applied sciences

MDPI

Article

Finding Influencers Based on Social Interaction and Graph Structure in Social Media

Jongtae Lim [1], Hwanyong Choi [1], Sanghyun Choi [1], Kyoungsoo Bok [2] and Jaesoo Yoo [3,*]

[1] Department of Bigdata, Chungbuk National University, Cheongju 28644, Republic of Korea; jtlim@chungbuk.ac.kr (J.L.); chlghksdyd24@naver.com (H.C.); chois@chungbuk.ac.kr (S.C.)
[2] Department of Artificial Intelligence Convergence, Wonkwang University, Iksan 54538, Republic of Korea; ksbok@wku.ac.kr
[3] School of Information & Communication Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea
* Correspondence: yjs@chungbuk.ac.kr; Tel.: +82-261-3230

Abstract

With the development of online social media, influencer detection methods on these platforms have become an important area of study. However, existing influencer detection methods often place significant emphasis on the number of followers, which can lead to a drawback in maintaining the influence of users who have not been very active recently. In this paper, we propose an influencer detection method that takes both social interactions and the graph structure of social media into account. By considering both social interactions and graph structure, the proposed method prevents influence scores of users who have not been recently active from remaining disproportionately high. To demonstrate the superiority of the proposed method, we conducted a performance comparison with existing methods.

Keywords: social media; social interaction; graph structure; influencer detection; page rank

1. Introduction

Online social networks have emerged with the development of Web 2.0 technologies and the increased use of the Internet. Early social networks were mainly created to connect friends, but now they are used in various areas such as business, news, and entertainment, and several platforms exist, including Facebook, Twitter, Instagram, and LinkedIn. The importance of online social networks is steadily increasing [1]. Using a small number of influencers to spread information to many users is important for effective advertising or disaster information dissemination. For this reason, research on influencer detection methods has been actively conducted [1–6].

With the advancement of the information and communication technology field and the growing influence of online social media, many studies have explored applications that leverage the characteristics of influencers who exert impact on numerous followers to disseminate information to a large audience through a small number of influential individuals [4]. Most existing influencer detection methods place significant emphasis on the number of follower edges in a social graph. However, relationships between users in social media platforms, once established, tend not to be easily dissolved. Consequently, approaches that rely solely on follower counts often suffer from the issue that inactive followers disproportionately contribute to the perceived influence of a user. Therefore,

Academic Editors: Chengliang Chai and Yu Sun

Received: 29 November 2025
Revised: 6 January 2026
Accepted: 9 January 2026
Published: 10 January 2026

 https://doi.org/10.3390/app16020738

in the rapidly evolving social media environment, it is necessary to consider various additional features beyond follower count when identifying influencers.

In this paper, we propose an influencer detection method that considers both social interactions and graph structure in social media. The proposed method uses a dissemination score, which is calculated from the influencer's social interactions, and an infrastructure score, which is calculated from the graph structure. By considering both social interactions and graph structure, the method solves the problem where users with many inactive followers receive an overly high final influencer score. To show the effectiveness of the proposed method, we conduct various performance evaluations.

The structure of this paper is as follows. Following the introduction in Section 1, Section 2 analyzes existing methods for detecting influencers in social media and discusses their limitations. Section 3 describes the proposed influencer detection method. Section 4 presents the results of a comparative evaluation between the proposed method and existing methods, and analyzes the performance evaluation results. Finally, Section 5 concludes the paper and discusses future research directions.

2. Related Work

Recent studies have focused on detecting influencers who can have a significant impact on networks. Existing research can be categorized into non-graph-based methods, graph-based methods, and network models.

2.1. Non-Graph-Based Methods

Non-graph-based methods do not use the structure of the network at all. Instead, they detect influencers using non-graph information such as interactions or content [1]. In [1], features from social media platforms such as Twitter, Facebook, and Instagram are used. These features include the number of shares, results of sentiment analysis, monthly interactions, user activity time, and the number of posts. Using these variables, multiple regression models such as Lasso [7] and k-NN regression [8] are applied to estimate the influencer score of users.

Because non-graph-based methods do not use relational and connection information in social media, they can easily miss important contextual and connection information. They are also less adaptable to structural changes in social media, and, compared to dynamic network models, it is difficult to detect or update new influencers and relationship changes in real time. In addition, they do not consider network topology, which contains important information for information diffusion. Furthermore, when machine learning is used, labeled data are required for training.

2.2. Graph-Based Methods

Graph-based methods calculate user influence by using the structure of the network to detect influencers [4–6]. These methods can be broadly divided into information diffusion models and structural centrality methods. Representative information diffusion models include the Independent Cascade (IC) model, Linear Threshold (LT) model, and Susceptible-Infectious-Recovered (SIR) model [6]. These models were originally designed based on the spread of infectious diseases and simulate the diffusion of information between users in the network through edge-based propagation probabilities.

Structural centrality methods include well-known approaches such as PageRank [9,10] and K-Shell Decomposition [11,12]. PageRank-based methods (hereafter PR) apply the PageRank algorithm to the social network and select the top-k users with the highest PageRank scores [5]. K-Shell-based methods (Evidential Centrality-Related Method, hereafter ECRM) use the number of followers and the K-Shell similarity between a user and their followers [6].

