



## A New Method for Detecting Influential Nodes in Social Network Graphs Using Deep Learning Techniques

Azad Noori<sup>1\*</sup>

<sup>1</sup>Faculty Member, Department of Computer Engineering, Faculty of Electrical and Computer Engineering, Technical and Vocational University (TVU), Tehran, Iran.

### ARTICLE INFO

**Article Type:**  
Original Research

**Received:** 10.19.2021  
**Revised:** 03.08.2022  
**Accepted:** 03.14.2022

**Keyword:**  
Social Networks  
Complex Graphs  
Influence Maximization  
Deep Learning  
Sparse Autoencoder

**\*Corresponding Author:**  
Azad Noori  
**Email:** [a-noori@tvu.ac.ir](mailto:a-noori@tvu.ac.ir)

### ABSTRACT

One of the most important issues in large social networks is identifying influential users to maximize the diffusion of news and messages which is popularly known as the Social Influence Maximization Problem (SIM Problem). The success of the diffusion process in these networks depends on the influential users' selection mechanism. On the other hand, with the increase in growth rate and data size in the graph of large social networks, one of the main challenges is the large number of nodes and edges which makes any processing problematic. Implementing traditional methods on large graphs with high-dimensional data is difficult and time consuming, and more efficient methods must be used. In this paper, a new method for reducing the graph size of social networks using deep learning is proposed, followed by providing a novel and effective solution to the Social Influence Maximization Problem by considering the minimum overlap between nodes. The findings of the simulation in the real world show better performance of the proposed method in terms of execution time and spread of influence than traditional techniques.



---

## EXTENDED ABSTRACT

---

### Introduction

With the emergence of social networks in recent years, studying, processing and extracting available information in these networks has attracted the attention of many researchers. This information has various applications in numerous industries and businesses, and is used by various stakeholders. One essential aspect is the identification of influential users in the network or the problem of maximizing the impact in social networks due to its importance in various applications such as viral marketing, targeted advertising, and recommendation systems. Personalized messages and the selection of influential Twitters have been the focus of researchers and various solutions have been proposed in recent years.

With the growth of the scale of social networks and the increase in the number of networks users, which sometimes reach over hundreds of millions of people, and the increase in data dimensions, the graphs produced to display these networks are very complicated and the implementation of traditional methods on these very large graphs is practically impossible and time-consuming. Therefore, it is necessary to use techniques that increase processing speed of operations on these graphs and increase computational efficiency. In this article, a new method, called DeepMCIM, is presented, which uses deep learning (sparse autoencoder) to extract the structure of the graph and then identify the effective nodes on it.

### Methodology

In this article, after pre-processing the data using the sparse autoencoder, the features and the main topology of the graph were extracted. Then, with the aid of the MCIM algorithm, the operation of identifying the effective nodes was carried out considering the overlap of the nodes. The sparse autoencoder greatly reduced the number of parameters required for training, simplified the training process, and overcame the problem of local minima and the problems of traditional autoencoders.

In general, the proposed method, called the DeepMCIM algorithm, can be summarized as follows:

- 1- Data preprocessing, including obtaining the adjacency matrix as well as the adjacency matrix of second-order neighbors.
- 2- Reducing the dimensions of the main graph and extracting its topology using sparse autoencoder.
- 3- Identification of effective nodes using multi-criteria influence maximization (MCIM) algorithm.

### Results and discussion

To evaluate the proposed method (DeepMCIM), it was compared with MCIM, HC, K-SHELL and distance-based coloring method with degree centrality criterion (DCD). 4 real world datasets were used as follows:

- Gnutella dataset
- Pretty Good Privacy (PGP)

- Astro (AST) dataset
- Twitter (TWT) dataset

In the first test, the influence spread of the proposed method (DeepMCIM) and other methods were compared on different datasets. The size of the initial basket in this experiment was fixed and equal to 40 for all algorithms. In addition, the UM method was used to determine the value of  $\beta$ , and its value was considered equal to 0.06 for all. The SIR diffusion model was used to evaluate the diffusion capability of the primary portfolio in different ways. For this purpose, the nodes belonging to the initial basket set were first placed in the I state and the other nodes in the S state, and the expansion process was started by this model. At the end of the process, the number of nodes that were in the R state was considered as the expansion of the initial basket collection.

The results of the tests showed that the influence spread of the proposed algorithm was better than other methods. Experiments demonstrated that in the Twitter dataset (TWT), which had more nodes and a larger graph than the other datasets, the performance difference of the proposed algorithm was greater than other competitors (Figure 1).

In the second experiment, the proposed method was compared with the above methods in terms of execution time. The results can be seen in Table 1. An increase in the size of the data set, noticeably increased the efficiency of the proposed method in terms of execution time compared to other methods, particularly in the AST and TWT data sets.

**Table1. Comparison of execution time (in milliseconds) of the proposed method and other methods in the real world data set.**


| DEEP-MCIM | MCIM | DCD  | HC    | K-SHELL | Dataset                   |
|-----------|------|------|-------|---------|---------------------------|
| 102       | 171  | 149  | 9125  | 41      | Gnutella (GLA)            |
| 210       | 262  | 236  | 7936  | 12      | Pretty Good Privacy (PGP) |
| 5962      | 9378 | 9216 | 85432 | 98      | Astro (AST)               |
| 4215      | 7363 | 8456 | 36213 | 36      | Twitter (TWT)             |

## Conclusion

In the present research, a new method for extracting the important features of the social network graph was presented, and using deep sparse autoencoders, the topology of the network was extracted performing the operation of reducing the dimensions of the graph. Then, by considering the direct neighbors and second-degree neighbors and using the graph adjacency matrix through the MCIM method, influential nodes were identified. The proposed method was compared and evaluated with 4 other methods (MCIM, DCD, HC, K-SHELL) on the real-world data set. The results of the simulation confirmed the better performance of the proposed method compared to the other methods in terms of execution time and influence spread. In particular, when the size of the dataset was large and had a large number of nodes and edges, and the size of the initial basket was high, the performance of the proposed method was far superior to the other methods. However, for small graphs and with the size of the initial basket, this difference in performance was not very noticeable because in all these methods, usually, the initial elements of the basket are sorted and selected based on centrality criteria such as degree. In addition to the overhead it created, the size reduction operation did not make a noticeable difference to the final results.



