separated parts of a social network and thus can be highly influential in propagating information. Sela et al. [26] have proposed a method targeting distinct parts of the network when desynchronized seeding is allowed. All these endeavors have relied on solely using the topological structure to identify influential users in a network. However, to the best of our knowledge, existing works have overlooked the potential of inter-retweet intervals in a cascade that may provide important signatures to identify a better set of influential nodes in a network, when combined with structural information.

B. Modeling cascade popularity

Predicting cascade popularity that corresponds to estimating the final cascade size is a well-studied problem in state-of-the-art literature. For instance, Cheng et al. [27] have studied a sample of large photo reshare cascades on Facebook to predict their future growth. Taxidou et al. [28] have modeled information diffusion in realtime over the social network by analyzing cascades with incomplete information. Pramanik et al. [29] have investigated the role of mentions in modeling tweet popularity while Cheng et al. [30] have characterized the recurrence of large popular cascades on Facebook. Weng et al. [31] have revealed that the initial diversity of a cascade across several network communities is a good predictor for future popularity, whereas Zhang et al. [32] have exploited ego networks to predict the behavior of users. In addition, some works have also built on the theory of self-exciting point processes [33], [34] to develop statistical models to predict tweet popularity. In short, all the aforementioned works have used different types of features and models to estimate cascade popularity.

C. Modeling temporal retweet patterns of cascades

Distinct temporal patterns present in the inter-retweet intervals of a cascade have been exploited to distinguish different user categories [35] as well as to identify different categories of retweeting activity on Twitter [36]. In addition, Bhowmick et al. [37] have leveraged on inter-retweet intervals to detect cascade diffusion across multiple localities of the network. However, all these works have overlooked the potential of using pattern of inter-retweet intervals in a cascade to identify users playing key roles in diffusing information to different parts of the network; such users may serve as an indicator for identifying the influential nodes of a network.

III. Dataset and problem statement

A. Dataset

In this paper, we rely on two types of public datasets; we call the first one as *schematic* dataset which mostly contain the collection of tweets posted during famous events such as *15-M Movement* and *IPL 2018* in addition to tweets posted by celebrities such as *Lady Gaga*. Table I depicts the collection of tweets posted during these events[2], which constitutes the *schematic* dataset. We leverage on the schematic dataset in order to conduct data study and the respective analysis. Precisely, the schematic dataset contains the tweet/retweet ID, ID of the user who posted the tweet/retweet and timestamp of posting the corresponding tweet/retweet. In addition, a retweeted post contains the ID of the original tweet that provides a link to the original tweet, which allows us to identify all the retweet events belonging to the same cascade. Notably, the follower network information is not available in *schematic* datasets.

In addition, we collect the *comprehensive* dataset in order to perform the in-depth data analysis and the evaluation of proposed *SmartInf* algorithm. This dataset contains the collection of tweets (tweet IDs) and users who posted them during the Arab-Spring movement[3] (*Algeria and Egypt datasets*) [38] and the *2015 Nepal earthquake*[4]. Similar to *schematic* dataset, we crawled the tweet content with respective tweet/retweet ID, profile of the user who posted the tweet/retweet, timestamp of posting the corresponding tweet/retweet and ID of original tweet for a retweeted post. This allows us to identify all retweet events belonging to the same cascade. In addition, we have also crawled from Twitter the corresponding follower network of participating users who tweeted/retweeted in some cascade. Salient features of these datasets are summarized in Table II. Even though the *comprehensive* datasets are relatively small in size, they have the typical advantage of providing information of (re)tweet diffusion and the underlying social (follower) network, which makes them rich and expensive to collect.

B. Notations and representations

We represent each cascade in the dataset as a sorted sequence of retweet events based on their time of posting. For a cascade C of size n_C originated by user u_0^C at time t_0^C, we have the time series of retweets ordered based on timestamps denoted by $(t_0^C, t_1^C, , \ldots, t_{n_C}^C)$ with the corresponding list of retweeting users $(u_0^C, u_1^C, , \ldots, u_{n_C}^C)$. Here r_i^C denotes the i^{th} retweet of the cascade C which has been posted by user u_i^C at time t_i^C. Given this time series, we define the sequence of inter-retweet time intervals denoted by

[2] http://www.cnergres.iitkgp.ac.in/blog/2019/03/08/twitter-cascade-dataset/

[3] http://www.cnergres.iitkgp.ac.in/blog/2018/02/28/arab-spring-twitter-dataset/

[4] http://crisisnlp.qcri.org/lrec2016/content/2015_nepal_eq.html

$T^C = (T_0^C, T_1^C, \ldots, T_{n_C-1}^C)$ for the cascade C as the time interval between two consecutive retweets in C such that the i^{th} inter-retweet time interval is computed as $T_i^C = t_{i+1}^C - t_i^C$. Additionally, the reaction time μ_i of a user u_i^C retweeting at t_i^C in C may be defined as the time interval between receiving the tweet from her friend (say u_j^C at time $t_j^C < t_i^C$) and retweeting it; hence, $\mu_i = t_i^C - t_j^C$.

The Twitter follower network can be represented as a directed graph $G = (U, E)$ where $U = \{u_1, u_2, \ldots, u_N\}$ is the set of users and $E = \{(u_i, u_j) : u_i, u_j \in U\}$ is the set of directed links from u_i to u_j denoting the who-follows-whom relationship between users. $F(u)$ denotes the set of followers of a user $u \in U$.

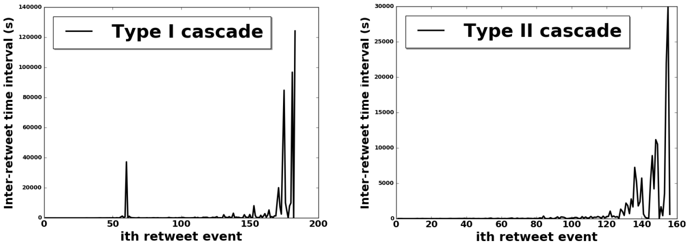
C. Construction of cascades and follower network

In order to construct the cascades, we first extract the tweet/retweet ID, ID of the user who posted the tweet/retweet and the timestamp of posting, from the collected *schematic* and *comprehensive* datasets. Next, we distinguish a post as tweet or retweet based on the *original tweet ID* field of the post. If the *original tweet ID* field shows -1, the post is designated as a tweet, otherwise it is a retweet. In case of a retweet, the *original tweet ID* field of the post links it with the original tweet. Subsequently, we extract all retweets belonging to the same tweet r_0^C and construct the cascade C formed by that tweet. Next, we sort all retweet events r_i^C belonging to a cascade C in increasing order of their timestamps t_i^C to form the temporal sequence of retweets for a cascade C. The users u_i^C posting retweets in the cascade C forms the sequence of users participating in cascade C. Finally, we preprocess the dataset by filtering out all small-sized cascades with number of retweets < 10.

In order to create the underlying follower network from the *comprehensive* dataset, we crawl the list of followers of each participating user who posted a tweet/retweet. Given the follower set $F(u)$ of a user u, we form a directed link from every follower to the user u. We repeat this process of forming directed links for every participating user in a cascade across all cascades in the dataset. Finally, we obtain the complete follower network G for that dataset.

D. Problem statement

In this paper, our primary objective is to identify a ranked list of influential users $\mathscr{S}$ in a social network G from temporal retweet sequence T^C of cascades, which can spread the information to a large population in G. Given the sequence of inter-retweet time intervals T^C and corresponding list of retweeting users $(u_0^C, u_1^C, , \ldots, u_{n_C}^C)$ for a cascade C, as well as follower network G, our goal is to identify a ranked seed set $\mathscr{S} \subseteq U$ of size β, such that the mean exposed population $V(\mathscr{S})$ is maximized at the end of the diffusion process, when $\mathscr{S}$ is targeted as a seed set. Mathematically, our problem can be formulated as identifying the ranked influencer set $\mathscr{S}$ such that for top-k influencers $\mathscr{S}^k \subseteq \mathscr{S}$, $\forall k$, we obtain $max(V(\mathscr{S}^k))$.



(a) Type I cascade

(b) Type II cascade

Fig. 1: Pattern of inter-retweet intervals for Type I & Type II cascades based on presence or absence of peaks in intermediate phase for *Algeria* dataset

IV. SmartInf: Methodology for identification of influential nodes

In this section, we propose *SmartInf* (stands for *Smart Influencer*), a fast, unsupervised algorithm to identify the *influential nodes* in a social network. First, we establish the key intuition behind the proposed methodology from the empirical data. Leveraging on this intuition, we develop *SmartInf*[5] which has the following two phases. First, we constitute the ranked set of influential nodes only from the temporal retweet sequence T^C. Next, we refine this set to diversify the population exposed through these influential nodes.

A. Key intuition: Influential node & cascade retweet sequence T^C

In the following, we illustrate the role played by the temporal sequence of retweets in identifying influential nodes.

