IMINIMIZE: A System for Negative Influence Minimization via Vertex Blocking

Siyi Teng
East China Normal University
tsyxxxka@gmail.com

Jiadong Xie
University of New South Wales
xiejiadong0623@gmail.com

Mingkai Zhang
Guangzhou University
mingkai_zhang@e.gzhu.edu.cn

Kai Wang
Antai College of Economics and Management, Shanghai Jiao Tong University
w.kai@sjtu.edu.cn

Fan Zhang
Guangzhou University
zhangf@gzhu.edu.cn

ABSTRACT

The rapid rise and prevalence of social platforms have created great demands on effective schemes to limit the influence of negative information, e.g., blocking key vertices for influence minimization. However, there is currently no system providing practical schemes to solve the negative influence minimization problem with a blocking budget effectively and efficiently in the literature. In this demo, we present IMINIMIZE, the first interactive system that provides audiences with vertex-blocking schemes over different budgets and demonstrates via visualization for comparison vividly and directly, aiming to help minimize the negative influence spreading in networks. Our IMINIMIZE system applies an advanced greedy algorithm to select blocked vertices with both high efficiency and effectiveness. Furthermore, we extend IMINIMIZE to the application of epidemic controlling and prevention and show the usability of IMINIMIZE through two case studies of real-life applications.

CCS CONCEPTS

• **Information systems** → *Data mining*.

KEYWORDS

Negative information, Influence diffusion, Graph analytics system

ACM Reference Format:

Siyi Teng, Jiadong Xie, Mingkai Zhang, Kai Wang, Fan Zhang. 2023. IMINIMIZE: A System for Negative Influence Minimization via Vertex Blocking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3614743

1 INTRODUCTION

With the prevalence of social network platforms and the global epidemic, modeling influence diffusion [2, 26] and effective strategies for minimizing the negative influence [6, 11, 17, 21, 22, 24] have attracted increasingly more attention than ever before. In this paper, we demonstrate IMINIMIZE, a novel graph analytic and mining system to block b vertices for minimizing the negative influence diffusion. Specifically, given a graph G with a seed set S and the budget b, the problem aims to find a vertex set B with at most b non-seed vertices such that the expected influence spread from the seed set S is minimized. Example use cases of IMINIMIZE include:

Online Social Network Influence Minimization. Suppose a group of users (i.e., seed vertices) in an online social network have already spread negative information (e.g., rumors and fake science) and who have seen the information may start the propagation. We have a budget for the blocking cost, i.e., the maximum number of users that can be blocked. Note that IMINIMIZE will choose b non-seed vertices to block such that the spread of misinformation in the social network is minimized since the negative content has already been posted by the seed users. In real scenarios, blocking does not mean prohibiting users from posting all content, but leads to manually/automatically reviewing the content posted by the blocked users to make sure no negative information is posted.

Epidemic Controlling and Prevention. In the important application of epidemic control and prevention, we can detect infectious places through epidemiological investigation, while the infected people may have left and spread the epidemic. In order to reduce the spread of the epidemic as much as possible, we have to discover if there are potential infectious people. Therefore, the most effective way for epidemic control is to "block" other places, e.g., set checkpoints for quick detection in some key intersections to hinder epidemic diffusion. In such a scenario, we model the road network as a directed graph, with vertices representing different locations, and IMINIMIZE can help to choose the best checkpoints by providing the suggested blocked vertices.

Although selecting key blocked vertices is a powerful strategy for the negative influence minimization [6, 11], finding the optimal blocked vertex set is computationally expensive (the problem is NP-Hard and APX-hard unless P=NP [21]). Thus, previous works propose several heuristic methods to choose blocked vertices, e.g., degree [6, 11], betweenness [24]. Among these methods, the greedy algorithm has shown the best performance [17, 22]. However, since the computation of influence spread is #P-hard [3], the state-of-the-art greedy solution relies on Monte-Carlo simulations to estimate

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom.