## ارائه روشی جدید برای تشخیص گره‌های پرنفوذ در گراف شبکه‌های اجتماعی با استفاده از روش‌های یادگیری عمیق

آزاد نوری\*<sup>1</sup> 

۱- عضو هیات علمی، گروه مهندسی کامپیوتر، دانشکده مهندسی برق و کامپیوتر، دانشگاه فنی و حرفه‌ای، تهران، ایران.

### چکیده

### اطلاعات مقاله

#### نوع مقاله: مقاله پژوهشی

دریافت مقاله: ۱۴۰۰/۰۷/۲۷

بازنگری مقاله: ۱۴۰۰/۱۲/۱۷

پذیرش مقاله: ۱۴۰۰/۱۲/۲۳

#### کلید واژگان:

شبکه‌های اجتماعی  
گراف‌های پیچیده  
پیشینه‌سازی تأثیر  
یادگیری عمیق  
خودرمزنگارهای خلوت

یکی از مسائل مهم در شبکه‌های اجتماعی بزرگ، شناسایی کاربران بانفوذ برای پیشینه‌سازی انتشار اخبار و پیام‌ها است که عموماً تحت عنوان مشکل پیشینه‌سازی تأثیر در شبکه‌های اجتماعی (مشکل SIM)، شناخته می‌شود. موفقیت روند انتشار در این شبکه‌ها بستگی به مکانیسم انتخاب کاربران تأثیرگذار دارد. از طرفی با افزایش سرعت رشد و حجم داده‌ها در گراف شبکه‌های اجتماعی بزرگ یکی از معضلات اصلی، تعداد بسیار زیاد گره‌ها و یال‌هاست که انجام هر نوع پردازشی روی آن را با مشکلات متعدد روبه‌رو می‌سازد. اجرای روش‌های سنتی بر روی گراف‌های بزرگ و دارای داده‌های با ابعاد بالا، سخت و زمان‌بر است و باید روش‌های مؤثرتری به‌کار گرفته شود. در این مقاله ما با استفاده از یادگیری عمیق، روش جدیدی برای کاهش ابعاد گراف شبکه‌های اجتماعی پیشنهاد داده و سپس با در نظر گرفتن حداقل هم‌پوشانی بین گره‌ها تلاش می‌کنیم تا راه‌حل جدید و مؤثری را برای مسئله پیشینه‌سازی تأثیر ارائه دهیم. در ادامه نتایج حاصل از شبیه‌سازی در دنیای واقعی، نشان‌دهنده عملکرد بهتر روش پیشنهادی از نظر زمان اجرا و میزان گسترش نفوذ نسبت به تکنیک‌های سنتی است.

\*نویسنده مسئول: آزاد نوری

پست الکترونیکی:

[a-noori@tvu.ac.ir](mailto:a-noori@tvu.ac.ir)

## مقدمه

با ظهور شبکه‌های اجتماعی در سالیان اخیر، مطالعه و پردازش و استخراج اطلاعات موجود در این شبکه‌ها مورد توجه محققان بسیاری قرار گرفت. این اطلاعات، کاربردهای متنوع و مختلفی در صنایع و کسب و کارهای گوناگون دارد و مورد بهره‌برداری ذی‌نفعان مختلف قرار می‌گیرد. امروزه شبکه‌های اجتماعی، نقش مهمی در انتشار اطلاعات، نظر، ایده، نوآوری، شایعات و غیره در مقیاس وسیع ایفا می‌کنند [۱]. این فرایند گسترش، اهمیت زیادی در بازاریابی و پروسی<sup>۱</sup> دارد [۲]. در شبکه‌های اجتماعی بزرگ که معمولاً به شکل گراف نمایش داده می‌شوند، گره‌های گراف معادل کاربران و یال‌های گراف معادل ارتباطات بین کاربران مختلف است (مانند آنچه در شکل ۱ قابل مشاهده است). یکی از حوزه‌های مهم در این قسمت، شناسایی کاربران پرنفوذ در شبکه یا همان مشکل بیشینه‌سازی تأثیر در شبکه‌های اجتماعی<sup>۲</sup> است که با توجه به اهمیت آن در کاربردهای مختلفی مانند بازاریابی و پروسی، تبلیغات هدفمند<sup>۳</sup>، سیستم‌های ارائه‌دهنده توصیه‌های شخصی‌سازی شده<sup>۴</sup> و انتخاب توئیترهای تأثیرگذار [۳]، مورد توجه محققان بوده و راه‌حل‌های مختلفی در سالیان اخیر پیشنهاد شده است. برای مثال تبلیغ نام تجاری (برند) از طریق بازاریابی آنلاین را در نظر بگیرید؛ جایی که هدف، جذب کاربران برای خرید یک محصول خاص است. بهترین راه برای انجام این کار، انتخاب مجموعه‌ای از کاربران بسیار تأثیرگذار و توزیع نمونه رایگان محصول بین آنها است. بسیاری از آنها محصول را دوست خواهند داشت و بر همسایگان خود تأثیر می‌گذارند تا آن محصول را امتحان کنند. این کاربران تازه‌آگاه بر همسایگان خود تأثیر خواهند گذاشت و این روند آشنایی ادامه خواهد یافت و در نهایت بخش بزرگی از کاربران برای تبلیغ محصول تلاش می‌کنند که منجر به بهبود چشمگیر درآمد کسب‌شده می‌شود.