Observing peaks in T^C: We start with the rigorous analysis of the inter-retweet intervals T^C of cascades C observed in the *schematic* and *comprehensive* datasets, shown in Tables I and II respectively. Close inspection of T^C reveals that there exists a *peak* in T^C for some of the cascades, designating a significantly large inter-retweet time interval between two consecutive retweet events. Depending on the phase of occurrence of the *first peak* in T^C, we classify cascades into the following two types [37] (see Fig. 1 for illustration):

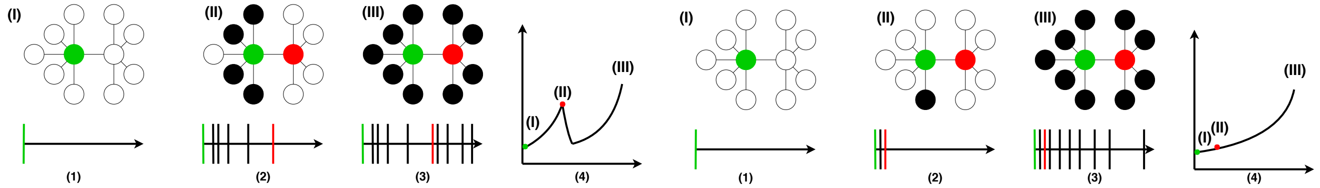
Type I cascades: Characterized by the occurrence of its first peak at the intermediate phase (early peak) of their sequence of inter-retweet time intervals T^C, much before the occurrence of its last retweets (see Fig. 1(a)).

Type II cascades: Characterized by the absence of a peak at the intermediate phase of T^C. The first peak occurs when the last few retweet events take place (late peak), towards the end of the cascade diffusion (see Fig. 1(b)).

Table III shows the fraction of Type I and Type II cascades observed across various datasets. From this table, we observe that datasets such as *Algeria*, *Egypt*, *15-M* and *IPL 2018* have majority of cascades classified as Type I. On the other hand, the *Nepal* and *Lady Gaga* datasets have $< 40\%$ of Type I cascades. However, both these datasets consists of large number of cascades implying that the number of Type I cascades for them is quite substantial even though majority of cascades are classified as Type II.

Formation of Type I and Type II cascades: Intuitively, the dynamics of Type I and Type II cascades may be explained

[5] https://github.com/ayan-0305/SmartInf/



(a) Dynamics of Type I cascades. First, the seed tweets (I, in green). When the tweet saturates a locality (retweeting nodes shown in black in II), the inter-retweet intervals increase (II). When it gets exposed to a new locality through an anchor node (in red), a new wave of diffusion occurs in the second locality, resulting in a fall of the inter-retweet intervals (III).

(b) Dynamics of Type II cascades. First, the seed tweets (I, in green). When the tweet gets exposed to a new locality before saturating the current locality through the node (in red), the inter-retweet intervals remain low (II). When most retweet events have occurred from users in both localities (retweeting nodes shown in black), it results in a rise of the inter-retweet intervals (III).

Fig. 2: Dynamics behind Type I and Type II cascades. Below each phase of the cascade diffusion process, the time series of the cascade is displayed, where each stroke on the timeline corresponds to a retweet. The corresponding sequence of inter-retweet intervals is displayed in (4).

TABLE III: Fraction of Type I and Type II cascades

Dataset	Type I %	Type II %	Dataset	Type I %	Type II %
Algeria	80	20	IPL 2018	58	42
Egypt	61	39	Lady Gaga	37	63
Nepal	38	62	15-M	56	44

through the diffusion of tweets across multiple localities. Fig 2(a) demonstrates the diffusion of a Type I cascade and its corresponding inter-retweet intervals. Once posted, a tweet initially gains popularity & is frequently retweeted by different users within its locality. Hence, the interval between two consecutive retweet events is initially low, as retweets may occur in close succession from users in that locality (see Fig. 2(a)(1)). However, as the locality gets saturated with the redundant tweet content, the post *rarely* gets retweeted in that locality, which in turn increases the inter-retweet intervals. Hence, the saturation in the diffusion locality [37] is reflected by the surge in inter-retweet time intervals. In this scenario, a retweet caused by the *anchor node* may diffuse the post to a new diffusion locality (Fig. 2(a)(2)). The exposure of the new piece of information in this second locality triggers frequent retweets, dropping the inter-retweet time intervals (Fig. 2(a)(3)). In this type of diffusion process, the pattern of inter-retweet time intervals correspond to that of a Type I cascade. The early peak and drop in case of Type I cascades denote a saturation followed by the migration of the tweet to multiple diffusion localities (Fig. 2(a)(4)). On the contrary, in case of a Type II cascade, the tweet migrates to the second diffusion locality, even before saturating the first locality (see Fig. 2(b) for illustration). Hence, no peak and drop pattern appear in the corresponding sequence of inter-retweet time intervals for Type II cascades (Fig. 2(b)(4)).

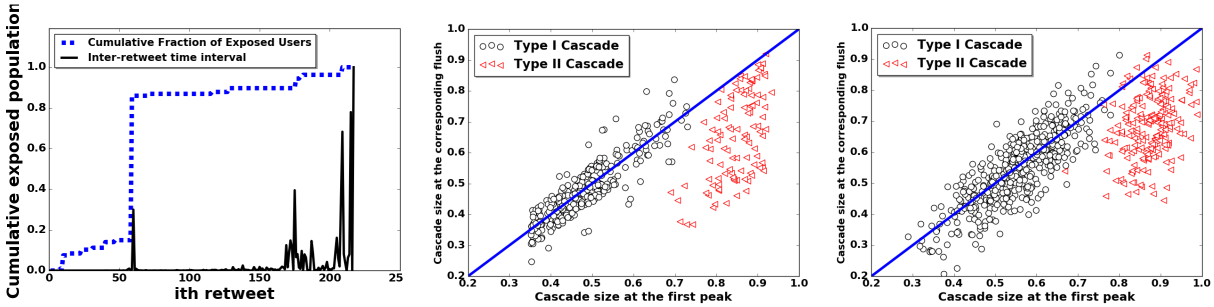
Peaks as an indicator of influential nodes: Empirically, we observe that the first peak in Type I cascades reflects the sudden exposure of the tweet to a new diffusion locality. We define a user u in the network as an exposed user with respect to a cascade C if u is a follower of at least one retweeting user in cascade C. Let M^C denote the entire exposed population for the cascade C while M_i^C is the set of exposed users after i^{th} retweet r_i^C. The ratio $P_i^C = \frac{|M_i^C|}{|M^C|}$ corresponds to the fraction of users in M^C who have been exposed to C after r_i^C; thus, $P^C = (P_1^C, \ldots, P_n^C)$ is the sequence of cumulative fraction of users exposed after each retweet. We illustrate a typical Type I cascade in Fig. 3(a), where the black line denotes the sequence of inter-retweet time intervals T^C of cascade C and the dotted blue line denotes the respective sequence of cumulative fraction of users P^C exposed to the tweet after every retweet event i. Notably, in Fig. 3(a), there exists a sudden rise in the exposed population P^C close to the first peak in the corresponding sequence of inter-retweet intervals T^C. We call this sudden rise a *flush*, indicating a sudden exposure of the cascade to a whole new population (diffusion locality) after retweet by a specific user u_p^C, called the *anchor node* for the cascade C. Interestingly, first peaks and flushes tend to co-occur together in all Type I cascades (see Fig. 3(b) and 3(c)), within a short time shift. Fig. 3(b) and 3(c) show the high correlation between phase of occurrence of the first peak and corresponding flush across all Type I cascades in *Algeria* and *Egypt* datasets respectively while the correlation is low for Type II cascades as first peaks of Type II cascades are not accompanied by any flush; this co-occurrence is specific to Type I cascades. In our dataset, we observe this co-occurrence for 82% of Type I cascades in *Algeria* (71% for *Egypt*), when allowing a maximal shift of three retweet events.

B. Phase 1: Detection of influential nodes from sequence of inter-retweet time intervals T^C

Leveraging on the aforesaid observations, we propose the *SmartInf* algorithm to detect influential nodes from retweet sequence of a cascade as follows:

Step 1: Peak detection: We apply a simple outlier detection technique [39] on the distribution (lognormal) obtained from the sequence of inter-retweet intervals T^C to detect peaks for cascade C. Let μ_{T^C} and σ_{T^C} denote the mean and standard deviation of the distribution obtained from the sequence T^C. We classify an interval T_i^C as a peak if $T_i^C > \mu_{T^C} + 2 * \sigma_{T^C}$. The usage of this threshold ensures the handling of noise in interval data when detecting peaks that could be due to the large variance of the distributions in play. Moreover, we apply the similar outlier detection technique to detect potential flushes from the distribution of cumulative fraction of exposed users P^C in a cascade C.

Step 2: Identifying potential influencers: From the sequence of retweets in a Type I cascade C, we identify the potential influential nodes I^C as follows. Consider the first peak in cascade C where user u_p^C retweeting at time t_p^C causes



(a) Co-occurrence of first peak in inter-retweet time intervals and flush in cumulative exposed population for a typical Type I cascade in *Algeria* dataset.