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614743

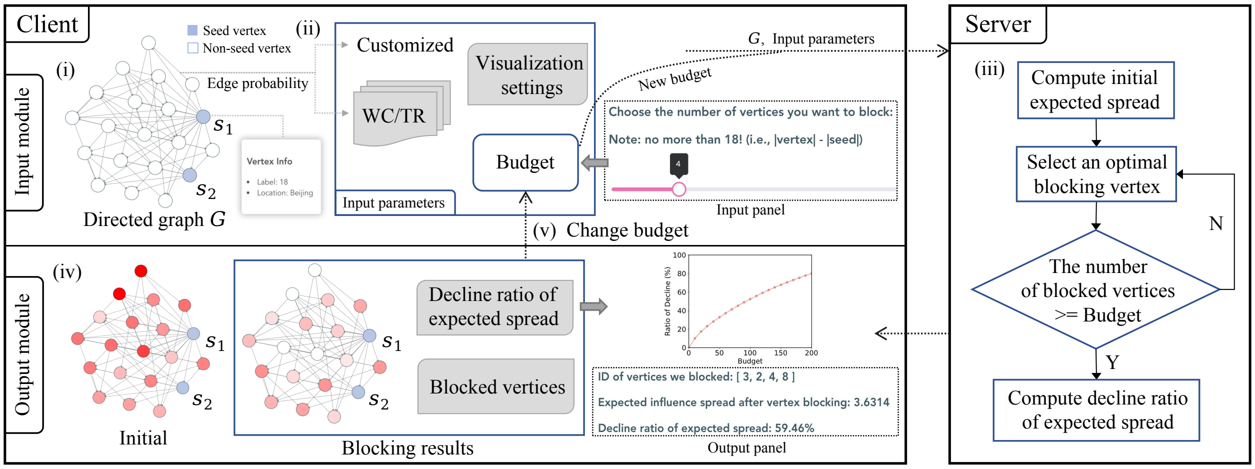


Figure 1: System Overview

the influence spread, resulting in high computational costs. To address this challenge, we propose an advanced greedy algorithm with a novel dominator tree-based estimation technique, which can achieve up to 6 orders of magnitude higher efficiency than the state-of-the-art greedy algorithm without sacrificing effectiveness. While the technical details are detailed in [21], we aim to introduce the system IMinimize, which applies the advanced greedy algorithm, to CIKM audiences through this demo.

The demo has three parts. In the first part, we will introduce the system architecture of IMinimize (Section 2), and then we will present the related works and systems to show the novelty of IMinimize (Section 3). In the final part, we demonstrate IMinimize in two real-life datasets to show its useful features and IMinimize also offers the flexibility for audiences to upload their own datasets for customized analysis and exploration (Section 4).

2 SYSTEM ARCHITECTURE

The system architecture of IMinimize is illustrated in Figure 1. It employs a client-server architecture and consists of three components: input module, server process module, and output module.

The input module specifies the graph data and parameter settings, and it also allows audiences to change the budget of blocked vertices. The server process module is the core phase of IMinimize and contains two parts: expected influence spread estimation and blocked vertex selection. The output module displays the blocking scheme and visualizes the final influence diffusion results after blocking key vertices.

2.1 Input Module

The input module of IMinimize consists of two parts: graph data uploading and parameters settings, which are shown in the upper-left part of Figure 1.

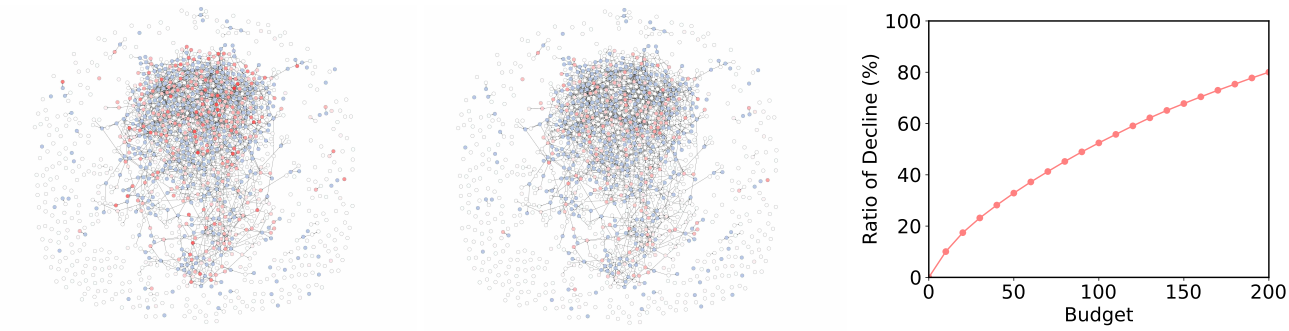
For graph data uploading, IMinimize allows audiences to upload their own directed graph data files in a specific format. The graph data can contain customized activation probability of each edge, e.g., learning from users' historical action data [26]. If unavailable, users can choose two simple heuristic influence probability assignments, i.e., (i) Trivalency (TR) model, which assigns $p_{u,v} = TRI$ for each edge (u, v), where TRI is uniformly selecting a value from $\{0.1, 0.01, 0.001\}$ [3, 22]; and (ii) Weighted cascade (WC) model, which assigns $p_{u,v} = 1/d_v^{in}$, where d_v^{in} denotes the in-degree of vertex v [4, 14]. In addition, the graph data can also contain supplementary information, e.g., hyperlinks to users' homepages in online social networks. Next, the budget (i.e., the number of blocked vertices) and some visualization settings (e.g., whether to show the vertex id or the edge probability) can be set via the interface, and the results will be shown in the visualization panel.