به‌طور طبیعی، تعداد محصولات نمونه رایگان به دلایل اقتصادی، محدود خواهد شد. از این رو، اگر بتوان نمونه‌های رایگان را بین کاربران بسیار تأثیرگذار توزیع کرد این فرایند، ثمربخش خواهد بود. همان‌گونه که پیشتر نیز گفته شد این مسئله به‌عنوان مشکل بیشینه‌سازی تأثیر اجتماعی شناخته می‌شود و روش‌های متعددی برای حل این مسئله ارائه شده است. یکی از منابع خوب در این زمینه که به مروری بر الگوریتم‌های ارائه‌شده در زمینه بیشینه‌سازی تأثیر در شبکه‌های اجتماعی پرداخته است در [۴] آمده است همچنین در [۵] نویسندگان به مروری بر این الگوریتم‌ها پرداخته‌اند. در منبع [۶] نویسندگان با استفاده از روش TOPSIS عوامل مختلفی را برای شناسایی گره‌های مؤثر در نظر گرفته و پس از محاسبه امتیازات مثبت و منفی برای همه گره‌ها، در هر مرحله، گره دارای بیشترین امتیاز را به سید اولیه اضافه کرده‌اند. در [۷-۹] نویسندگان روش‌هایی برای شناسایی گره‌های پرنفوذ براساس تشخیص اجتماعات ارائه کرده‌اند.

از طرفی با رشد مقیاس شبکه‌های اجتماعی و افزایش تعداد کاربران این شبکه‌ها که گاه به بالای صدها میلیون نفر می‌رسند (نظیر شبکه فیسبوک با ۲.۸ میلیارد کاربر [۱۰]) و افزایش ابعاد داده‌ها، گراف‌های تولیدشده برای نمایش این شبکه‌ها بسیار پیچیده می‌شود و اجرای روش‌های سنتی بر روی این گراف‌ها بسیار بزرگ، عملاً امکان‌پذیر نمی‌باشد و بسیار زمان‌بر خواهد بود (ساعت‌ها، روزها و حتی هفته‌ها بسته به اندازه گراف). بنابراین استفاده از روش‌هایی که منجر به افزایش سرعت عملیات پردازش روی این گراف‌ها و افزایش کارایی محاسباتی شود، ضروری به‌نظر می‌رسد

<sup>1</sup> Viral marketing

<sup>2</sup> Social Influence Maximization Problem (SIM Problem)

<sup>3</sup> Target advertisement

<sup>4</sup> Personalized recommendation

[۱۱]. در حال حاضر، یادگیری عمیق<sup>۱</sup> یک راه حل است [۱۲]. با یادگیری عمیق، مدل های محاسباتی می توانند بازنمایی داده ها را در سطوح مختلف انتزاعی یاد بگیرند که کاملاً برای داده های شبکه، مناسب است. علاوه بر این، توانایی آنها برای یادگیری ویژگی های غیرخطی بسیار افزایش می یابد و این تکنیک در طیف وسیعی از زمینه ها که داده ها دارای روابط داخلی هستند، مانند بینایی ماشین و پردازش زبان طبیعی، موفق بوده است. علاوه بر این، شبکه های عصبی عمیق چند لایه می توانند ابعاد داده ها را کاهش دهند [۱۳] که دامنه بالقوه وظایف تجزیه و تحلیل شبکه مانند تشخیص جامعه، طبقه بندی گره ها و پیش بینی پیوند را گسترده تر می کند.



شکل ۱. تصویر گراف شبکه های اجتماعی [۱۴].

از یادگیری عمیق در پردازش تصویر [۱۵]، تشخیص گفتار [۱۶]، امنیت فضای مجازی [۱۷؛ ۱۸] و پردازش زبان طبیعی [۱۹] استفاده می شود. استفاده از روش های یادگیری عمیق به رایانه ها اجازه می دهد تا ویژگی های مهمی که تشخیص آنها با روش های دستی دشوار است را بیاموزند. نویسندگان در [۲۰] الگوریتم جدیدی برای تشخیص اجتماع در گراف های وزن دار با استفاده از یادگیری عمیق ارائه کرده اند. همچنین در [۲۱] روشی بر پایه یادگیری عمیق برای شناسایی گره های پرنفوذ در شبکه ارائه شده است.

در این مقاله، روش جدیدی که به اختصار DEEP-MCIM نامیده می شود ارائه شده است که با استفاده از یادگیری عمیق (خودرمننگارهای خلوت) ساختار گراف استخراج و سپس عملیات شناسایی گره های مؤثر روی آن انجام می شود.

<sup>1</sup> Deep learning

## بیان مسئله

شناسایی مجموعه‌ای از  $k$  کاربر شبکه به‌عنوان سبب اولیه<sup>۱</sup> که تحت یک مدل انتشار خاص است، تعداد کاربرانی که پیام دریافت می‌کنند را به حداکثر می‌رساند.

در ادامه در این مقاله، نخست مدل‌های انتشار<sup>۲</sup> معرفی می‌شود و در ادامه معیارهای مرکزیت<sup>۳</sup> بیان می‌شوند. یکی از ویژگی‌های بارز گراف‌های شبکه‌های اجتماعی، مسئله خلوت بودن<sup>۴</sup> گراف است که در این پژوهش تلاش شده است تا با استفاده از خودرمنزنگارهای خلوت<sup>۵</sup> این مسئله مد نظر قرار گیرد و توپولوژی شبکه استخراج گردد و سپس با استفاده از روش‌های شناسایی گره‌های بانفوذ و در نظر گرفتن مسئله پوشش مستقیم و غیرمستقیم مشابه آنچه در [۶] تحت عنوان الگوریتم MCIM انجام شده است، گره‌های مؤثر در شبکه شناسایی شود.