(b) Co-occurrence of flushes and peaks appear for Type I cascades in *Algeria* dataset

(c) Co-occurrence of flushes and peaks appear for Type I cascades in *Egypt* dataset

Fig. 3: Co-occurrence of flush effect and first peaks observed for Type I Cascades only across *Algeria* and *Egypt* datasets

Algorithm 1 Ranking influential nodes to obtain $\mathcal{T}$

```
procedure RANKED LIST OF INFLUENCERS $\mathcal{T}$
Input: Inter-retweet time intervals $T^C$ for all Type I cascades $C \in \mathbf{T}$, set of all users $U$ in the follower network
Output: Ranked list of influencers $\mathcal{T} = (u_1, \ldots, u_\alpha)$
    for all $u \in U$ do
        Set $f(u) = 0$    ▷ $f(u)$ denotes frequency of appearance of user $u$ in activation period $\delta$ across all Type I cascades $\mathbf{T}$
    end for
    Set $\sqcup = \emptyset$
    for $C \in \mathbf{T}$ do
        $t_p^C, t_q^C = PeakDet(T^C)$ ▷ Apply outlier detection as described in Step 1 of Sec. IV-B
        Set $I^C = \emptyset$
        for all $u_i^C \in C$ do
            if $t_p^C \le t_i^C \le t_q^C$ then  ▷ Denotes $u_i^C$ retweets within activation period $\delta^C = t_q^C - t_p^C$
                Set $I^C = I^C \cup u_i^C$
                Set $f(u_i^C) = f(u_i^C) + 1$
            end if
        end for
        Set $\sqcup = \sqcup \cup I^C$
    end for
    Sort $\sqcup$ in non-increasing order of $f(u)$ to obtain the ranking $\mathcal{T} = (u_1, \ldots, u_\alpha)$ such that $f(u_1) \ge f(u_2) \ge \ldots \ge f(u_\alpha)$
end procedure
```

the rise, and the user u_q^C retweeting at t_q^C causes the fall in the sequence of inter-retweet intervals T^C. Since the first peak is usually accompanied by a flush in the cumulative fraction of exposed users P^C, we claim that the anchor nodes, escalating the diffusion, are present in the interval $\delta^C = t_q^C - t_p^C$. We designate the activation period δ^C of cascade C as the time interval between the first rise and the subsequent fall observed in T^C, and the users retweeting within this interval are denoted as the potential influential nodes I^C. Notably, δ^C may vary across the cascades C. Considering all the Type I cascades $\mathbf{T}$ in the dataset, we obtain the set of potential influential nodes $\sqcup$ of size α as $\sqcup = \bigcup_{C \in \mathbf{T}} I^C$. While constructing the set $\sqcup$, we keep track of the frequency of appearance of each user $u \in \sqcup$ in I^C.

Step 3: Ranked list of influencers $\mathcal{T}$: The idea behind Step 2 is to identify the set of users I^C, truly responsible for the migration of a Type I cascade C to a new diffusion locality, as they retweet within the activation window δ^C. However, few sporadic users, who do not play any role in the migration of the cascade, may also appear in I^C and therefore get unfairly favoured. However, the presence of such sporadic users in I^C is quite irregular across different cascades C; nevertheless, the truly influential users are likely to be consistently present in I^C, since they retweet more often around the peak of Type I cascades. Hence, in Step 3, we rank the set of users $\sqcup$ according to their frequency of appearance in I^C; as a result, truly influential users should be ranked high. This procedure will filter out the unfairly favoured users, who sporadically appear in I^C, placing them at the bottom of the ranked list $\mathcal{T}$. Hence, in this phase, we obtain the preliminary ranked list of influential nodes $\mathcal{T}$ by only relying on the temporal sequence of retweets. This phase is completely follower network agnostic.

C. Phase 2: Constructing the final ranked influencer list $\mathscr{S}$ from $\mathcal{T}$

In the second phase, we exploit the available follower information $F(u)$ of node $u \in \mathcal{T}$ to refine the preliminary ranked list of influential nodes $\mathcal{T}$. Precisely, we refine the ranked influencer list $\mathcal{T}$ to obtain $\mathscr{S}$ of size $\beta \le \alpha$ to concentrate on the set of influential nodes with capacity to reach a diverse population. The purpose of this refinement phase is to eliminate those influential users from $\mathcal{T}$ who have high follower overlap with other users in $\mathcal{T}$, resulting in the refined set $\mathscr{S}$ in which users have minimum follower overlap.

To obtain $\mathscr{S}$, we implement a variation of the *set cover problem* [40] that aims to cover all elements of a universal set using a minimum number of subsets. In our context, the problem consists of checking whether every node, chosen in order of their position in the ranked list $\mathcal{T}$ allows the content to be exposed to some new users. Following the influencers in order of the ranked list $\mathcal{T}$ at each step, we incrementally populate the list $\mathscr{S}$ by checking whether the corresponding influencer u in $\mathcal{T}$ *newly* exposes the content to atleast one of its followers (hence confirming that not all the followers of u were already exposed to the content by previous influencers in $\mathscr{S}$). This ensures that influential nodes in the refined list $\mathscr{S}$ of size β cover a diverse population in the network. Algorithm 2 describes the exact procedure to obtain $\mathscr{S}$ denoting influential nodes recommended by *SmartInf*.

D. Computational complexity

In case of *SmartInf*, the detection of peaks as well as the identification of potential influential users belonging to the set $\sqcup$ have the complexity $O(|\mathbf{T}|)$ where $\mathbf{T}$ is the set of Type I cascades. Algorithm 1 computes the frequency of appearance of users in $\sqcup$. For each cascade $C \in \mathbf{T}$, it updates the frequency of appearance for the ν^C users retweeting during the activation window δ^C. Thus, the time required for updating this frequency across all cascades in $\mathbf{T}$ is $\nu^C|\mathbf{T}| \sim O(|\mathbf{T}|)$. Further, sorting the users in $\sqcup$ to obtain $\mathcal{T}$ has the complexity $O(\alpha \log \alpha) \sim O(N \log N)$ since $|\mathcal{T}| = \alpha$ and $\alpha \leq N$. Thus, the total time complexity of Algorithm 1 is $O(|\mathbf{T}|)+O(N \log N)$. In Algorithm 2, we check whether each user in the ranked list $\mathcal{T}$ exposes any new follower with the complexity $O(\alpha) \sim O(N)$. So, the overall time complexity of *SmartInf* is $O(|\mathbf{T}|) + O(N \log N)$.

V. Experimental Setup

In this section, we first introduce the state-of-the-art baseline algorithms for finding influential users in a network. Next, we describe the implementation details and the procedure for evaluating the performance of the proposed *SmartInf* algorithm against the various baseline algorithms.

A. Baseline algorithms

We implement the following competing algorithms for identifying the set of influential users in a network:

1) **MCDWE score:** This method [18] combines the core number of a node, its degree and its neighbours' diversity based on k-shell decomposition to rank users.
2) **K-truss decomposition:** This [17] is a triangle-based extension of the k-core decomposition of graphs that extracts a denser subgraph compared to k-core [16]; it is structurally closer to a clique achieving faster and wider epidemic spreading.
3) **MCDWE-Activity:** This is a *hybrid method* that combines activity rate and structural information based on MCDWE scores [18] of users. In this method, we simply compute the product of frequency of retweets and MCDWE scores of all users; we then sort users in decreasing order of this product to get the ranking.
4) **SmartInf-Temp:** This is a variation of our proposed *SmartInf* algorithm. *SmartInf-Temp* implements only the phase 1 of **SmartInf** algorithm (Algorithm 1) and generates the ranked list of influential nodes $\mathcal{T}$. Thus, this method obtains a list of influential nodes purely from temporal retweet sequence T^C of Type I cascades.

Algorithm 2 Refining ranked list $\mathcal{T}$ to obtain $\mathscr{S}$

```
procedure REFINED LIST OF INFLUENCERS $\mathscr{S}$
Input: Follower network $G = (U, E)$ and ranked list of influencers $\mathcal{T} = (u_1, \ldots, u_\alpha)$
Output: Refined list of influencers $\mathscr{S} = (u_1, \ldots, u_\beta)$
    Set $\mathscr{S} = \emptyset$, $Exposed = \emptyset$, $ind = 0$
    $F(u_i) \leftarrow$ set of followers of user $u_i$
    while $ind < |\mathcal{T}|$ and $|Exposed| < |U|$ do
        $ind = ind + 1$
        if $|\{u_{ind} \cup F(u_{ind})\} \setminus Exposed| > 1$ then
            $\mathscr{S} = \mathscr{S} \cup u_{ind}$    ▷ Insert $u_{ind}$ into $\mathscr{S}$ if it exposes any new follower
            $Exposed = Exposed \cup \{u_{ind} \cup F(u_{ind})\}$
        end if
    end while
end procedure
```

B. Evaluation procedure

We evaluate the quality of influential users, obtained from *SmartInf* as well as the competing algorithms, from the following three perspectives. First, we introduce a suite of metrics which measures the quality of the influential nodes from multiple perspectives. We normalize the value of a metric for a user u by the corresponding size of the cascades in which u retweeted. Higher metric value indicates stronger influence. Secondly, we consider the set of influential nodes as seeds of the network and conduct numerical simulations following standard epidemic models to compute the volume of final infected population. Seed nodes, resulting in higher infected population indicate better influential nodes. We implement *SmartInf* as well as the baseline algorithms in *Python* programming language. We execute these algorithms as well as perform their evaluation on *Ubuntu 16.04.5 LTS* server machine with 2.20 GHz CPU processing speed, 128 GB memory and the kernel used is *Linux 4.4.0-141-generic*.

1) Classical metrics to compute node influence:: We consider the standard centrality metrics such as (a) Pagerank centrality (**PR**) [41], (b) Eigenvector centrality (**EV**) [42] and (c) Betweenness centrality (**BW**) [43] to evaluate the performance of the influential nodes recommended by *SmartInf* algorithm. The aforesaid metrics essentially capture the structural centrality of the influential nodes.