Because social media graphs are extremely large, graph-based methods often require dividing the full network into subgraphs to perform centrality analysis with high computational complexity, such as betweenness centrality. Additionally, on social media, once a relationship is established, it usually remains unless there is a special reason for removal. Therefore, methods that consider only the structural characteristics of the graph tend to favor older users when detecting influencers. These methods also struggle in situations where a user has many connections, but neighboring nodes do not actively interact with them. Therefore, rather than relying on indirect measures such as similarity [6], it is necessary to adopt a method that measures actual social interactions and the real propagation process as scores and utilizes them for influencer detection.

In influence maximization studies, researchers have investigated selecting a specific number of nodes in social media to maximize information diffusion [13–20]. In [13], an approximate algorithm based on submodular function optimization was proposed. In [14,17,19], Graph Neural Networks (GNNs) were used to optimize the influence maximization problem based on data. In particular, [19] improved accuracy by using deep graph representation learning in social networks. Refs. [18,20] investigated the prediction of influence maximization using deep reinforcement learning. However, the methods also have limitations, as they either do not consider social interactions and graph structure or consider them only in a limited way. Moreover, GNN, deep representation learning, and deep reinforcement learning-based approaches aim to improve accuracy, but they need high computational resource requirements and long training times.

2.3. Network Models

Network models can be classified into static network models, snapshot network models, and dynamic network models. Static network models do not reflect changes in the network over time. Snapshot and dynamic network models require continuous collection of network change information, which incurs collection costs, and they depend on the APIs of social media platforms for ongoing data collection, which is a limitation.

3. The Proposed Influencer Detection Method

3.1. Characteristics

Figure 1 shows the overall structure of the proposed influencer detection method. The proposed method is based on the social media platform X (Twitter). The method consists of building the social graph and retweet graph, feature analysis, and influencer score calculation. The proposed method returns the users with the highest influence as the result by considering both the infrastructure score and the dissemination score.

3.2. Feature Analysis

Feature analysis is the process of deriving the attributes needed to calculate influence from the graphs constructed from social media data. Figure 2 shows the calculation process for the attributes in the feature analysis stage. To indirectly measure the influence of each user in the social graph and the retweet graph, user attributes are calculated for all users in the entire network. In this process, the attributes used for calculation consist of social features and retweet features. The social features and retweet features are calculated based on the social graph and retweet graph, respectively.

Social features based on graph structure include the number of followers and the K-Shell iteration value of each user. The number of followers indicates how many people a user can potentially influence. The K-Shell iteration value measures the centrality of a user within the graph. The proposed method assumes that users with higher centrality in the graph are more likely to participate in information propagation. The number of followers and the K-Shell iteration value are denoted as F_i and $F(it_i)$, respectively. The number of followers considers only incoming edges for each user. The K-Shell iteration value is calculated using the K-Shell decomposition method. Here, the K-Shell iteration value refers to the iteration at which the node (user) is removed, rather than the K-Shell value of the node itself. The reason for using the K-Shell iteration value instead of the K-Shell value is that the iteration value provides a simple and effective measure of node centrality [4].

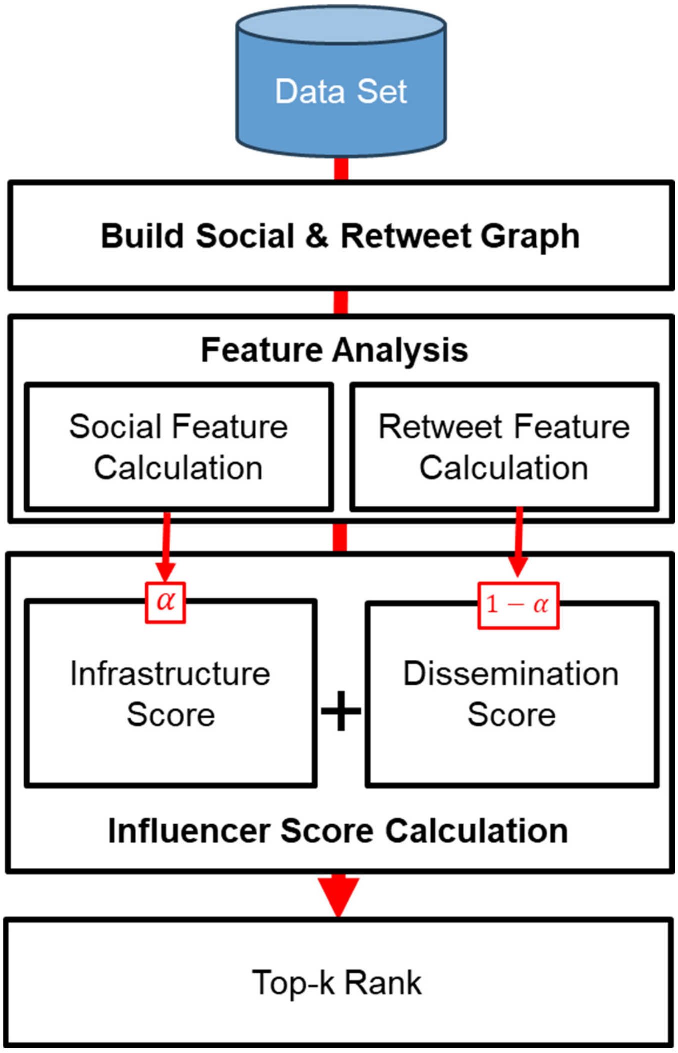

Figure 1. Overall processing procedure of the proposed method.