در بخش بعدی، نتایج حاصل از شبیه‌سازی و آزمایش‌های در شبکه‌های دنیای واقعی و نیز شبکه‌های مصنوعی آورده شده و در ادامه، این الگوریتم با روش‌های مشابه مقایسه شده است که نشانگر بهبود کارایی و عملکرد خوب روش پیشنهادی است.

## مفاهیم مبنایی

در حال حاضر مدل‌های انتشار مختلفی، معرفی شده‌اند و در حال استفاده می‌باشند که برخی از مهم‌ترین آنها در اینجا به‌صورت مختصر، معرفی می‌شود.

## مدل‌های آستانه<sup>۶</sup>

در این مدل‌ها، هر گره می‌تواند در حالت فعال یا غیرفعال باشد. هر گره  $v$  دارای آستانه  $\theta_v$  است و هر یال  $(u, v)$  دارای وزن  $w_{uv}$  است. با توجه به مقادیر آستانه و مجموعه اولیه گره‌های فعال، فرایند در گام‌هایی مجزا ادامه می‌یابد. در هر برچسب زمانی مانند  $t+1$  گره  $v$  از حالت غیرفعال به حالت فعال می‌رود، به شرطی که:

$$\sum_{u \in N_v^a} w_{uv} > \theta_v \quad (1)$$

$N_v^a$  نشان‌دهنده مجموعه همسایگان فعال گره  $v$  در زمان  $t$  است. فرایند تا زمانی ادامه می‌یابد که دیگر امکان فعال‌سازی گره جدیدی میسر نباشد. تعداد نهایی گره‌های فعال، نشان‌دهنده گسترش نفوذ گره‌های فعال اولیه است.

<sup>1</sup> Seed Set

<sup>2</sup> Spreading models

<sup>3</sup> Centrality metrics

<sup>4</sup> Sparsity

<sup>5</sup> Sparse autoencoder

<sup>6</sup> Threshold models

## مدل‌های آبخاری<sup>۱</sup>

مدل‌های آبخاری مستقل نیز یکی دیگر از مدل‌های معروف در این حوزه است. در این روش نیز هر گره می‌تواند در حالت فعال یا غیرفعال باشد. در هر برجسب زمانی، هر گره فعال، هریک از همسایگان غیرفعال خود را با احتمال  $\beta$  به حالت فعال منتقل می‌کند و سپس خود به حالت غیرفعال می‌رود.

## مدل‌های اپیدمیک<sup>۲</sup>

در سال‌های اخیر برای مدل‌سازی فرایند انتشار پیام در شبکه‌های اجتماعی، از مدل‌های اپیدمیک، بسیار استفاده می‌شود. یکی از معروف‌ترین مدل‌های اپیدمیک، مدل  $SIR^3$  است که در پژوهش‌های اخیر بسیار مورد استفاده محققین بوه است [۶؛ ۲۲]. بنابراین ما نیز در این مقاله از همین مدل استفاده خواهیم کرد. در این مدل هر گره می‌تواند در ۳ حالت قرار گیرد:

- ۱- **حساس یا مستعد (S):** کسی که هنوز بیمار نشده اما اگر در معرض بیماری قرار گیرد، آلوده می‌شود.
  - ۲- **آلوده شده (I):** فردی که بیمار شده و می‌تواند دیگران را آلوده کند.
  - ۳- **ریکاوری شده یا بهبود یافته (R):** فردی که در اثر بیماری یا واکسینه شدن بهبود یافته است.
- در ابتدا گره‌های متعلق به مجموعه سبب اولیه در حالت I قرار دارند و گره‌های باقیمانده در حالت S قرار دارند. در هر لحظه، هر گره  $v$  در حالت I به دنبال آلوده کردن همسایگان خود است؛ بنابراین هریک از همسایگان را در حالت S با احتمال  $\beta$  آلوده می‌کند و آنها را به حالت I می‌برد و سپس خودش با احتمال  $\alpha$  به سمت حالت R حرکت می‌کند. این روند تا زمانی که گره‌هایی در حالت I وجود دارد، تکرار می‌شود. سرانجام تعداد گره‌های R، نشان‌دهنده گسترش نفوذ مجموعه سبب اولیه است. برای افزایش دقت، فرایند SIR چندین بار اجرا می‌شود و میانگین تعداد گره‌های R به‌عنوان گسترش نفوذ در نظر گرفته می‌شود.
- برای تعیین میزان نفوذ گره‌ها و رتبه‌بندی آنها بر این اساس، معیارهای مرکزیت<sup>۴</sup> مختلفی وجود دارد از جمله:

## مرکزیت درجه<sup>۵</sup>

در این روش، درجه هر گره و به عبارت دیگر تعداد همسایگان یک گره در نظر گرفته می‌شود. بالا بودن درجه گره، نشان‌دهنده اهمیت آن گره و میزان تأثیرگذاری آن گره بر سایر گره‌هاست.

<sup>1</sup> Cascading models

<sup>2</sup> Epidemic models

<sup>3</sup> Susceptible-Infected-Recovered

<sup>4</sup> Centrality metrics

<sup>5</sup> Degree centrality

### مرکزیت در میان بودن<sup>۱</sup>

در این روش ابتدا کوتاه‌ترین مسیر بین هر دو گره در شبکه مانند (v و u) محاسبه می‌شود و سپس برای هر گره خاص مانند i بررسی می‌شود که چند بار در این کوتاه‌ترین مسیرها حضور دارد. تعداد دفعات حضور گره i در کوتاه‌ترین مسیرها بین هر دو گره دلخواه u و v بیانگر مرکزیت در میان بودن گره i و میزان تأثیر آن است.

### مرکزیت نزدیکی<sup>۲</sup>

در این روش، مرکزیت گره i برابر است با مجموع معکوس فاصله کوتاه‌ترین مسیرها از گره i به سایر گره‌های شبکه. هرچه گره، مرکزی‌تر باشد؛ به همه گره‌های دیگر نزدیک‌تر است.