2) Twitter specific metrics to compute node influence: **(a) Gain in retweet count ($\mathcal{R}^u$):** This metric measures the impact of retweets of a given user (say u) on the final size of the cascade [43], [44]. Consider a cascade C of size n_C where k_C^{th} retweet has been posted by the user u_k^C. This metric assumes that each of the $n_C - k_C$ new retweets occurring after the k_C^{th} retweet is equally contributed by the first k_C users. Hence, the gain in size of the cascade C due to the retweet posted by user u_k^C is defined as $\frac{n_C - k_C}{k_C}$. Finally, the *Gain in retweet count* $\mathcal{R}^u$ for the user u is computed as the average gain over all the cascades (N^u) in which u participates

$$\mathcal{R}^u = \frac{1}{|N^u|} \sum_{C \in N^u} \frac{n_C - k_C}{k_C n_C} \tag{1}$$

TABLE IV: Mean score of the influence metrics for sets of top-k influential nodes taking $k = 10, 20, 50$ on *Algeria dataset*

Number of seeds	**k=10**				**k=20**				**k=50**			
Algorithm	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$
SmartInf	**0.480**	**0.054**	**0.350**	0.036	**0.470**	0.036	**0.400**	0.030	**0.450**	0.017	**0.360**	0.010
SmartInf-Temp	0.450	0.041	0.330	**0.040**	0.420	**0.066**	0.320	**0.050**	0.420	**0.036**	0.280	**0.040**
MCDWE	0.080	0.003	0.090	0.002	0.090	0.004	0.070	0.003	0.060	0.002	0.050	0.002
K-truss	0.226	0.030	0.100	0.006	0.230	0.010	0.120	0.005	0.190	0.006	0.110	0.003
MCDWE-Activity	0.360	0.037	0.280	0.028	0.310	0.028	0.150	0.014	0.220	0.013	0.100	0.006

TABLE V: Mean score of the influence metrics for sets of top-k influential nodes taking $k = 10, 20, 50$ on *Egypt dataset*

Number of seeds	**k=10**				**k=20**				**k=50**			
Algorithm	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$
SmartInf	**0.405**	**0.210**	**0.300**	**0.092**	**0.340**	0.040	**0.310**	0.010	0.350	0.020	**0.300**	0.010
SmartInf-Temp	0.318	0.058	0.261	0.006	0.310	**0.200**	0.260	**0.080**	**0.390**	**0.150**	0.270	**0.060**
MCDWE	0.064	0.002	0.006	0.002	0.040	0.001	0.010	0.001	0.020	0.009	0.010	0.004
K-truss	0.122	0.005	0.060	0.004	0.110	0.003	0.050	0.002	0.120	0.004	0.040	0.001
MCDWE-Activity	0.167	0.009	0.022	0.012	0.160	0.007	0.080	0.004	0.150	0.010	0.070	0.005

TABLE VI: Mean score of the influence metrics for sets of top-k influential nodes taking $k = 10, 20, 50$ on *Nepal dataset*

Number of seeds	**k=10**				**k=20**				**k=50**			
Algorithm	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$	$\mathcal{R}^u$	$\mathcal{I}^u$	$\mathcal{E}^u$	$\mathcal{H}^u$
SmartInf	**0.340**	**0.030**	**0.200**	**0.020**	**0.270**	**0.030**	**0.180**	**0.020**	**0.250**	0.020	**0.160**	**0.020**
SmartInf-Temp	0.220	0.020	0.140	0.010	0.230	**0.030**	0.140	**0.020**	0.230	**0.030**	0.140	**0.020**
MCDWE	0.170	0.0009	0.150	0.0004	0.150	0.006	0.120	0.004	0.160	0.006	0.140	0.004
K-truss	0.210	0.006	0.130	0.004	0.180	0.008	0.110	0.005	0.190	0.007	0.120	0.004
MCDWE-Activity	0.200	0.015	0.140	0.013	0.210	0.016	0.140	0.010	0.200	0.018	0.140	0.014

TABLE VII: Comparison of mean score of the classical centrality metrics for top-10 influential nodes recommended by *SmartInf* and randomly selected top-10 nodes

Algorithm	**PR** (10^{-5})	**EV** (10^{-5})	**BW** (10^{-5})
SmartInf (Algeria)	53.9	600	100
Random (Algeria)	0.583	10	0.0001
SmartInf (Egypt)	1.96	78.6	9.83
Random (Egypt)	0.0002	0.000002	0.00003
SmartInf (Nepal)	8.9	40	4000
Random (Nepal)	0.031	0.0188	20

where n_C is the size of a cascade C in which u retweeted as the ${k_C}^{th}$ user. Thus, this metric, which is agnostic to the underlying follower network structure, favors users retweeting in the early stage of large cascades, compared to the users (a) involved in small cascades, or (b) participating in the later stage of the large cascades. These favoured users act as precursors of the popularity of the tweet; hence, they should be preferentially targeted for viral diffusion.

(b) Gain in exposed user count ($\mathcal{E}^u$): This metric measures the volume of newly exposed population a_C, who retweeted the content in the cascade C only due to retweet posted by a specific user (say u) [43], [44]. The exposure to the content may be caused directly or indirectly via the follower links of u. The *Gain in exposed user count* $\mathcal{E}^u$ for user u over all the cascades (N^u) in which u participated can be expressed as

$$\mathcal{E}^u = \frac{1}{|N^u|} \sum_{C \in N^u} \frac{a_C}{k_C n_C} \tag{2}$$

where a_C denotes volume of population who got newly exposed by u (the k_C^{th} retweeting user in C) as well as retweeted in cascade C. This metric scores high for those influential users, who tend to be responsible for the first exposure of the tweet to many of the retweeting users in a cascade.

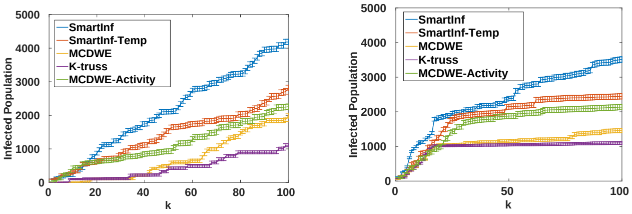
(c) Active gain in retweet count and exposed user count ($\mathcal{I}^u$ and $\mathcal{H}^u$): These metrics take the activity (retweet) rate $\mathcal{P}^u$ of user u into consideration and combine $\mathcal{P}^u$ along with the aforesaid two metrics $\mathcal{R}^u$ and $\mathcal{E}^u$ respectively. Precisely, we define

$$\mathcal{I}^u = \mathcal{P}^u.\mathcal{R}^u \tag{3}$$

$$\mathcal{H}^u = \mathcal{P}^u.\mathcal{E}^u \tag{4}$$

where the retweet rate $\mathcal{P}^u$ is estimated empirically from the cascades in which u participated. In principle, these metrics score high for the users who are more likely to retweet the tweet content as well as whose gains $\mathcal{R}^u$ and $\mathcal{E}^u$ are high.

3) Epidemic simulation to measure node influence: We implement the standard susceptible-infected (SI) model of epidemic simulation [45] to evaluate the quality of a set of influential nodes in the network. In this simulation, initially all the nodes in the network are in susceptible (S) state. A susceptible (S) node can change its state to infected (I) by (re)tweeting the initial post. We consider the recommended set of influential nodes as seeds of the network. First, we infect the seed nodes; here, this transition of the seeds from susceptible (S) to infected state (I) indicates tweeting the initial post. Once a susceptible node in the network becomes infected, its followers, who are in susceptible state, get exposed to the post, and in the next step may change their state to infected by retweeting the post. This process continues until no new susceptible node in the network can be infected. Each susceptible follower node u gets a single chance to change its state to infected, depending on its probability to get infected β_u (since we discard the re-exposure phenomenon). The infection probability β_u varies across the users $u \in U$ in the network, and is computed empirically as the fraction of cascades in which u retweeted, against all cascades where she got exposed to the tweet. The average volume of the final infected population over 1000 realizations denote the quality of the recommended list of influential nodes (seeds).



(a) *Algeria* dataset

(b) *Egypt* dataset

Fig. 4: Spreading power for vaying number of top-k influential nodes on empirical networks detected by different methods (errorbars indicate 2 standard deviations)

VI. Evaluation on empirical data

In this section, we evaluate the performance of *SmartInf* on the empirical follower network & the cascades obtained from the *comprehensive* dataset (*Algeria*, *Egypt* & *Nepal* datasets). Next, we provide some insightful observations highlighting the benefits of influential nodes obtained using *SmartInf*. Finally, we demonstrate the scalability of *SmartInf* as well as show its robustness in the presence of missing data.

A. Evaluation based on classical centrality metrics

We compute the average scores of the three classical centrality metrics **PR**, **EV** and **BW**, introduced in Sec. V-B1 for the top-10 influential users recommended by *SmartInf* vis-a-vis 10 randomly chosen users of the respective datasets. Here we stochastically select the 10 users, repeat the experiment for 500 iterations and compute the average centrality scores. From Table VII, we observe that the top-10 influential nodes identified by *SmartInf* exhibit higher centrality scores compared to the randomly selected nodes, consistently across *Algeria*, *Egypt* and *Nepal* datasets. This implies that although *SmartInf* does not explicitly filter the nodes with highest centrality scores, however, the influential users recommended by *SmartInf* exhibit decently high centrality scores.