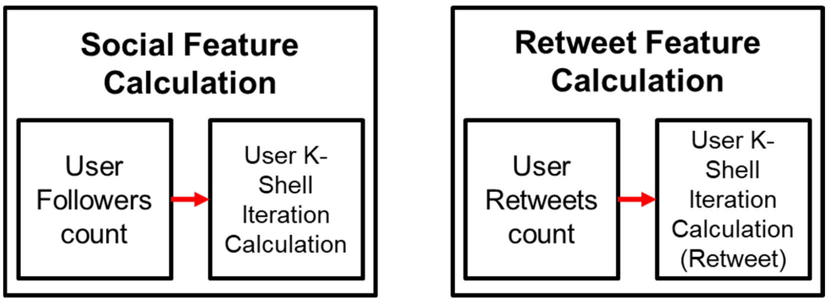

Figure 2. Process of feature calculation.

First, for all users in the entire graph, the incoming edges for the number of followers are calculated. Then, the K-Shell iteration value for each user is computed through a graph decomposition process.

Retweet features based on social interactions include the number of retweets and the Retweet K-Shell iteration value of each user. The number of retweets indicates how much attention and engagement a user receives from others. The retweet K-Shell iteration value measures the centrality of a user in the retweet graph. The number of retweets and the retweet K-Shell iteration value are denoted as R_i and $R(it_{\mathrm{i}})$, respectively. The number of retweets considers only incoming edges for each user. The K-Shell iteration value is calculated for the retweet graph in the same way as it is calculated for the social graph.

3.3. Influencer Score Calculation

Influencer score calculation is the process of computing the influencer score by combining the dissemination score and the infrastructure score derived in the feature analysis stage. The influencer score is obtained by assigning weights to the dissemination score and the infrastructure score and then summing them.

3.3.1. Infrastructure Score Calculation

Figure 3 shows the process for calculating the infrastructure score. The infrastructure score is calculated using the number of followers and the K-Shell iteration value of each user in the social graph. A high infrastructure score indicates that the user or the user's followers have high centrality in the graph structure or a large number of followers, which makes information propagation more effective.

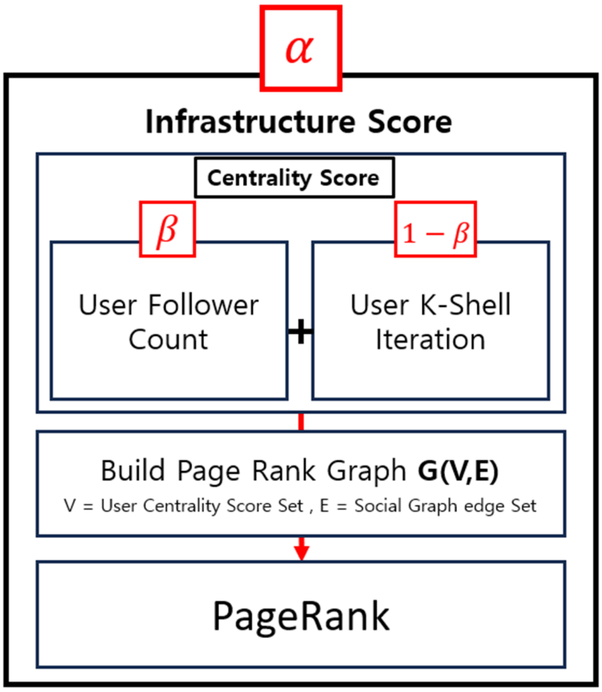

Figure 3. Process of infrastructure score calculation.

The centrality score is obtained by assigning weights to the number of followers and the K-Shell iteration value of a user and summing them. This centrality score is used as the initial PageRank score, and the final PageRank score, obtained after iterating until the convergence condition (10^{-6}) [5] is met, is used as the infrastructure score. This score is denoted as PR_i. The infrastructure score is calculated according to Equation (1). The assigned weight β is a user-defined value that can be adjusted depending on the application, and a value of 0.5 is used in the experiments in this paper.

$$PR_i = PR((\beta \times F_i) + ((1 - \beta) \times F(it_i))) \quad (1)$$

3.3.2. Dissemination Score Calculation

Figure 4 shows the process for calculating the dissemination score. The dissemination score is calculated using the number of retweets and the K-Shell iteration value of each user in the retweet graph. A high dissemination score indicates that the user receives a high level of attention from surrounding users, making it more favorable for information propagation. The dissemination score is calculated by assigning weights of γ and 1-γ to the number of retweets and the retweet K-Shell iteration value, respectively, and summing them. The dissemination score is denoted as DS_i and is calculated according to Equation (2). The assigned weight γ is a user-defined value that can be adjusted depending on the application, and a value of 0.5 is used in the experiments in this paper.

$$DS_i = (\gamma \times R_i) + ((1-\gamma) \times R(it_i)) \quad (2)$$

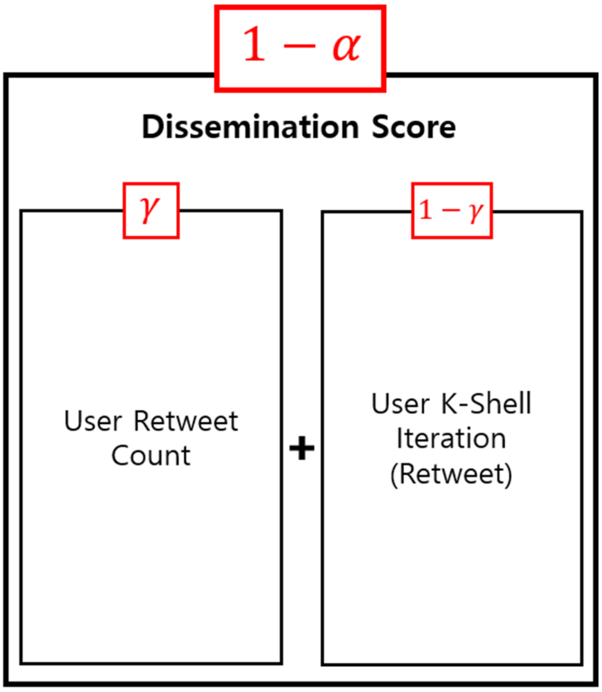

Figure 4. Process of dissemination score calculation.