2.2 Server Process Module

The server process module is the core phase in IMinimize (the right part of Figure 1). It mainly contains two parts: expected spread computation and greedily optimal selection of blocked vertices.

Expected Influence Spread Estimation. As computing the expected influence spread of a seed set is #P-hard [3], and the exact computation solution can only be used in small graphs (e.g., with a few hundred edges) [7], IMinimize uses Monte-Carlo Simulations (MCS) to estimate the expected influence spread [4]. In each round of MCS, it removes every edge (u, v) with $(1 - p_{u,v})$ probability. Let G' be the resulting graph after the removing, and the set $R(S)$ contains the vertices in G' that are reachable from at least one seed from seed set S (i.e., there exists at least one path from seed $s \in S$ to each vertex in $R(S)$). For the original graph G and seed set S, the expected size of set $R(S)$ equals the expected influence spread under the independent cascade model [4]. In IMinimize, we set the number of MCS rounds as 10^5 for estimating.

Blocked Vertex Selection. The blocked vertex selection is the core part of the server process. To provide a fast selection with high effectiveness, in this module, IMinimize applies an advanced greedy algorithm [21]. For influence minimization via blocking vertex, the



(a) Initial Expected Spreading **(b) After Blocking 100 Vertives** **(c) Ratio of Decline v.s. Budget**

Figure 2: Minimizing "MelCup" Diffusion in Twitter

greedy algorithm, which is the state-of-the-art over all previous proposed heuristics algorithms [17, 22]. A typical process is as follows: we start with an empty set $B = \emptyset$, and then iteratively add vertex u into set B that leads to the largest decrease of the expected influence spread, until $|B| = b$. Although effective, the greedy algorithm needs a large computation cost because we have to enumerate every candidate vertex in each round and use Monte Carlo Simulations to compute the decrease of expected spread from removing the vertex. To deal with this, we propose a novel estimation algorithm based on graph samples and dominator trees [21], i.e., (i) we estimate the expected influence spread through the average influence spread on each graph sample (the law of large numbers [23]); and (ii) on each graph sample, we apply the dominator tree to compute the decrease of the expected influence spread led by each candidate blocker at once (Theorem 6 in [21]). As IMinimize applies the greedy algorithm with this new framework of expected spread estimation for selecting the candidates, it achieves higher efficiency without sacrificing effectiveness, compared with previous greedy algorithms [17, 22].

We will refer audiences to [21] for more details of the advanced greedy algorithm with the novel estimation technique. Our empirical study in [21] on 8 real-life datasets and theoretical analysis validates that the algorithm in IMinimize is faster than previous state-of-the-art by more than 3 orders without loss of effectiveness.

2.3 Output Module

The output module displays a visualization panel for analysis and comparison of the blocking results in a user-friendly manner. The suggested blocked vertices (i.e., the list of ids of the blocked vertices) and the decline ratio of expected influence spread after blocking will be shown in the panel. The visualization of the graph uses the tool of AntV-G6[1]: a directed graph is drawn according to the uploaded graph data file, with the labels on vertices or edges, e.g., vertex supplementary information, edge activation probabilities. Moreover, the visualized graph allows audiences to adjust its layout. The colors of vertices represent their states. Specifically, the seed vertices are colored blue, and non-seed vertices are colored red, i.e., a non-seed vertex in a darker red color means it has a larger probability of being activated. Thus audiences can visually see the effect of the blocking scheme by comparing the degree of redness before and after blocking. Audiences can also evaluate the effect of different blocking budgets for blocking through the visualized graph and the decline ratio.