### مرکزیت K-SHELL<sup>۳</sup>

در این روش، به هر گره یک شاخص K-SHELL بر اساس مکان‌های توپولوژیکی گره‌ها در شبکه اختصاص داده می‌شود؛ به طوری که گره‌هایی که به هسته شبکه نزدیک‌تر هستند دارای K-SHELL بالاتری هستند. گره‌هایی با K-SHELL بیشتر به‌عنوان گره‌های تأثیرگذار در نظر گرفته می‌شوند. روش‌های متعدد دیگری نیز برای محاسبه مرکزیت وجود دارد و هرکدام از آنها دارای مزایا و معایب خاص خود است. از این رو برخی از محققان در پژوهش‌های خود به‌جای انتخاب یکی از این روش‌ها برای رتبه‌بندی گره‌ها از نظر میزان تأثیر، از چند معیار به‌طور هم‌زمان استفاده کرده‌اند. در این روش‌ها، هریک از گره‌ها به‌عنوان راه‌حلی با معیارهای مختلف مانند مرکزیت درجه، مرکزیت درمیان بودن و غیره در نظر گرفته می‌شود. بنابراین، مشکل تعیین تأثیر گسترش گره‌ها و رتبه‌بندی آنها به‌عنوان یک مشکل چندمعیاره تعریف شده است و گره‌ها سپس با استفاده از روش TOPSIS رتبه‌بندی شده‌اند.

علاوه بر رتبه‌بندی گره‌ها یا کاربران، مسئله بعدی، حذف گره‌ها یا کاربرانی است که محدوده مشترکی از شبکه را پوشش می‌دهند و در واقع کار یکسانی انجام می‌دهند. محققان متعددی درباره این مسئله تحقیق کرده‌اند و علاوه بر موارد فوق، مسئله پوشش گره‌ها را نیز مورد توجه قرار داده‌اند مانند زارعی و همکاران [۶] که ما نیز در این راه‌حل، همسایگان مستقیم و همسایگان غیرمستقیم (مرتب دوم) را در نظر می‌گیریم.

### روش پیشنهادی

ابتدا تعاریف موردنیاز در زیر ارائه می‌شود و سپس روش پیشنهادی ارائه می‌گردد:  
فرض می‌کنیم که گراف بدون وزن و بدون جهت  $G=(V,E)$  را داریم به‌گونه‌ای که:

$$V=\{v_1, v_2, \dots, v_n\} \quad \text{مجموعه گره‌ها}$$

$$E=\{e_1, e_2, \dots, e_n\} \quad \text{مجموعه یال‌ها}$$

<sup>1</sup> Betweenness centrality

<sup>2</sup> Closeness centrality

<sup>3</sup> K-Shell centrality

### تعریف ۱ (ماتریس مجاورت)

ماتریس مجاورت برای یک گراف با  $n$  رأس (گره)، یک ماتریس  $n \times n$  است که در این مقاله آن را با نماد  $X$  نشان می‌دهیم. مقدار درایه  $i$ ام ماتریس مجاورت، برابر با تعداد یال‌های بین رأس  $i$  و رأس  $j$  است که در مورد گراف ما این مقدار ۰ یا ۱ است.

### تعریف ۲ (همسایگان مرتبه دوم یک گره)

اگر  $N(v)$  مجموعه همسایگان گره  $v$  باشد، آن‌گاه مجموعه همسایگان همه گره‌های موجود در  $N(v)$  به جز خود گره  $v$  را مجموعه همسایگان مرتبه دوم  $v$  می‌نامیم. برای به دست آوردن ماتریس مجاورت همسایگان مرتبه دوم، کافی است ماتریس مجاورت را به توان ۲ برسانیم که در این مقاله آن را با نماد  $Z$  نمایش می‌دهیم.

### استخراج ویژگی

برای استخراج ویژگی‌های توپولوژیکی شبکه، از خودرمنگاری‌های خلوت عمیق<sup>۱</sup> یا به طور خلاصه خودرمنگاری خلوت عمیق استفاده می‌شود. ابتدا، مفهوم خودرمنگاری را معرفی می‌کنیم. سپس، یک خودرمنگاری خلوت عمیق می‌سازیم، ویژگی‌های ماتریس مجاورت  $X$  یک شبکه را استخراج می‌کنیم و ماتریس کم بعد  $H$  را که توپولوژی شبکه را بیان می‌کند، به دست می‌آوریم.

خودرمنگاری، یک ابزار یادگیری عمیق بدون نظارت<sup>۲</sup> است که برای استخراج ویژگی و فشرده‌سازی نمایش داده‌های با ابعاد بالا، یا به عبارت دیگر برای کاهش ابعاد استفاده می‌شود. خودرمنگاری ابتدا ورودی را فشرده و ویژگی‌های آن را استخراج می‌کند (نمایش داده با ابعاد کم)، سپس تلاش می‌کند تا با استفاده از این ویژگی‌های استخراج‌شده، ورودی خود را بازسازی کند. هر چقدر که عمل بازسازی ورودی با موفقیت بیشتری انجام شود و خروجی، شباهت بیشتری به ورودی داشته باشد؛ یعنی اختلاف بین ورودی و خروجی کمتر باشد، نشان‌دهنده موفقیت‌آمیز بودن کاهش ابعاد و استخراج ویژگی‌های درست است.

خودرمنگاری از یک انکدر<sup>۳</sup> و یک دیکدر<sup>۴</sup> تشکیل می‌شود. انکدر می‌تواند ورودی را به یک نمایش کم‌بعد<sup>۵</sup>، فشرده کند. این نمایش کم‌بعد در واقع همان ویژگی‌های استخراج‌شده از ورودی است. در مرحله بعد دیکدر با استفاده از این نمایش کم‌بعد، ورودی را بازسازی می‌کند.