3.4. Top-K Rank

The influencer score, called the User Influence Score (UIS), is obtained by multiplying the infrastructure score and the dissemination score by their respective weights and summing the results. The formula is expressed in Equation (3), where the weights for the infrastructure score and the dissemination score are determined through hyperparameter tuning to achieve optimal performance. The final influencer score is calculated for all nodes in the graph. The users are then sorted in descending order based on their UIS, and the top-k users are selected as the initial propagators of information. The assigned weight α is a user-defined value that can be adjusted depending on the application, and a value of 0.5 is used in the experiments in this paper, as PR_i and DS_i are assumed to have equal importance. If optimized, for example, through grid search, the performance of the proposed method could be improved.

$$UIS_i = (\alpha \times PR_i) + ((1-\alpha) \times DS_i) \quad (3)$$

Algorithm 1 shows the algorithm for Influencer score calculation. Input is a social graph and a retweet graph. Output is Rank [v]. Lines 01~08 show the Feature Analysis step. Lines 09~16 show the Influencer Score Calculation step. Lines 17~20 show the Top-k Rank step.

Algorithm 1. Influencer Score Calculation

input	Social graph G < V, E>, Retweet graph G_r < V_r, E_r>
output	Rank [v]
1	foreach node v in V
2	add v indegree edges to F_i
3	K_SHashMap = K-Shell Decomposition (G)
4	foreach node v in V_r
5	add v indegree edges to R_i
6	K_{SR}HashMap = K-Shell Decomposition (G_r)
7	foreach node v in V_r
8	DSi = (Equation (2))
9	G_{PR} < V_{PR}, E_{PR}> = Empty Graph
10	foreach node v in V
11	add v to V_{PR}
12	foreach edge e in E
13	add e to E_{PR}
14	foreach node v in V_{PR}
15	v's initial page score = (Equation (1))
16	Infrastructure Score Map < V_{ism}, score > = PageRank (G_{PR})
17	for each node v in V
18	UIS_i = (Equation (3))
19	sort UIS_i list by descending
20	return UIS list

4. Performance Evaluation

4.1. Evaluation Environments

To demonstrate the effectiveness of the proposed influencer detection method, a performance evaluation is conducted with existing influencer detection methods [5,6]. The comparison evaluates both processing time and information propagation performance. In particular, we compare the results of the Independent Cascade (IC) simulation to show that considering both social interactions and graph structure is advantageous for information propagation. The performance evaluation environment is summarized in Table 1. The proposed method utilizes the Higgs Twitter Dataset [21] provided by Stanford SNAP for finding influencers. The graphs used for performance evaluation are listed in Table 2.

Table 1. Performance evaluation environments.

Environment	Specification
Processor	Intel(R) Core (TM) i7-9700K CPU @ 3.60 GHz
Memory	RAM 24.0 GB
OS	Windows 10 64 bit
Language	Python 3.10.1

Table 2. Data set.

Data	V	E	Clustering Coefficient
Social Graph	456,626	14,855,842	0.1887
Retweet Graph	256,491	328,132	0.0156

4.2. Evaluation Results of IC Model Propagation According to the Top-K Values

Figures 5–8 show the IC model propagation according to the top-k value of the social graph and retweet graph. For the proposed method, the top-k nodes in both the social graph and the retweet graph were evaluated with k = 5, 10, 15, 20, 30, 50, and 100. For each top-k, 100 runs of the Independent Cascade (IC) model were performed, and the average number of influenced nodes was measured. In addition, the ratio of the number of influenced nodes to k (i.e., the number of nodes influenced per influencer) was calculated to identify the k value that provides the highest propagation efficiency per influencer.

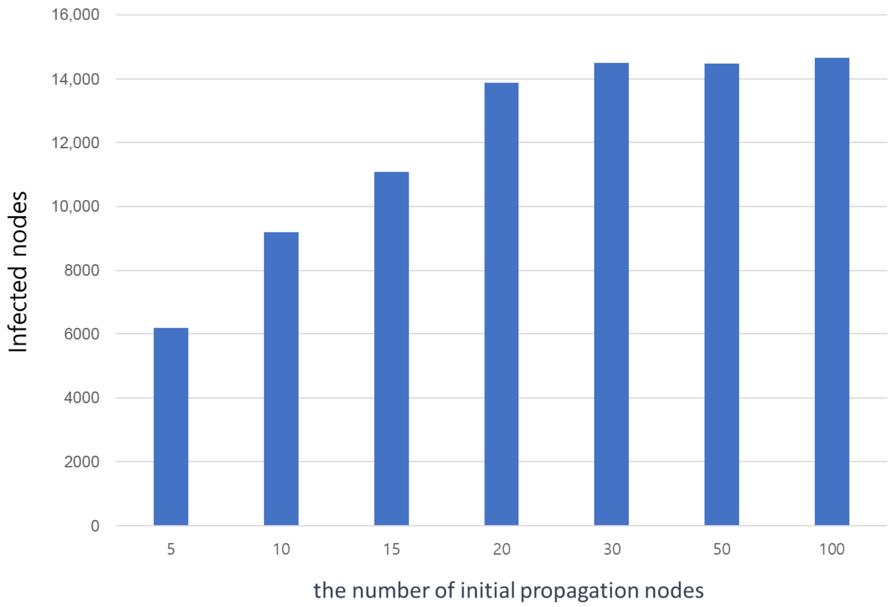

Figure 5. Result of propagation simulation according to the k (Social Graph).