Subsequently, we investigate if the nodes with top centrality score get recommended by *SmartInf*. In Table VIII, we show a comparison between average scores for top-10 users ranked based on the classical centrality metrics vis-a-vis average centrality scores of the top-10 influential users recommended by *SmartInf*. In this table, we observe that for all three classical centrality metrics, the average centrality scores of the top-10 *SmartInf* recommended users are lower than the average scores of top-10 users ranked based on the respective centrality metrics. This implies that the nodes with the highest centrality scores are not always recommended by *SmartInf*. Close investigation reveals that such structurally central nodes may remain (i) inactive in posting retweets or (ii) mostly participate in non-popular cascades. Hence, those central nodes may not play a key role in migration of the cascade to a different diffusion locality. Consequently, although they possess a structurally favourable position in the follower network, they are not recommended by *SmartInf*.

TABLE VIII: Comparison of mean score of the classical centrality metrics for top-10 influential nodes recommended by *SmartInf* and top-10 nodes with highest centrality scores

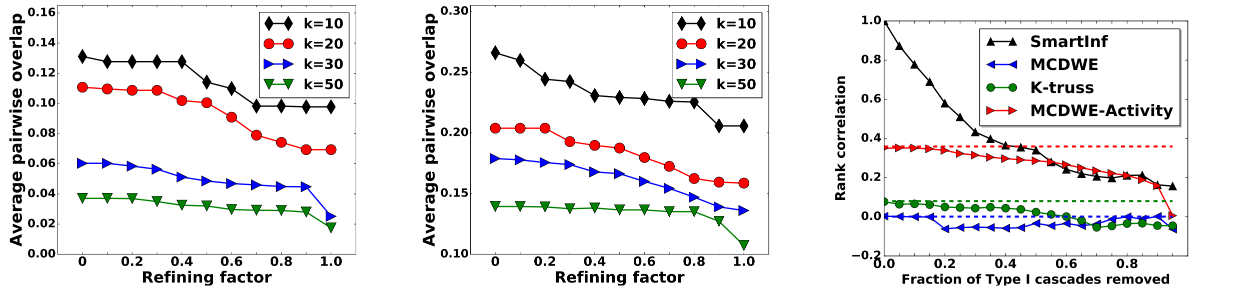
Measure	**PR** (10^{-5})		**EV** (10^{-5})		**BW** (10^{-5})	
Dataset	***SmartInf***	***Top-PR***	***SmartInf***	***Top-EV***	***SmartInf***	***Top-BW***
Algeria	53.9	100	600	6000	100	640
Egypt	1.96	50	78.6	900	9.83	120
Nepal	8.9	40	40	620	400	1200

B. Evaluation based on Twitter specific influence metrics

We obtain the top-k recommended influential nodes $\mathscr{S}^k$ from *SmartInf* as well as from the baseline algorithms. For each recommendation algorithm, we compute the average score of the metrics proposed in Sec. V-B2 for the identified top-k (for $k = 10, 20, 50$) influential nodes. In Tables IV and V, we observe that both the *Gain in retweet count* $\mathcal{R}^u$ and *Gain in exposed user count* $\mathcal{E}^u$ attain highest average scores for top-k influential nodes $\mathscr{S}^k$ identified by *SmartInf* (and the variation *SmartInf-Temp*) across both Algeria and Egypt datasets respectively. This indicates that influential users detected by *SmartInf* and *SmartInf-Temp* mostly retweet in the early part of large cascades, as well as responsible for exposing the tweet to a large fraction of retweeting users, compared to the influential nodes identified by the baselines. Side by side, in case of *SmartInf* and *SmartInf-Temp*, the high retweet rate of the recommended influential nodes results in high active gain in retweet count ($\mathcal{I}^u$) and exposed user count ($\mathcal{H}^u$). Since the baseline algorithms (*MCDWE* and *K-truss*) mostly rely on network structure, the average scores of $\mathcal{R}^u$ and $\mathcal{E}^u$ are low, as influential nodes identified by such methods either retweet in small cascades or towards the later part of large cascades. Moreover, some of these influential nodes do not even participate in retweeting activity. Hence, the values of $\mathcal{I}^u$ and $\mathcal{H}^u$ are very poor for the baselines. Notably, *MCDWE-Activity* performs better than *MCDWE* and *K-truss* across all the metrics since it combines retweet rate of users with structural information. However, it performs worse than *SmartInf* and *SmartInf-Temp* since corresponding influential nodes with high retweet rate in case of *MCDWE-Activity* do not retweet early in large cascades and expose the tweet to a limited fraction of retweeting users.

C. Evaluation based on epidemic simulation

Considering the recommended influential nodes as seeds, we conduct the epidemic simulation as described in Section V-B3, to obtain the final infected population. In Fig. 4(a) and 4(b), we show the average volume of final infected population over 1000 realizations for varying number (top-k) of seed nodes $\mathscr{S}^k$ in Algeria and Egypt datasets respectively. We observe that the seed nodes corresponding to *SmartInf* achieves the largest infected population, compared to the baselines. This result demonstrates the spreading capacity of the influential nodes identified by the proposed *SmartInf* algorithm. As we increase the number of seed nodes (k), we observe that the final volume of infected population increases quite steadily for *SmartInf*. On the other hand, for baselines, the volume of infected population increases slowly for smaller values of k and then saturates for larger k, signifying that there is increasing overlap in the infected population when the number of seeds (k) increases. We can also observe the improvement in performance of *SmartInf* over its variation *SmartInf-Temp*; this



(a) Evidence of decreasing follower overlap with increase in refining factor among influential nodes detected by *SmartInf* in *Algeria* dataset

(b) Evidence of decreasing follower overlap with increase in refining factor among influential nodes detected by *SmartInf* in *Egypt* dataset

(c) Robustness of *SmartInf* against removal of Type I cascades in *Algeria* dataset

Fig. 5: Impact of refining factor on performance and robustness of *SmartInf*

depicts the significance of the refinement phase that minimizes the overlap in population infected by different seeds, thus reaching out to a diverse audience.

D. *Dissecting SmartInf*

In this section, we investigate the factors responsible for the superior performance of *SmartInf* over the baseline algorithms. For this purpose, we evaluate the quality of influential nodes obtained using *SmartInf* from different aspects.

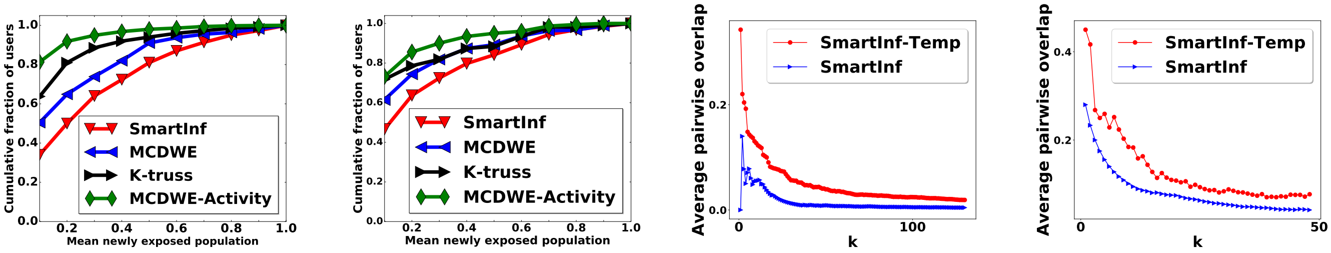
(a) Exposing the new diffusion locality: First, we show that influential nodes $\mathscr{S}$ detected by *SmartInf* exposes the tweet to a new diffusion locality, compared to baselines, when they retweet in a cascade. This establishes the role of detected influential nodes as anchor nodes introduced in Sec. IV. In order to compute the newly exposed population $F_{\mathscr{N}}(u_i^C)$ for an influential node u_i^C (retweeting at time t_i^C) in cascade C, we discount all the followers $\bigcup(F(u_j^C))$ of preceding users u_j^C retweeting at time $t_j^C < t_i^C$, from the follower set of u_i^C denoted as $F(u_i^C)$. The remaining users $F_{\mathscr{N}}(u_i^C) = F(u_i^C) \setminus \bigcup(F(u_j^C))$ constitute the newly exposed population of u_i^C. Subsequently, we compute the mean volume of newly exposed population for the influential node across all the cascades where she retweets. In Figs. 6(a) and 6(b), we plot the cumulative distribution of this mean newly exposed population for influential nodes obtained using *SmartInf* as well as the baselines in *Algeria* and *Egypt* datasets respectively. We observe that a significant fraction (38% for *Algeria* and 28% for *Egypt*) of top-k influential nodes $\mathscr{S}^k$ for *SmartInf* (for $k = 50$) have a very high volume of newly exposed population (> 0.4) compared to the baselines. This shows that *SmartInf* efficiently detects those influential nodes (following Sec. IV-B) from retweet sequence T^C, who are responsible for exposing the cascade to a new diffusion locality.

(b) Exposing diverse population: Next, we show that in *SmartInf*, refinement of the ranked influencer list $\mathcal{T}$ to obtain $\mathscr{S}$ indeed enables spreading tweet to a diverse population. In order to demonstrate that, we plot the average pairwise overlap among the top-k influencers obtained from *SmartInf-Temp* ($\mathcal{T}$) and *SmartInf* ($\mathscr{S}$) respectively, for $2 \leq k \leq \beta$ as shown in Fig. 6(c) for *Algeria* dataset (Fig. 6(d) shows the same for *Egypt* dataset). In both the datasets, we observe that this mutual overlap is much higher in case of *SmartInf-Temp*, compared to *SmartInf*, which enables the refined influencer list to reach a diverse population.