ما بردار  $x_i \in R^{n \times l}$  را از ماتریس مجاورت  $X = \{x_1, x_2, \dots, x_n\}$  برمی‌داریم و به عنوان ورودی خودرمنگاری استفاده می‌کنیم تا بردار کاهش‌بعدیافته  $h_i$  را به دست آوریم. سپس بردار کاهش‌بعدیافته  $h_i$  را به دیکدر می‌دهیم تا بردار  $X'_i$  را که هم‌اندازه  $X_i$  است به دست آوریم. اگر بردار  $X'_i$  بسیار شبیه به بردار ورودی  $X_i$  باشد، آن‌گاه بردار کاهش‌بعدیافته

<sup>1</sup> Deep sparse autoencoders

<sup>2</sup> Unsupervised

<sup>3</sup> Encoder

<sup>4</sup> Decoder

<sup>5</sup> Latent spatial representation

$h_i$  می‌تواند به‌طور مؤثری اطلاعات بردار اصلی را نمایش دهد. ما شبکه را با استفاده از الگوریتم پس‌انتشار خطا<sup>۱</sup> آموزش می‌دهیم و پارامترهای آنکدر و دیکدر را به‌گونه‌ای تنظیم می‌کنیم که منجر به حداقل شدن خطای بازسازی شود. در این مرحله، بردار خروجی  $X'_i$  را برابر بردار ورودی  $X_i$  قرار می‌دهیم و در نهایت بردار  $h_i$  به‌دست‌آمده را به‌عنوان ویژگی‌های استخراج شده از  $X_i$  به‌دست می‌آوریم.

همان‌گونه که در بالا گفته شد از ماتریس مجاورت  $X$  به‌عنوان ورودی خودرمننگار (رمننگار) استفاده می‌کنیم. هر بردار سطری  $x_i \in R^{n \times 1}$  از ماتریس  $X$  نمایانگر وجود ارتباط بین گره  $i$  و سایر گره‌های شبکه است.  $X_i$  ورودی آنکدری با  $k$  نورون است و مقدار  $h_i$  برای هر لایه آنکدر با فرمول زیر به‌دست می‌آید:

$$h_i = s(wx_i + b) \quad (۲)$$

که  $s$  تابع فعال‌ساز<sup>۲</sup> لایه آنکدر است که ما در این مقاله از تابع سیگموئید<sup>۳</sup> که در زیر آورده شده است، استفاده می‌کنیم.  $w$  ماتریس وزن‌ها و  $b$  بایاس است.

$$s(x) = \frac{1}{1 + e^{-x}} \quad (۳)$$

اکنون بردار  $h_i$  از عملیات رمننگاری به‌دست می‌آید که در واقع مربوط به گره  $i$  ام است و ما از این بردار به‌عنوان ورودی لایه دیکدر (رمنگشا) استفاده می‌کنیم. حال از رابطه زیر برای آموزش و به‌دست آوردن خروجی لایه دیکدر استفاده می‌کنیم.

$$x'_i = s(w'h_i + b') \quad (۴)$$

$w'$  ماتریس وزن‌دار لایه دیکدر (رمنگشا) است و  $b'$  بایاس یا افست بردار لایه رمنگشایی است. طی فرایند آموزش، مقادیر  $w$ ،  $w'$ ،  $b$  و  $b'$  از طریق الگوریتم پس انتشار خطا و به‌صورت اتوماتیک، به‌گونه‌ای تنظیم می‌شود که منجر به حداقل شدن اختلاف بین  $X'_i$  و  $X_i$  شود. یعنی مقدار زیر، کمینه شود.

$$\begin{aligned} L(x, x') &= \|x - x'\|^2 \\ &= \|x - s(w'h_i + b')\|^2 \\ &= \|x - s(w's(wx_i + b) + b')\|^2 \end{aligned} \quad (۵)$$

برای بهبود عملکرد و افزایش کارایی، از مفهوم خلوت بودن برای واحدهای لایه پنهان استفاده می‌کنیم. در مورد تابع فعال‌سازی سیگموئید برای نورون‌ها، اگر خروجی نورون ۱ باشد، نورون فعال می‌شود. در غیر این صورت، نورون فعال نمی‌شود. علاوه بر این، واژه خلوت به این معنی است که در اکثر موارد، نورون فعال نمی‌شود. ما می‌توانیم از

<sup>۱</sup> Back-propagation

<sup>۲</sup> Activation function

<sup>۳</sup> Sigmoid

خروجی نورون به عنوان درجه فعال سازی نورون استفاده کنیم و از واگرایی<sup>۱</sup> KL برای افزودن یک قید خلوت بودن به خودرمننگار استفاده کنیم.

$$\sum_{j=1}^k KL\left(\rho \parallel 1/n \sum_{i=1}^n h_i\right) \quad (6)$$

بنابراین خطای بازسازی برای خودرمننگار پیشنهادی به شکل زیر محاسبه می شود.

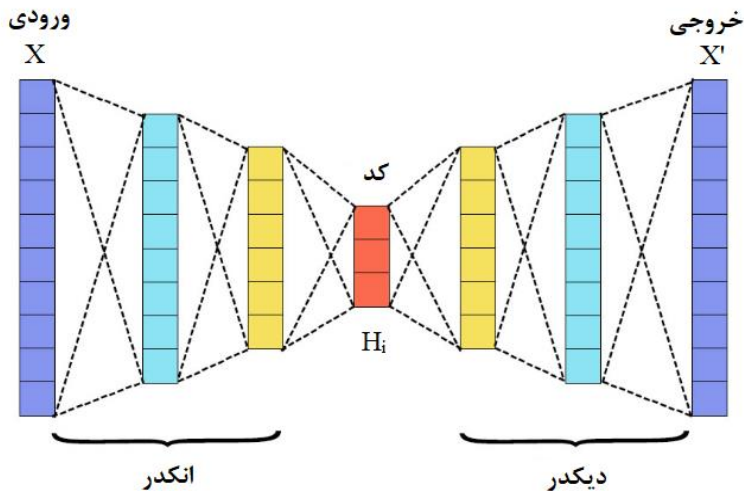
$$L(x, x') = \|x - s(w's(wx_i + b) + b')\|^2 + \quad (7)$$

$$\partial \sum_{j=1}^k KL\left(\rho \parallel 1/n \sum_{i=1}^n s(wx_i + b)\right)$$