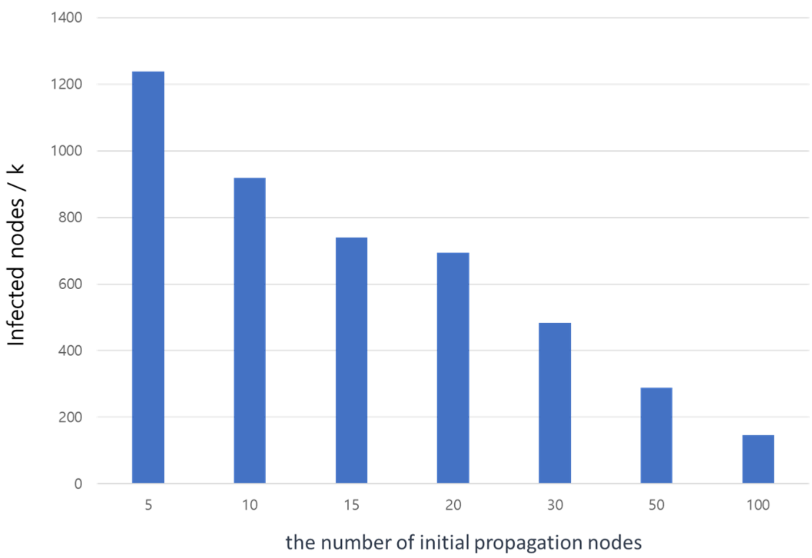

Figure 6. Result of propagation simulation/k according to the k (Social Graph).

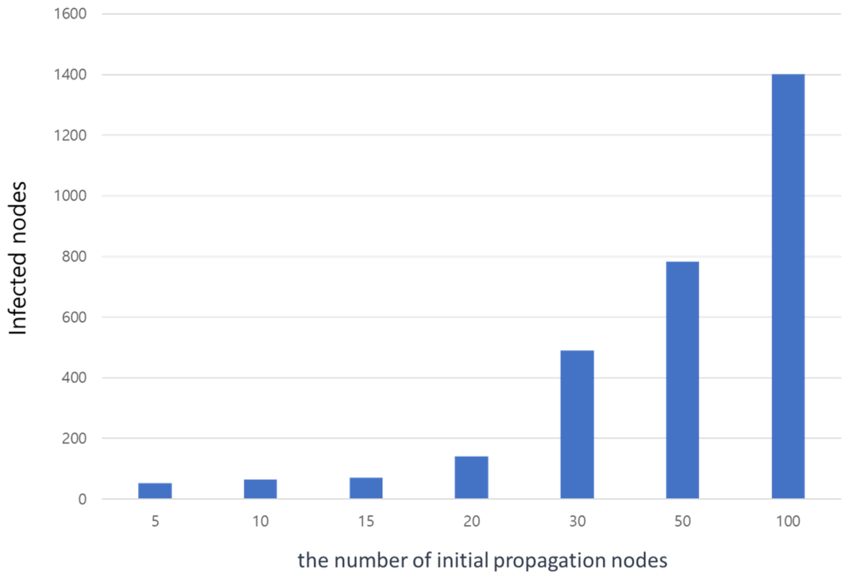

Figure 7. Result of propagation simulation according to the k (Retweet Graph).

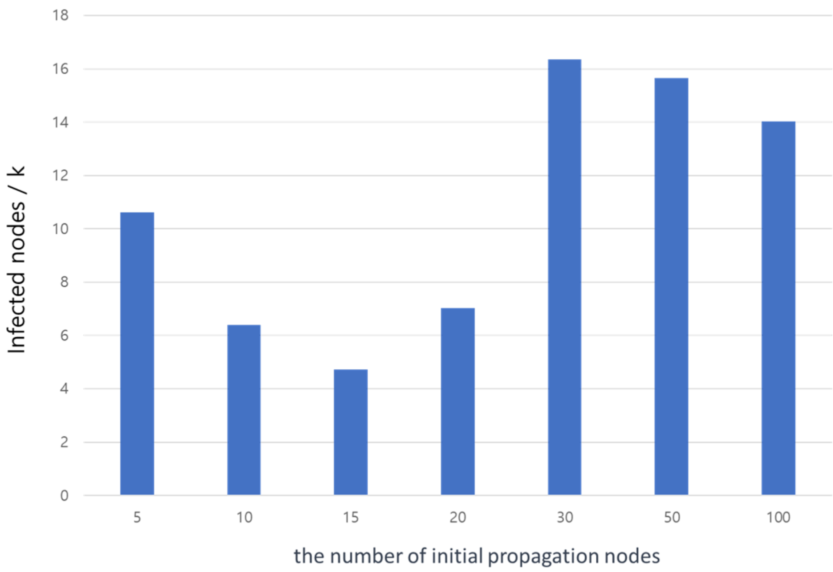

Figure 8. Result of propagation simulation/k according to the k (Retweet Graph).