Further, we introduce the refining factor ρ ($0 \leq \rho \leq 1$) which indicates the extent to which the ranked list $\mathcal{T}$ is refined. Precisely, we apply Algorithm 2 on *SmartInf-Temp* only upto a certain fraction (ρ) of nodes in $\mathcal{T}$, following the ranking order. We plot the average pairwise overlap among top-k influential nodes for different values of ρ in Figs. 6(d) and 5(b) for *Algeria* and *Egypt* datasets respectively; we observe that the average pairwise overlap decreases with ρ, which is consistent over different values of k across both datasets.

(c) Properties of the influencers: Here we investigate several key features of the detected top-k influential nodes $\mathscr{S}^k$ using *SmartInf* compared to baselines as shown in Fig. 8 (for $k = 50$). We observe that *SmartInf* identifies influential nodes with higher retweet rate compared to the baselines *MCDWE* and *K-truss*, however lower than *MCDWE-Activity* which explicitly takes retweet frequency into consideration. In a similar vein, we observe that influential nodes of *SmartInf* having decent follower count, higher than *MCDWE-Activity* (though lower than *MCDWE* and *K-truss* which primarily relies on identifying high degree nodes belonging to core of the network). Nevertheless, *SmartInf* influencers expose the post to a new diffusion locality, computed as the newly exposed population after the retweet, as compared to baselines. Last, we observe that the mean reaction time is also lowest for the influencers of *SmartInf*. Notably, the influential nodes $\mathscr{S}$ identified by *SmartInf* achieves the aforesaid properties, without explicitly considering the topological structure as well as retweeting behavior of users.

(d) Efficiency of the top influencers: Finally, we measure the efficiency of *SmartInf* in terms of the (i) retweet influence, which indicates the average retweet count for all messages posted by the recommended influential nodes [13] and (ii) infected population, obtained from the epidemic simulation (as described in Section V-B3) considering the recommended influential nodes as seeds. We consider three classes of recommended influencers obtained from *SmartInf*– (1) top score (top-10 ranked users), (2) moderate score (median-10 ranked



(a) Distribution of mean exposed population for influential nodes obtained from different methods in *Algeria* dataset

(b) Distribution of mean exposed population for influential nodes obtained from different methods in *Egypt* dataset

(c) Evidence of reduced follower overlap in *Algeria* dataset after refinement among influential nodes from *SmartInf*

(d) Evidence of reduced follower overlap in *Egypt* dataset after refinement among influential nodes from *SmartInf*

Fig. 6: Key insights for superior performance of *SmartInf*

TABLE IX: Total execution time (in seconds)

Dataset	*SmartInf*	*MCDWE*	*K-truss*	*MCDWE-Activity*
Algeria	8.95	802.77	2328.65	803.05
Egypt	198	5732.42	11427.55	7507.32
Nepal	10.7	1036.2	713.19	1036.7

users) and (3) low score (bottom-10 ranked users) influencers respectively. We normalize the values of retweet influence and infected population by that of the top score influencers respectively. In Figure 9, we observe that the top ranked influencers recommended by *SmartInf* obtain high retweet influence and infected population compared to influencers with moderate or low *SmartInf* scores, which depicts the efficiency of *SmartInf* in correctly ranking the truly influential users at the top compared to the other nodes.

E. Scalability

We show that the proposed *SmartInf* algorithm is scalable in terms of execution time compared to state-of-the-art baseline methods. In Table IX, we report the total execution time of *SmartInf* as well as the baselines, across *Algeria*, *Egypt* and *Nepal* datasets. We observe that the running time of *SmartInf* is much lower compared to baselines, which is consistent with the time complexity computed in Sec. IV-D; on average, it achieves the speedup of $70-100$ times over the competing algorithms. Essentially, *SmartInf* does not require to directly measure the influence of each node in the whole network, which can be costly, in order to rank influential nodes.

F. Robustness of SmartInf

Finally, we investigate the robustness of the influential nodes $\mathscr{S}$ obtained from *SmartInf* against the volume of available cascades. In Fig. 5(c), we compute Spearman rank correlation [46] between the top-k influential nodes (for $k=50$), identified by *SmartInf* from the complete dataset and the corresponding set of influential nodes when a certain fraction of Type I cascades are removed from the data[6]. We observe that the rank correlation is high (0.34) even on removal of 50% of Type I cascades in *Algeria* dataset, depicting the robustness of *SmartInf* even in the face of missing data.

VII. Evaluation on synthetic data

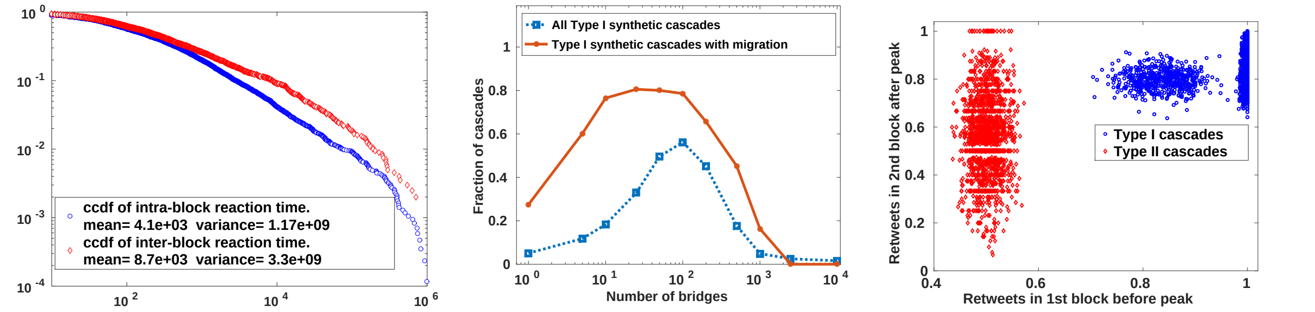
In this section, we delve deep and dissect the performance of the proposed *SmartInf* algorithm on a simple simulation setup. This setup reduces the complexity of the empirical data and allows us to focus only on the relevant dynamics. First, we construct a simple synthetic follower network containing only two diffusion localities and then simulate synthetic cascades on top of it. This synthetic setup helps us to demonstrate the key factors behind the performance improvement of *SmartInf*, as observed from the empirical data.

A. Development of synthetic setup

(a) Synthetic follower network: We construct a synthetic network based on the stochastic block model [47] with two blocks. Such a simple network exhibits a natural partition of nodes through its blocks, which inherently correspond to two different diffusion localities. We construct a directed network composed of two Erdos-Renyi (E-R) [48] sub-networks E_1 and E_2 with parameters (n_1, n_2, p_1, p_2, q) where n_i and p_i denote the size and density of the block E_i, whereas q is the inter-block density. We designate m links connecting the two blocks as *bridges*, and their extremities as *bridge nodes*. We set $m \approx n_1 n_2 q$ where $n_1 n_2 q \ll \min(n_1^2 p_1, n_2^2 p_2)$ [48], ensuring the natural partition of the network into two localities between which diffusion of tweets may occur. In our simulation, we fix the parameter values as $n_1 = 1000$, $n_2 = 1000$, $p_1 = 0.2$, $p_2 = 0.3$ and $q = 5 * 10^{-5}$ such that $m \approx n_1 n_2 q = 50$.

(b) Synthetic cascades: Following Sec. V-B3, we generate the synthetic cascades using standard susceptible-infected (SI) model [45], where initially all the nodes are in susceptible (S) state. We initiate a cascade D from a random node u_0^D and switch its state to infected (I), considering u_0^D tweeted the initial post at time t_0^D. We assume that users exposed to the tweet may get only one chance to retweet with probability γ_u. Hence, the followers of u_0^D who got exposed to the post, get infected (with probability γ_u) by retweeting the post. The retweet time instance t_i^D of user u_i^D can be computed from the reaction time μ_i. Empirical analysis reveals that reaction time depends on whether both u_i^D and u_j^D (user who exposed the cascade D to u_i^D) belong to (a) the same block or (b) different blocks. Accordingly, we assign the reaction time distribution μ_{ident} if both u_i^D and u_j^D belong to the same block and

[6] The principle of *SmartInf* relies on the Type 1 cascades.



(a) CCDF distribution for intra-block and inter-block reaction times from *Algeria* dataset. The latter is significantly broader.