[1]https://g6.antv.antgroup.com/en

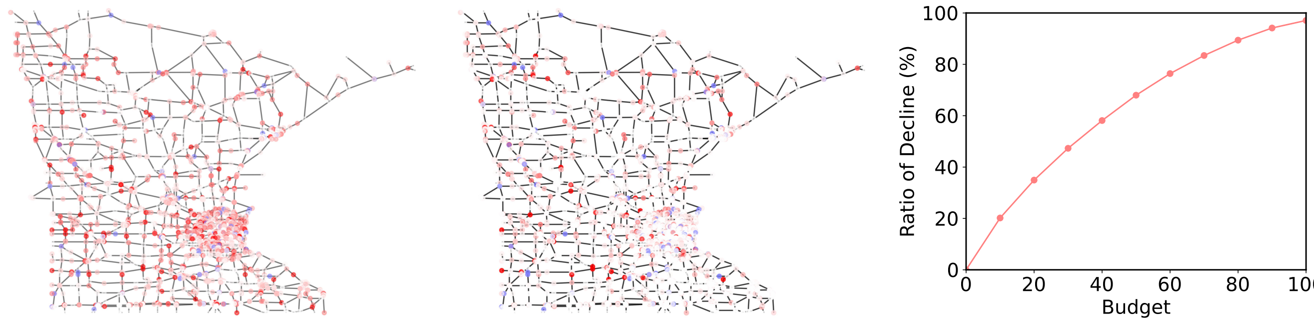
3 RELATED WORKS AND NOVELTY

Influence Minimization. As surveyed in [25], most works for negative influence minimization consider proactive measures (e.g., blocking vertices) to minimize the influence spread, motivated by the feasibility of structure change for influence study [9, 13, 16, 20]. In real networks, we may use various methods to find the key vertices, e.g., degree [6, 11], betweenness and out-degree [24] and greedy heuristics [17, 22]. In addition, there are some other strategies to limit the influence spread, e.g., blocking edges (i.e., finding k edges to remove) [5], adding seeds for the opposing campaign (e.g., truth is the opposing campaign of rumor) [1, 8, 15]. Moreover, some works consider more factors into the propagation models, e.g., user experience [16], evolution of user opinions [12]. In our IMinimize system, we focus on an efficient selection of vertices as the blockers to minimize the influence spread of negative information in the networks (e.g., social networks, road networks).

Graph Analytics and Mining Systems. There are various graph analytic and mining systems developed in different scenarios for real applications [18, 19, 27]. Seastar [18] demonstrates a novel GNN training framework to simplify model development and improve training efficiency. DPGraph [19] is a web-based end-to-end platform for evaluating private algorithms on graph data. HDAG-Explorer [27] presents an interactive system for hierarchical DAG summarization. Different from the above systems, our system IMinimize focuses on minimizing the negative influence diffusion via vertex blocking for different types of networks.

4 DEMONSTRATION OVERVIEW

In this section, we introduce the demonstration of IMinimize. IMinimize enables audiences to interact with our system and enjoy data exploration. Specifically, we demonstrate the system in two real-world applications: influence diffusion minimization on social networks (Section 4.1) and infectious diseases control and prevention on road networks (Section 4.2).



(a) Initial Expected Spreading **(b) After Blocking 50 Vertices** **(c) Ratio of Decline v.s. Budget**

Figure 3: Virtual Epidemic Controlling in Minnesota Road Network

4.1 Minimizing The Event Diffusion on Twitter

We present an application for minimizing the specific event diffusion on social networks. We use the event-based `Melcup17` dataset[2] collected from Twitter [26]. The dataset collects 78300 status messages posted by 34573 users in 3 days through tracking the terms which are related to "MelCup". In this case study, the vertices represent the users on Twitter, and the edges represent the influence propagation. The existence of the edge from u to v means that user v is influenced by user u, and the probability of this edge is computed based on the interaction history between u and v, e.g., following relations between them. We compute the activation probability for each edge on the graph data following [26], which contains 1485 vertices and 70756 edges. We then select 412 vertices as the seeds, because they posted the specific content initially, i.e., never been influenced by others who have posted the same topic before. We show the initial expected spreading situation in Figure 2a (we filter the edges whose probability is less than 0.1 for clear demonstration), with seed vertices in blue color and non-seed vertices in red color. Figure 2b shows that after blocking 100 vertices, the infected probability of vertices has declined significantly (around 50%), compared with Figure 2a. The decline ratios of infected probabilities under different budgets are shown in Figure 2c, which shows blocking suggested vertices from IMinimize can effectively hinder the influence diffusion on Twitter.