In all bar chart figures in Section 4, the x-axis represents k, indicating the number of initial propagators in the IC simulation, and the y-axis represents the number of nodes influenced by the IC simulation. As shown in Figure 5, the average propagation gradually increases as k increases, reaching the maximum value when k = 100, with an average of 14,666 influenced nodes. However, the increase in average propagation slows down when k exceeds 30. In a limited network, the diffusion process increasingly involves redundant coverage of vertices, which explains why adding more initial propagators does not yield proportional improvements in propagation efficiency. The average propagation per influencer provides an approximate measure of how many nodes each influencer propagates to, and it was highest at top-5, with a value of 1238. The IC edge propagation probability was set to 0.003.

For the retweet graph, unlike the social graph, the increase in average propagation became larger as k increased. In particular, the average propagation exhibited a sudden surge when k reached 30. Furthermore, the value of average propagation per influencer was also highest at k = 30. This indicates that, in the network used in the experiments, k = 30 enables the most effective diffusion to cover the entire network. This result suggests that, even when different datasets are used, the propagation efficiency will similarly increase gradually as k increases and reach its maximum at a certain value of k.

4.3. Comparison of Evaluation Results with Existing Methods

Figure 9 shows the comparison of execution time between the proposed method and existing methods. The execution time was measured from the moment the code was run until the top-k influencers were identified. The existing methods compared were Follower, PR, and ECRM.

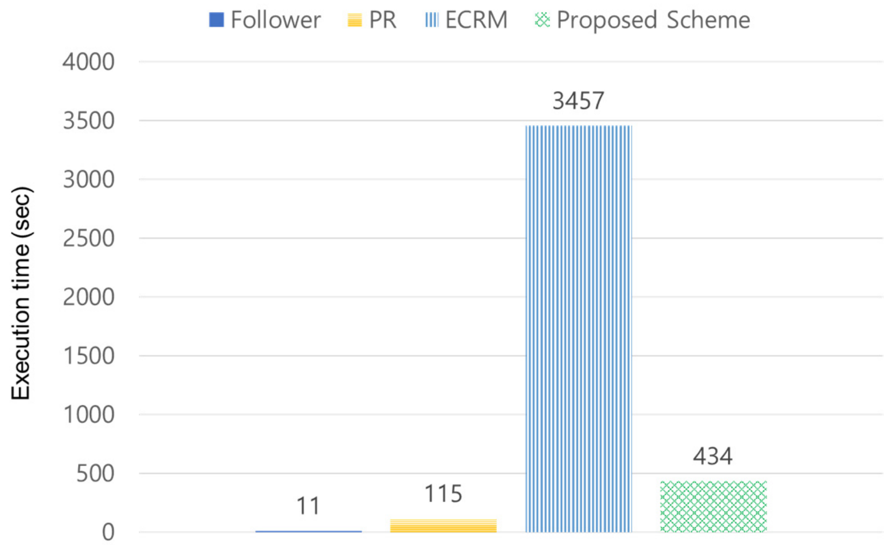

Figure 9. Comparison of execution time.

The Follower method simply selects the top-k users with the highest number of followers. PR selects the top-k users based on PageRank scores computed from the social network. This complexity is known to be $O\,(T \cdot \mid E \mid)$. ECRM calculates the number of followers and the K-Shell similarity between users and their neighbors. The complexity of k-shell decomposition is known to be $O(\mid V \mid + \mid E \mid)$. As a result, when combined with the cost of computing similarity, ECRM exhibits a very high overall computational complexity. In this case, the neighbor similarity is considered up to two hops to compute the influencer score and select the top-k users.

The performance evaluation results show that the Follower method had the fastest execution time, while ECRM exhibited the lowest performance. This is because the Follower method simply selects users with many followers as influencers. The proposed method achieved the third-best performance. Since the proposed method includes PR calculations, it takes longer to execute than PR alone. The proposed method performs k-shell decomposition on each graph and executes PageRank once to compute the infrastructure score. Therefore, its complexity can be regarded as $O(\mid V \mid + \mid E \mid + T \cdot \mid E \mid)$. Finally, although not shown in Figure 9, machine learning-based methods [17–20] are analyzed to require a substantial amount of training time and to exhibit higher execution times than the ECRM method.

Figure 10 shows the results of the propagation simulation on the social graph. In the IC model, a method is considered more effective if a larger number of nodes are influenced when the simulation ends. To achieve a high number of influenced nodes, it is important to select the initial propagators carefully. In this experiment, the top-k users selected by each method were used as the initial propagators. The values of k were set to 5, 10, 15, 20, 30, 50, and 100, representing the number of initial propagators in the IC simulation.