(b) Fraction of Type I synthetic cascades obtained for different number of bridges

(c) Proportion of retweets across different blocks before and after peak reveals migration for synthetic cascades

Fig. 7: Evidence of co-occurrence of flushes and peaks for Type I cascades in synthetic data

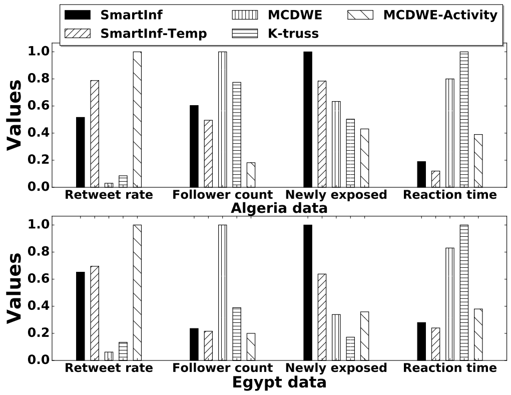


Fig. 8: Evaluation of *SmartInf* based on several key features

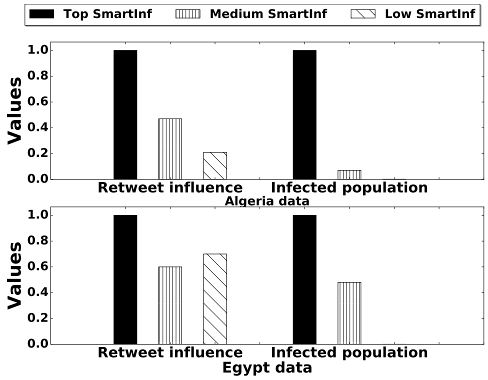


Fig. 9: Comparison of different influential users detected by *SmartInf*

μ_{diff} if u_i^D and u_j^D belong to different blocks respectively. In Fig. 7(a), empirical complementary cumulative distribution functions (ccdf) on the *Algeria* dataset μ_{ident} and μ_{diff} show that the distribution μ_{diff} for inter-block retweets is broader and heavy tailed (with a larger mean), compared to μ_{ident} which favors a faster spread of the tweet inside a diffusion locality (similar to [49]). Hence, while forming the synthetic cascade D, we generate the reaction time μ_i of a retweeting user u_i^D from the suitable reaction time distributions (μ_{ident} and μ_{diff} for intra-block and inter-block respectively), and subsequently compute the retweet time instance t_i^D. Finally for cascade D, we obtain the sequence of inter-retweet time intervals denoted by $T^D = (T_0^D, T_1^D, \ldots, T_{n_D-1}^D)$. We summarize the details of the synthetic data in Table X.

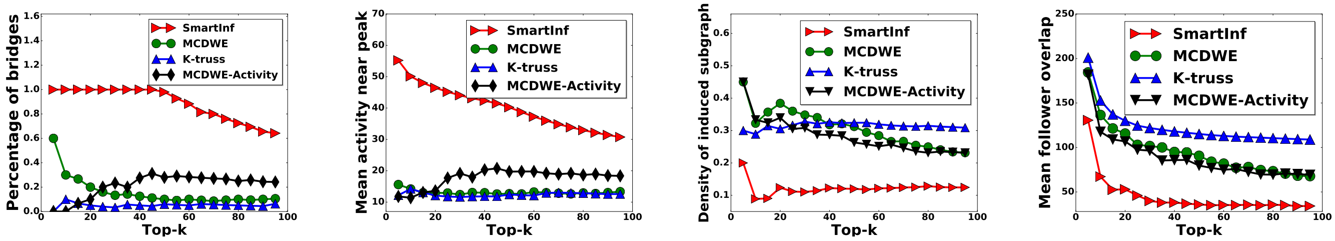
TABLE X: Details of synthetic data

#Tweets	#Retweets	#Cascades	#Users	Maximum cascade size	Mean cascade size
10000	3.2×10^6	10000	2000	476	327.51

(c) Constructing synthetic Type I & Type II cascades: In Fig. 7(b), we decide on the number of bridging links m for which maximum number of Type I cascades emerge and also verify whether the temporal peaks in Type I cascades co-occur with the flush of tweets to a new block, similar to Fig. 3(a). Precisely, we claim that a flush occurs if at least 95% of the retweeting users before the peak belong to first block, whereas atleast 80% of the retweeting users after the peak belong to second block. Notably, a small m restricts the tweet from diffusing to the second block; on the other hand, a large m quickly allows the tweet to diffuse to the second block, without even saturating the first block (as observed in Type II cascades). We observe that keeping the correct number of bridging links m (say, $m = 50$) allows diffusion of tweets from one block to another, while generating enough number of Type I cascades which initially saturate the first block before such a migration. As an evidence, Fig. 7(b) confirms that Type I synthetic cascades indeed reflect the co-occurrence of first peak and flush for 80% of such cascades for $m = 50$, similar to what we revealed in Sec. IV on empirical data. In this line, Fig. 7(c) shows the discrimination observed between the Type I and Type II synthetic cascades; in case of Type I, a large volume of retweets before the peak occur in the first block, whereas the same occurs in the second block after the peak.

B. Experimental observations

We rely on the synthetic dataset to investigate the factors behind the superior performance of the proposed *SmartInf* algorithm. For *SmartInf*, we observe that a majority of bridge nodes emerge in the set of top-k influential nodes $\mathscr{S}^k$ (see Fig. 10(a)), without explicit utilisation of the structural information. The bridge nodes play an important role for diffusing a cascade across multiple localities. On the other hand, the percentage of bridge nodes are low among the top-k influential nodes detected by the baselines; the reason is



(a) Percentage of bridge nodes in top-k influential users (b) Mean activity near peak for top-k influential users (c) Density of subgraph induced by top-k influential users (d) Mean overlap of followers across top-k influential users

Fig. 10: Properties of influential nodes identified by *SmartInf* compared to baselines on synthetic network

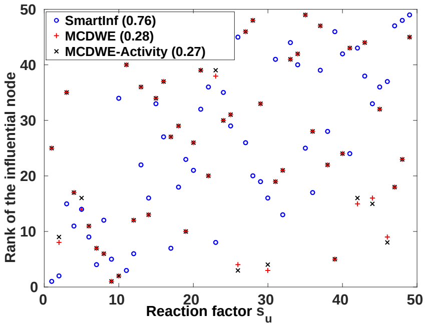


Fig. 11: Biased scaling highly correlates to ranking of influential nodes identified using *SmartInf* (correlation given in brackets) in synthetic network that naturally favors faster propagating bridge nodes

that influential nodes identified by baselines (*MCDWE* and *K-truss*) belong to the core of the network (occupying central positions within a diffusion locality of the network) and hence are effective in diffusing a cascade only within its own locality. Moreover, Fig. 10(b) shows that influential nodes identified by *SmartInf* exhibit high frequency of retweets near peaks of Type I synthetic cascades compared to baselines, which shows their utility in terms of quickly diffusing the tweet in the second block (locality). Additionally, apart from the retweet rate, reaction time (μ) also plays an important role in faster diffusion of tweets to a new locality. We concentrate on the bridging nodes which appears in the set of top-k influential nodes. During simulation setup, we assign a reaction time μ_u to the bridging node u from the (inter-block reaction time) distribution μ_{diff} with mean $\langle\mu_{diff}\rangle$. Now we define the reaction factor s_u for node u as $s_u = \frac{\mu_u}{\langle\mu_{diff}\rangle}$. The smaller reaction factor s_u of u indicates the faster migration of the tweet to the second block. Detecting and ranking such faster propagators at the top of the list may characterize the high quality of influential nodes. In Fig. 11, we show that for the proposed *SmartInf* algorithm, the ranking of the influential nodes in $\mathscr{S}$ is highly correlated (correlation coefficient is 0.76) with their reaction factor s_u, compared to baseline algorithms. Hence, *SmartInf* is able to detect this key feature enabling faster spread, where the baselines fundamentally fail to do so.

Finally, we show that *SmartInf* recommends a diverse set of influential nodes compared to baselines. We compute mean follower overlap over all user pairs in the detected set of influential nodes. Further, we consider the induced subgraph formed by the set of influential nodes and compute link density of this subgraph; high density indicates low diversity across the set of influential nodes. We observe that both link density (see Fig. 10(c)) and mean follower overlap (see Fig. 10(d)) are low for *SmartInf* compared to all baselines. Hence, this high diversity in the set of top-k influential nodes $\mathscr{S}^k$ obtained using *SmartInf* indicates that such users are capable of directly exposing the tweet to a larger part of the network.

VIII. Conclusion and Future Work

This paper addresses the important problem of identifying a set of influential users in a social network by leveraging on temporal sequence of retweets in Twitter cascades. To this end, we present *SmartInf*, an unsupervised algorithm leveraging on *anchor nodes* whose retweets can expose the cascade to a large new population. The ranked list of influential nodes obtained from temporal retweet sequences is then refined to ensure that a diverse population can be reached through influential nodes identified by *SmartInf*. We have demonstrated through extensive experiments on multiple empirical datasets from Twitter that *SmartInf* achieves significant performance boost over recent state-of-the-art baselines both in terms of proposed influence metrics as well as volume of infected population using epidemic simulation. Further, we have shown that influential users detected by *SmartInf* are responsible for faster tweet diffusion to a new locality and also facilitate spreading of tweets to a diverse population in addition to having high retweet rate. We have demonstrated that *SmartInf* is robust to missing data and scales linearly with size of the network achieving $30 - 60$ times speedup in execution time compared to baselines. Finally, we have investigated the factors behind the superior performance of *SmartInf* on a simple synthetic network through simulation of synthetic cascades on this network. Our experiments have revealed that *SmartInf* can detect such influential nodes which play bridging roles in the network connecting multiple localities as well as enable faster spread of information. All these results point to the fact that our proposed *SmartInf* algorithm provides a set of influential users that can quickly spread the message to a large and diverse population in the network.