خودرمننگار خلوت، تعداد پارامترهای مورد نیاز برای آموزش را تا حد زیادی کاهش می دهد و موجب ساده تر شدن فرایند آموزش می شود و بر مشکل حداقل های محلی و مشکلات خودرمننگارهای سنتی غلبه می کند. اکنون ما می توانیم به جای یک خودرمننگار، از چند خودرمننگار که بر روی هم انباشته شده اند، استفاده کنیم و بدین ترتیب از خودرمننگار خلوت عمیق استفاده می کنیم. در این مدل، تعداد لایه های مخفی افزوده می شود و هر لایه مخفی از ویژگی های یادگرفته شده از لایه قبل استفاده می کند و خود در ادامه آنها را فشرده تر و ویژگی های جدیدی استخراج می کند. ساختار خودرمننگار خلوت عمیق در شکل ۲ و ۳ دیده می شود. در این مرحله، ماتریس مجاورت و ماتریس مجاورت همسایگی مرتبه دوم را به عنوان داده های آموزشی و ماتریس مجاورت همسایگی مرتبه دوم را به عنوان داده آزمایشی به مدل می دهیم تا در نهایت مدل کاهش بعد یافته گراف اولیه، تولید شود. به طور کلی می توان روش پیشنهادی را که الگوریتم DEEP-MCIM نامگذاری می شود، به صورت زیر خلاصه کرد:

- ۱- پیش پردازش داده ها، شامل به دست آوردن ماتریس مجاورت و همچنین ماتریس مجاورت همسایگان مرتبه دوم
- ۲- کاهش ابعاد گراف اصلی و استخراج توپولوژی آن
- ۳- شناسایی گره های مؤثر با استفاده از الگوریتم MCIM

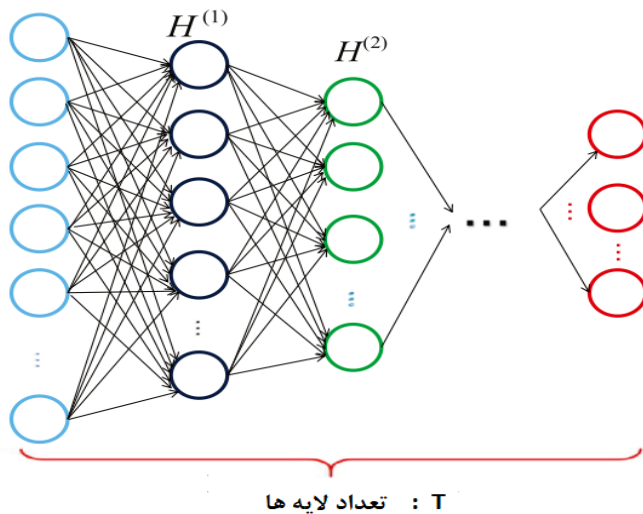
<sup>1</sup> KL divergence



شکل ۲. ساختار یک خودرمزنگار خلوت عمیق.

ورودی : ماتریس های  $X$  و  $Z$

خروجی : ماتریس  $X^{(T)}$



شکل ۳. ساختار یک خودرمزنگار خلوت عمیق با  $T$  لایه مخفی.

## روش انجام آزمایش، نتایج و بحث

برای ارزیابی روش پیشنهادی (DEEP-MCIM)، آن را با روش‌های [۶] MCIM، [۲۳] K-[۲۲] HC<sup>۱</sup>، SHELL و روش رنگ‌آمیزی براساس فاصله با معیار مرکزیت درجه (DCD)<sup>۲</sup> که در [۲۴] آمده است مقایسه کرده‌ایم. همچنین از چهار دیتاست دنیای واقعی به شرح زیر استفاده کرده‌ایم:

- Gnutella dataset
- Pretty Good Privacy (PGP)
- Astro (AST) dataset
- Twitter (TWT) dataset

مشخصات دیتاست‌ها در جدول ۱ قابل مشاهده است.

جدول ۱. مشخصات دیتاست‌های مورد استفاده در آزمایش‌ها.

| نام دیتاست                | تعداد گره‌ها | تعداد یال‌ها | میانگین درجه | میانگین درجه همسایگان مرتبه دوم |
|---------------------------|--------------|--------------|--------------|---------------------------------|
| Gnutella (GLA)            | ۸۱۰۴         | ۲۶۰۱۳        | ۶.۴۱۹        | ۱۰۷.۸۰۴                         |
| Pretty Good Privacy (PGP) | ۱۰۶۸۰        | ۲۴۳۱۶        | ۴.۵۵۴        | ۸۵.۹۷۶                          |
| Astro (AST)               | ۱۸۷۷۱        | ۱۹۸۰۵۰       | ۲۱.۱۰۲       | ۱۳۷۹.۴۷                         |
| Twitter (TWT)             | ۲۳۳۷۰        | ۳۳۱۰۱        | ۲.۸۳۳        | ۱۱۰.۸۹۶                         |

## شرایط آزمایش

برای مقایسه بهتر بین روش پیشنهادی و الگوریتم‌های فوق، از تنظیمات و پارامترها و دیتاست‌های مشابه الگوریتم‌های فوق استفاده شده است.