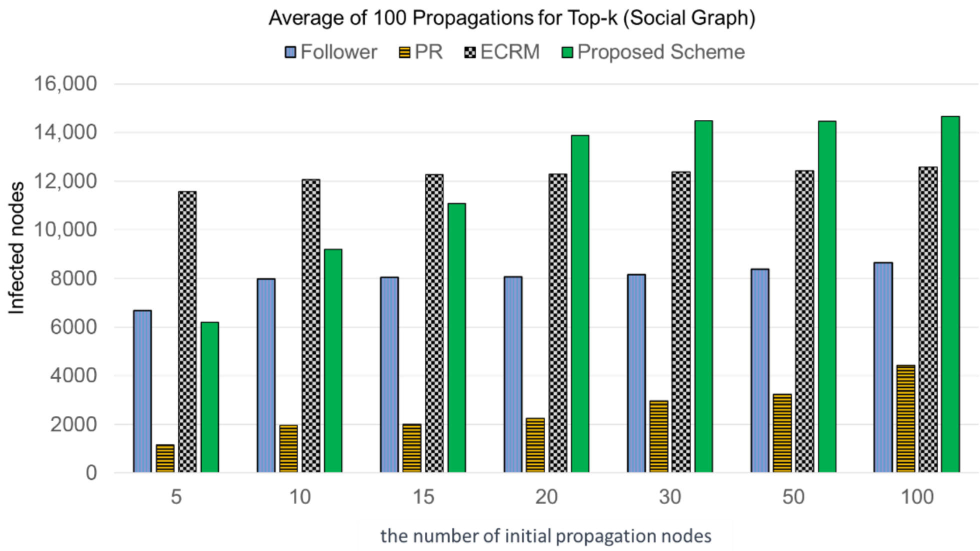

Figure 10. Result of propagation simulation (Social Graph).

The evaluation results show that the ECRM method performs best when k is between 5 and 15. However, for k values of 20 and above, the proposed method exhibits the highest performance. These results demonstrate that the proposed method achieves superior propagation performance on social media graphs.

Figure 11 shows the results of the propagation simulation on the retweet graph. The evaluation results indicate that the proposed method outperforms all existing methods across all values of k. Existing methods, which consider only social features, exhibited poor propagation performance on the retweet graph, with particularly large differences observed when $k \geq 30$.

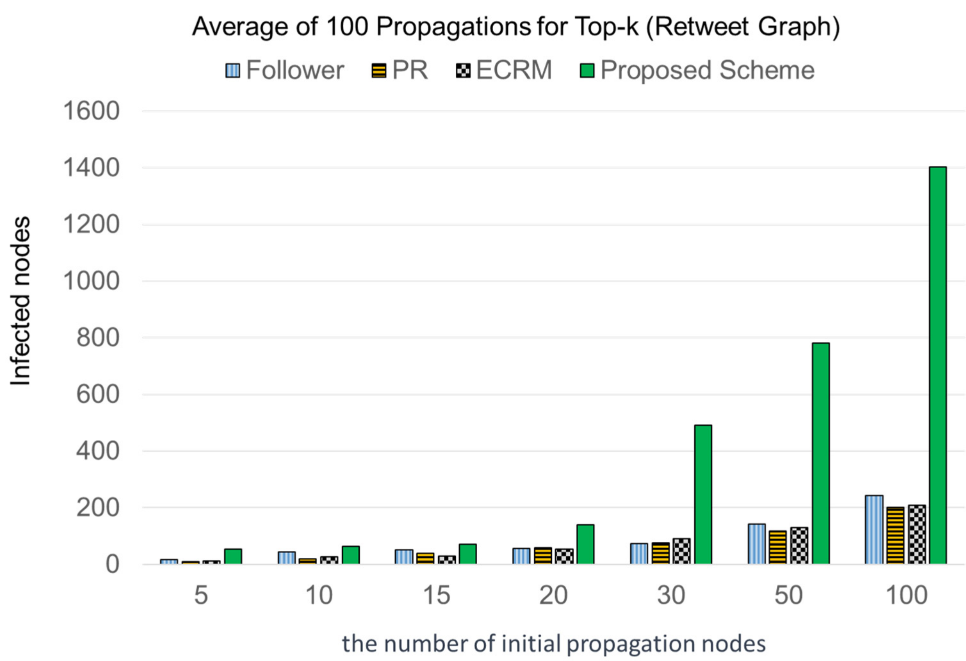

Figure 11. Result of propagation simulation (Retweet Graph).

Since the proposed method demonstrated superior performance in propagation simulations on both the social graph and the retweet graph, it is expected to perform effectively in practical applications of finding influencers. For machine learning-based methods [17–20], although their learning strategies differ, they are similar to ECRM in they do not incorporate propagation characteristics. As a result, their influencer selection outcomes are expected to be comparable, and the corresponding diffusion simulation results are likely to be similar to, or slightly better than, those of ECRM.

5. Conclusions

In this paper, we propose an influencer detection method based on social interaction and graph structure in social media. The proposed method combines the dissemination score, calculated from an influencer's social interactions, with the infrastructure score, derived from the graph structure. By considering both social interactions and graph structure, the proposed method addresses the issue of users with many inactive followers receiving disproportionately high influence scores.

Since the proposed method takes into account both dissemination and infrastructure characteristics, it can prevent the influence of recently inactive users from remaining excessively high. Performance evaluation results demonstrate that the proposed method achieves superior propagation performance on both social and retweet graphs.

One advantage of the proposed method is that it considers both social and retweet features. However, when considering retweet features, the propagation values can vary significantly depending on the relationships between users and the timing of interactions. This limits the method's ability to effectively reflect these variations in the edge weights. Therefore, future work will focus on developing an influencer detection method that incorporates various features through both infrastructure and dissemination scores to further enhance performance. In addition, we will conduct various experiments to demonstrate that the proposed method works effectively across different network environments with machine-learning-based methods such as graph neural network-based methods, time-series network-based influence maximization models, or methods combining deep representation learning. Furthermore, we will enhance accuracy by incorporating propagation characteristics into the methods. Finally, to enhance the robustness of the proposed method, we will improve the proposed method by referring to studies such as [22,23].