Our present work has multiple future research directions. First, one may be curious to investigate whether *SmartInf* recommended influential nodes vary significantly across the different topics (such as politics, sports, entertainment etc.)

in Twitter posts. Second, it may be elegant to automatically learn the activation window (δ^C) to identify the potential influential nodes, leveraging on the features extracted from inter-retweet intervals. Third, it is important to conduct numerical simulations on synthetic cascades with rich & realistic structural properties, which may provide deeper insights on the efficiency of *SmartInf*. Fourth, in the current endeavour, we have completely overlooked the role of Type II cascades in influential node detection. It would be an interesting research direction to exploit the Type II cascades in order to refine the list of influencers obtained from *SmartInf*.

Acknowledgements

This work has been partially supported by the DST-BELSPO funded Indo-Belgian collaborative project titled ”DYCIN - Analyzing the Dynamics of Critical Information Diffusion on Social Media: A Network Science initiative”; Concerted Research Action (ARC) supported by the Federation Wallonia-Brussels Contract ARC 14/19-060 ; and Flagship European Research Area Network (FLAG-ERA) Joint Transnational Call FuturICT 2.0 .

References

[1] E. Tonkin, H. D. Pfeiffer, and G. Tourte, “Twitter, information sharing and the london riots?” *ASIST*, vol. 38, no. 2, pp. 49–57, 2012.

[2] Q. Gao, F. Abel, G.-J. Houben, and Y. Yu, “A comparative study of users microblogging behavior on sina weibo and twitter.” UMAP, 2012, pp. 88–101.

[3] G. Wolfsfeld, E. Segev, and T. Sheafer, “Social media and the arab spring: Politics comes first,” *The International Journal of Press/Politics*, vol. 18, no. 2, pp. 115–137, 2013.

[4] A. Bovet, F. Morone, and H. A. Makse, “Validation of twitter opinion trends with national polling aggregates: Hillary clinton vs donald trump,” *Scientific Reports*, vol. 8, no. 1, p. 8673, 2018.

[5] E. Stai, E. Milaiou, V. Karyotis, and S. Papavassiliou, “Temporal dynamics of information diffusion in twitter: Modeling and experimentation,” *TCSS*, vol. 5, no. 1, pp. 256–264, 2018.

[6] R. Dong, L. Li, Q. Zhang, and G. Cai, “Information diffusion on social media during natural disasters,” *TCSS*, vol. 5, no. 1, pp. 265–276, 2018.

[7] D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network.” KDD, 2003, pp. 137–146.

[8] S. Huang, T. Lv, X. Zhang, Y. Yang, W. Zheng, and C. Wen, “Identifying node role in social network based on multiple indicators,” *PloS one, vol. 9, no. 8, p. e103733, 2014*.

[9] C. Xia, S. Guha, and S. Muthukrishnan, “Targeting algorithms for online social advertising markets.” ASONAM, 2016, pp. 485–492.

[10] A. Madotto and J. Liu, “Super-spreader identification using meta-centrality,” *Scientific reports*, vol. 6, p. 38994, 2016.

[11] D. Goldenberg, A. Sela, and E. Shmueli, “Timing matters: Influence maximization in social networks through scheduled seeding,” *TCSS*, vol. 99, pp. 1–18, 2018.

[12] J. Zhang, J. Tang, H. Zhuang, C. W.-K. Leung, and J. Li, “Role-aware conformity modeling and analysis in social networks,” AAAI, 2014, pp. 958–964.

[13] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, “Measuring user influence in twitter: The million follower fallacy,” ICWSM, 2010, pp. 10–17.

[14] D. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, and T. Zhou, “Identifying influential nodes in complex networks,” *Physica A*, vol. 391, no. 4, pp. 1777–1787, 2012.

[15] M. A. Al-garadi, K. D. Varathan, and S. D. Ravana, “Identification of influential spreaders in online social networks using interaction weighted k-core decomposition method,” *Physica A*, vol. 468, pp. 278–288, 2017.

[16] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir, “A model of internet topology using k-shell decomposition,” *PNAS*, vol. 104, no. 27, pp. 11 150–11 154, 2007.

[17] F. D. Malliaros, M.-E. G. Rossi, and M. Vazirgiannis, “Locating influential nodes in complex networks,” *Scientific reports*, vol. 6, p. 19307, 2016.

[18] A. Sheikhahmadi and M. A. Nematbakhsh, “Identification of multi-spreader users in social networks for viral marketing,” *Journal of Information Science*, vol. 43, no. 3, pp. 412–423, 2017.

[19] R. Cappelletti and N. Sastry, “Iarank: Ranking users on twitter in near real-time, based on their information amplification potential,” Social Informatics, 2012, pp. 70–77.

[20] Y. Zhu, D. Li, R. Yan, W. Wu, and Y. Bi, “Maximizing the influence and profit in social networks,” *TCSS*, vol. 4, no. 3, pp. 54–64, 2017.

[21] T. Lou and J. Tang, “Mining structural hole spanners through information diffusion in social networks.” WWW, 2013, pp. 825–836.

[22] L. He, C.-T. Lu, J. Ma, J. Cao, L. Shen, and P. S. Yu, “Joint community and structural hole spanner detection via harmonic modularity.” KDD, 2016, pp. 875–884.

[23] D.-W. Huang and Z.-G. Yu, “Dynamic-sensitive centrality of nodes in temporal networks,” *Scientific reports*, vol. 7, p. 41454, 2017.

[24] F. Erlandsson, P. Bródka, A. Borg, and H. Johnson, “Finding influential users in social media using association rule learning,” *Entropy*, vol. 18, no. 5, p. 164, 2016.

[25] Z. Jianqiang, G. Xiaolin, and T. Feng, “A new method of identifying influential users in the micro-blog networks,” *IEEE Access*, vol. 5, pp. 3008–3015, 2017.

[26] A. Sela, I. Ben-Gal, A. S. Pentland, and E. Shmueli, “Improving information spread through a scheduled seeding approach.” ASONAM, 2015, pp. 629–632.

[27] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, “Can cascades be predicted?” WWW, 2014, pp. 925–936.

[28] I. Taxidou and P. M. Fischer, “Online analysis of information diffusion in twitter,” WWW, 2014, pp. 1313–1318.

[29] S. Pramanik, Q. Wang, M. Danisch, S. Bandi, A. Kumar, J.-L. Guillaume, and B. Mitra, “On the role of mentions on tweet virality.” DSAA, 2016, pp. 204–213.

[30] J. Cheng, L. A. Adamic, J. M. Kleinberg, and J. Leskovec, “Do cascades recur?” WWW, 2016, pp. 671–681.

[31] L. Weng, F. Menczer, and Y.-Y. Ahn, “Predicting successful memes using network and community structure,” ICWSM, 2014, pp. 535–544.

[32] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li, “Social influence locality for modeling retweeting behaviors,” IJCAI, 2013, pp. 2761–2767.

[33] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, “Seismic: A self-exciting point process model for predicting tweet popularity,” KDD, 2015, pp. 1513–1522.

[34] R. Kobayashi and R. Lambiotte, “Tideh: Time-dependent hawkes process for predicting retweet dynamics.” ICWSM, 2016, pp. 191–200.

[35] G. Tavares and A. Faisal, “Scaling-laws of human broadcast communication enable distinction between human, corporate and robot twitter users,” *PLoS one*, vol. 8, no. 7, p. e65774, 2013.

[36] R. Ghosh, T. Surachawala, and K. Lerman, “Entropy-based classification of’retweeting’activity on twitter,” *arXiv preprint arXiv:1106.0346*, 2011.

[37] A. K. Bhowmick, M. Gueuning, J.-C. Delvenne, R. Lambiotte, and B. Mitra, “Temporal pattern of (re) tweets reveal cascade migration.” ASONAM, 2017, pp. 483–488.

[38] A. Bruns, T. Highfield, and J. Burgess, “The arab spring and social media audiences,” *ABS*, vol. 57, no. 7, pp. 871–898, 2013.

[39] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[40] Q. Yang, A. Nofsinger, J. McPeek, J. Phinney, and R. Knuesel, “A complete solution to the set covering problem,” CSC, 2015, pp. 36–41.

[41] Y. Ding, E. Yan, A. Frazho, and J. Caverlee, “Pagerank for ranking authors in co-citation networks,” *ASIST*, vol. 60, no. 11, pp. 2229–2243, 2009.

[42] B. Ruhnau, “Eigenvector-centralitya node-centrality?” *Social networks*, vol. 22, no. 4, pp. 357–365, 2000.

[43] F. Riquelme and P. González-Cantergiani, “Measuring user influence on twitter: A survey,” *IPM*, vol. 52, no. 5, pp. 949–975, 2016.

[44] Z.-y. Ding, Y. Jia, B. Zhou, Y. Han, L. He, and J.-f. Zhang, “Measuring the spreadability of users in microblogs,” *Journal of Zhejiang University SCIENCE C*, vol. 14, no. 9, pp. 701–710, 2013.

[45] M. J. Keeling and K. T. Eames, “Networks and epidemic models,” *The Royal Society*, vol. 2, no. 4, pp. 295–307, 2005.

[46] T. D. Gauthier, “Detecting trends using spearman’s rank correlation coefficient,” *Environmental forensics*, vol. 2, no. 4, pp. 359–362, 2001.

[47] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[48] E. Abbe, “Community detection and stochastic block models: recent developments,” *JMLR*, vol. 18, no. 1, pp. 6446–6531, 2017.

[49] O. Artime, J. J. Ramasco, and M. San Miguel, “Dynamics on networks: competition of temporal and topological correlations,” *Scientific reports*, vol. 7, p. 41627, 2017.