4.2 Virtual Epidemic Controlling in Minnesota

We use the data of the Minnesota road network[3] for simulating epidemic control virtually. The dataset contains 2642 vertices (i.e., intersection) and 3303 edges (i.e., roads). We randomly select 100 vertices as the seeds (in real applications, the intersections where infected people have arrived can be detected by epidemiological investigation [10]). As the road network is undirected, we regard them as two directed edges in the graph. For infectious intersections (i.e., seeds), the infected people may choose one road to leave. We assume the probability on each edge (u, v) is $1/d_u^{out}$, where d_u^{out} is the out-degree of vertex u. We show the initial expected spreading situation in Figure 3a, the seeds are colored blue and the depth of color red represents the probability of a vertex being infected.

[2] https://bit.ly/2UlQ3xW

[3] http://www.cise.ufl.edu/research/sparse/matrices/Gleich/minnesota

Figure 3b shows the situation when IMinimize chooses 50 vertices to block. It is clear that the infected probability of vertices in the graph is much smaller (the expected infected places decreased by around 67.99%). We also show the ratio of decline under the different budgets (i.e., numbers of blocked vertices) in Figure 3c. We can find that IMinimize achieves a larger ratio of spread decline with a larger budget on the number of blocked vertices.

4.3 Interactive on Customized Settings

Audiences can upload customized graph datasets through the input module or replace the blocker selection strategy by simple modification of the server process module. For interactive customized graphs, audiences can change the budget by dragging the scroll bar in the visualization panel, then IMinimize will visualize the results and show the decline ratio. Audiences can also interact with the visualized graph by changing its format or clicking the vertices to gain more information. The implementation of blocked vertex selection in IMinimize is separated, thus audiences can easily replace the blocker selection strategy with customized methods to explore more effective ways for blocking.

5 SUMMARY

We plan to introduce IMinimize, an efficient interactive system for negative influence minimization via vertex blocking, to CIKM audiences. In the demonstration, audiences will experience IMinimize by customizing input graphs and parameters to simulate the influence propagation in real-world application scenarios. Audiences can gain a clear and intuitive understanding of the high efficiency and effectiveness of IMinimize through the visualization panel. IMinimize is an open-source project. It is available at https://github.com/Tsyxxxka/IMinimize.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. This work is partially supported by the National Natural Science Foundation of China (62002073) and the Guangdong Basic and Applied Basic Research Foundation (2023A1515012603). Jiadong Xie is the corresponding author.

REFERENCES

[1] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. 2011. Limiting the spread of misinformation in social networks. In *WWW*. 665–674.

[2] Damon Centola. 2010. The Spread of Behavior in an Online Social Network Experiment. *Science* 329, 5996 (2010), 1194–1197.

[3] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*. 1029–1038.

[4] David Kempe, Jon M. Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *KDD*. 137–146.

[5] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. 2008. Minimizing the Spread of Contamination by Blocking Links in a Network. In *AAAI*.

[6] Newman M, Forrest Stephanie, and Balthrop Justin. 2002. Email Networks and the Spread of Computer Viruses. *Physical Review E* (2002), 035101.

[7] Takanori Maehara, Hirofumi Suzuki, and Masakazu Ishihata. 2017. Exact Computation of Influence Spread by Binary Decision Diagrams. In *WWW*. 947–956.

[8] Mohammad Ali Manouchehri, Mohammad Sadegh Helfroush, and Habibollah Danyali. 2021. A Theoretically Guaranteed Approach to Efficiently Block the Influence of Misinformation in Social Networks. *IEEE Trans. Comput. Soc. Syst.* (2021), 716–727.

[9] Liqiang Nie, Xuemeng Song, and Tat-Seng Chua. 2016. *Learning from Multiple Social Networks*. Vol. 8. Synthesis Lectures on Information Concepts Retrieval and Services. 118 pages.