Author Contributions: Conceptualization, J.L., H.C., S.C., K.B. and J.Y.; methodology, J.L., H.C., S.C., K.B. and J.Y.; software, H.C.; validation, J.L., H.C., S.C., K.B. and J.Y.; formal analysis, J.L., H.C., S.C., K.B. and J.Y.; data curation, J.L. and H.C.; writing—original draft preparation, J.L. and H.C.; writing—review and editing, K.B., J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Regional Innovation System & Education (RISE) program through the Chungbuk Regional Innovation System & Education Center, funded by the Ministry of Education (MOE) and the Chungcheongbuk-do, Republic of Korea (2025-RISE-11-014-03, 25%), Chungbuk National University BK21 program (2025, 25%), Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2026-RS-2020-II201462, 25%) and the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00245650, 25%).

Data Availability Statement: The original data presented in the study are openly available in Higgs Twitter Dataset at https://snap.stanford.edu/data/higgs-twitter.html (accessed on 21 May 2025) [21].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zareie, A.; Sheikhahmadi, A.; Jalili, M.; Fasaei, M.S.K. Finding influential nodes in social networks based on neighborhood correlation coefficient. *Knowl. Based Syst.* **2020**, *194*, 105555. [CrossRef]

2. Rashid, Y.; Bhat, J.I. Topological to Deep Learning Era for Identifying Influencers in Online Social Networks: A Systematic Review. *Multimed. Tools Appl.* **2024**, *83*, 14671–14714. [CrossRef]
3. Sihombing, P.; Nababan, E.B.; Budiman, M.A. The combination of the Leiden algorithm and graph coloring for influencers detection in large scale social networks. In Proceedings of the 2025 International Conference on Computer Sciences, Engineering, and Technology Innovation (ICoCSETI), virtually, 21 January 2025.
4. Alsaif, S.A.; Hidri, A.; Hidri, M.S. Towards inferring influential Facebook users. *Computers* **2021**, *10*, 62. [CrossRef]
5. Arora, A.; Bansal, S.; Kandpal, C.; Aswani, R.; Dwivedi, Y. Measuring social media influencer index insights from Facebook, Twitter and Instagram. *J. Retail. Consum. Serv.* **2019**, *49*, 86–101. [CrossRef]
6. Ren, J.; Wang, C.; Liu, Q.; Wang, G.; Dong, J. Identify influential spreaders in complex networks based on potential edge weights. *Int. J. Innov. Comput. Inf. Control* **2016**, *12*, 581–590.
7. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1996**, *58*, 267–288. [CrossRef]
8. Mack, Y.P. Local properties of k-NN regression estimates. *SIAM J. Algebr. Discret. Methods* **1981**, *2*, 311–323. [CrossRef]
9. Brin, S.; Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117. [CrossRef]
10. Yang, M.; Wang, H.; Wei, Z.; Wang, S.; Wen, J.R. Efficient algorithms for personalized PageRank computation: A survey. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 4582–4602. [CrossRef]
11. Ahmadi Beni, H.; Bouyer, A. Identifying influential nodes using a shell-based ranking and filtering method in social networks. *Big Data* **2021**, *9*, 219–232. [CrossRef] [PubMed]
12. Xian, Y.; Li, M.; Zhang, Q. A k-Shell Decomposition Structural Entropy of Complex Networks. *Phys. A Stat. Mech. Its Appl.* **2025**, *676*, 130859. [CrossRef]
13. Kempe, D.; Kleinberg, J.; Tardos, É. Maximizing the spread of influence through a social network. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003.
14. Kumar, S.; Mallik, A.; Khetarpal, A.; Panda, B.S. Influence maximization in social networks using graph embedding and graph neural network. *Inf. Sci.* **2022**, *607*, 1617–1636. [CrossRef]
15. Jaouadi, M.; Romdhane, L.B. A Survey on Influence Maximization Models. *Expert Syst. Appl.* **2024**, *248*, 123429. [CrossRef]
16. Yanchenko, E.; Murata, T.; Holme, P. Influence Maximization on Temporal Networks: A Review. *Appl. Netw. Sci.* **2024**, *9*, 16. [CrossRef]
17. Hemnath, M.R. Social media influence maximization prediction using improved graph neural networks. In Proceedings of the National Conference on AI & DS: Challenges and Opportunities in the Digital Era (NCAI-DS 2025), Tamil Nadu, India, 4 March 2025.
18. Huang, J.; Wang, S.; Xia, C.; Yang, Y.; Wang, W. Solving the influence maximization problem via a deep-reinforcement-learning-guided evolutionary approach. In Proceedings of the 2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS), Hangzhou, China, 16–18 August 2024.
19. Ling, C.; Jiang, J.; Wang, J.; Thai, M.T.; Xue, R.; Song, J.; Zhao, L. Deep Graph Representation Learning and Optimization for Influence Maximization. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023.
20. Chen, T.; Yan, S.; Guo, J.; Wu, W. ToupleGDD: A Fine-Designed Solution of Influence Maximization by Deep Reinforcement Learning. *IEEE Trans. Comput. Soc. Syst.* **2023**, *11*, 2210–2221. [CrossRef]
21. Leskovec, J. Higgs Twitter Dataset. Available online: https://snap.stanford.edu/data/higgs-twitter.html (accessed on 21 May 2025).
22. Ou, Z.; Wang, S. Finding robust and influential nodes on directed networks using a memetic algorithm. *Swarm Evol. Comput.* **2024**, *87*, 101542. [CrossRef]
23. Wang, S.; Jin, Y. MFEA-RCIM: A multifactorial evolutionary algorithm for determining robust and influential seeds from competitive networks under structural failures. *IEEE Trans. Cybern.* **2025**, *55*, 3624–3636. [CrossRef] [PubMed]