[10] Rosanna W Peeling, Piero L Olliaro, Debrah I Boeras, and Noah Fongwen. 2021. Scaling up COVID-19 rapid antigen tests: promises and challenges. *The Lancet infectious diseases* 21, 9 (2021), e290–e295.

[11] Albert Rita, Jeong Hawoong, and Barabasi Albert-Laszlo. 2000. Error and Attack Tolerance of Complex Networks. *Nature* 406 (2000), 378–382.

[12] Akrati Saxena, Wynne Hsu, Mong-Li Lee, Hai Leong Chieu, Lynette Ng, and Loo-Nin Teow. 2020. Mitigating Misinformation in Online Social Network with Top-k Debunkers and Evolving User Opinions. In *Companion of The Web Conference*. ACM / IW3C2, 363–370.

[13] Changfeng Sun, Han Liu, Meng Liu, Zhaochun Ren, Tian Gan, and Liqiang Nie. 2020. LARA: Attribute-to-feature Adversarial Learning for New-item Recommendation. In *WSDM*. 582–590.

[14] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD*. 75–86.

[15] Guangmo Tong, Weili Wu, Ling Guo, Deying Li, Cong Liu, Bin Liu, and Ding-Zhu Du. 2020. An Efficient Randomized Algorithm for Rumor Blocking in Online Social Networks. *IEEE Trans. Netw. Sci. Eng.* 7, 2 (2020), 845–854.

[16] Biao Wang, Ge Chen, Luoyi Fu, Li Song, Xinbing Wang, and Xue Liu. 2016. DRIMUX: Dynamic Rumor Influence Minimization with User Experience in Social Networks. In *AAAI*. 791–797.

[17] Senzhang Wang, Xiaojian Zhao, Yan Chen, Zhoujun Li, Kai Zhang, and Jiali Xia. 2013. Negative Influence Minimizing by Blocking Nodes in Social Networks. In *AAAI Workshops*, Vol. WS-13-17.

[18] Yidi Wu, Yuntao Gui, Tatiana Jin, James Cheng, Xiao Yan, Peiqi Yin, Yufei Cai, Bo Tang, and Fan Yu. 2021. Vertex-Centric Visual Programming for Graph Neural Networks. In *SIGMOD*. ACM, 2803–2807.

[19] Siyuan Xia, Beizhen Chang, Karl Knopf, Yihan He, Yuchao Tao, and Xi He. 2021. DPGraph: A Benchmark Platform for Differentially Private Graph Analysis. In *SIGMOD*. ACM, 2808–2812.

[20] Jiadong Xie. 2022. Hindering Influence Diffusion of Community. In *SIGMOD*. ACM, 2518–2520.

[21] Jiadong Xie, Fan Zhang, Kai Wang, Xuemin Lin, and Wenjie Zhang. 2023. Minimizing the Influence of Misinformation via Vertex Blocking. In *ICDE*. IEEE, 789–801.

[22] Ruidong Yan, Deying Li, Weili Wu, Ding-Zhu Du, and Yongcai Wang. 2020. Minimizing Influence of Rumors by Blockers on Social Networks: Algorithms and Analysis. *IEEE Trans. Netw. Sci. Eng.* 7, 3 (2020), 1067–1078.

[23] Kai Yao and Jinwu Gao. 2016. Law of Large Numbers for Uncertain Random Variables. *IEEE Transactions on Fuzzy Systems* 24, 3 (2016), 615–621.

[24] Qipeng Yao, Ruisheng Shi, Chuan Zhou, Peng Wang, and Li Guo. 2015. Topic-aware Social Influence Minimization. In *WWW*. 139–140.

[25] Ahmad Zareie and Rizos Sakellariou. 2021. Minimizing the spread of misinformation in online social networks: A survey. *J. Netw. Comput. Appl.* (2021).

[26] Zizhu Zhang, Weiliang Zhao, Jian Yang, Cécile Paris, and Surya Nepal. 2019. Learning Influence Probabilities and Modelling Influence Diffusion in Twitter. In *WWW*. ACM, 1087–1094.

[27] Xuliang Zhu, Xin Huang, Jinbin Huang, Byron Choi, and Jianliang Xu. 2020. HDAG-Explorer: A System for Hierarchical DAG Summarization and Exploration. *Proc. VLDB Endow.* 13, 12 (2020), 2973–2976.