همان‌گونه که پیش‌تر نیز اشاره شد از مدل پخش SIR برای ارزیابی قابلیت انتشار تأثیر سبب اولیه در روش‌های مختلف استفاده شده است. برای این منظور، گره‌های متعلق به مجموعه سبب اولیه ابتدا در حالت I قرار گرفته‌اند و سایر گره‌ها در حالت S قرار دارند و فرایند گسترش توسط این مدل آغاز شده است. در پایان فرایند، تعداد گره‌هایی که در حالت R قرار دارند، به‌عنوان گسترش نفوذ مجموعه سبب اولیه در نظر گرفته می‌شود. برای ارزیابی دقیق‌تر، فرایند SIR، ۵۰۰ بار اجرا شده است و تعداد متوسط گره‌های R به‌عنوان نفوذ گسترش مجموعه سبب اولیه گزارش شده است. مقدار  $\alpha$  در آزمایش SIR صفر فرض می‌شود و برای تعیین مقدار  $\beta$  از روش UM استفاده شده است که در آن فرض می‌شود که احتمال گسترش در همه یال‌ها با هم برابر است.

به‌منظور نیل به دقت بالاتر، هریک از روش‌ها ۲۰ بار روی مجموعه داده‌های مختلف اجرا می‌گردد و متوسط زمان اجرای آن محاسبه می‌شود. آزمایش‌ها روی سیستمی با ۸G RAM و CPU core i5 نسل ۱۰، ۲ گیگابایت

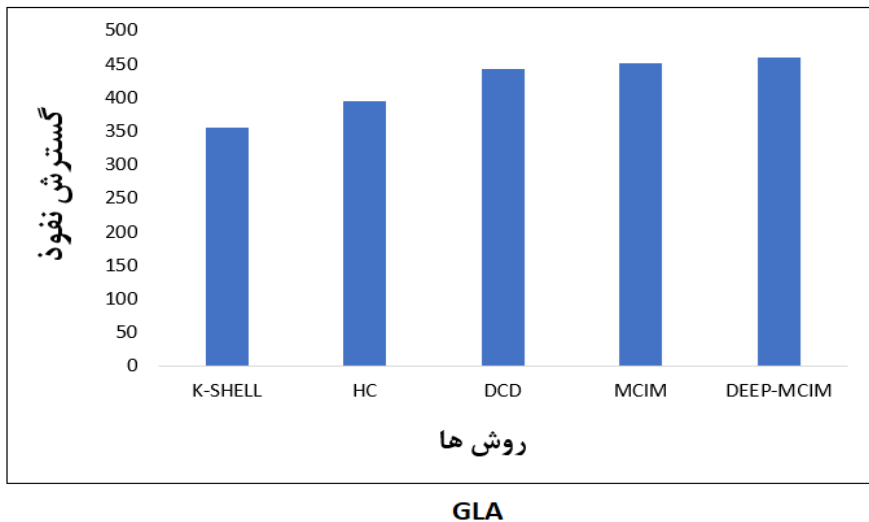
<sup>1</sup> Heuristic clustering

<sup>2</sup> Distance based coloring with degree centrality (dcd)

حافظه گرافیکی و هارد ۲۵۶ SSD و تحت سیستم عامل ویندوز ۱۰ و در محیط اسپایدر با زبان پایتون اجرا شده است.

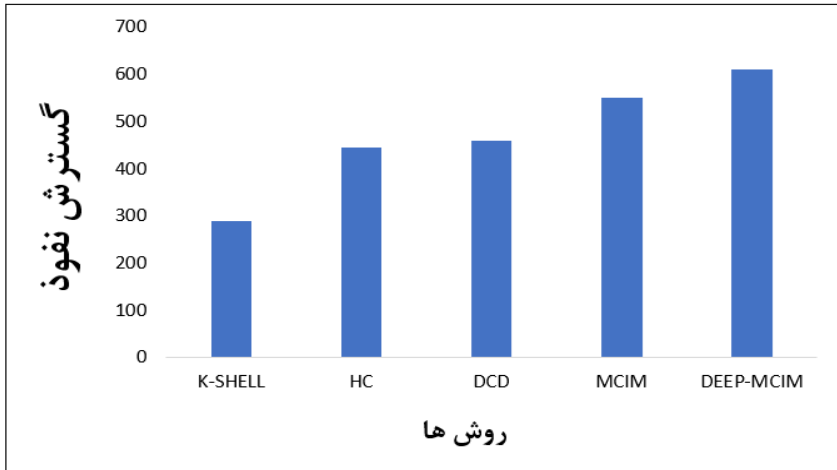
## نتایج و بحث

در اولین آزمایش، میزان گسترش نفوذ روش پیشنهادی (DEEP-MCIM) و سایر روش‌ها بر روی دیتاست‌های مختلف که در بالا مطرح شد، با هم مقایسه شدند. اندازه سید اولیه در این آزمایش برای همه الگوریتم‌ها ثابت و برابر ۴۰ انتخاب شده است. همچنین برای تعیین مقدار  $\beta$  از روش UM استفاده شده و مقدار آن برابر ۰.۰۶ برای همه در نظر گرفته شده است. نتایج این آزمایش در شکل ۴ تا ۷ برای دیتاست‌های مختلف قابل مشاهده است که همان‌گونه که مشخص است میزان گسترش نفوذ<sup>۱</sup> الگوریتم پیشنهادی نسبت به سایر روش‌ها از وضعیت بهتری برخوردار است. همان‌گونه که می‌بینید در مجموعه داده توپیتر (TWT) که تعداد گره‌های بیشتری وجود دارد و گراف بزرگ‌تری نسبت به سایر مجموعه داده‌ها داریم، اختلاف عملکرد الگوریتم پیشنهادی نسبت به سایر رقبا بیشتر است در حالی که در مجموعه داده‌های کوچک‌تر، این تفاوت کمتر است. از طرفی روش K-SHELL عملکرد ضعیف‌تری نسبت به بقیه از خود نشان داده است.



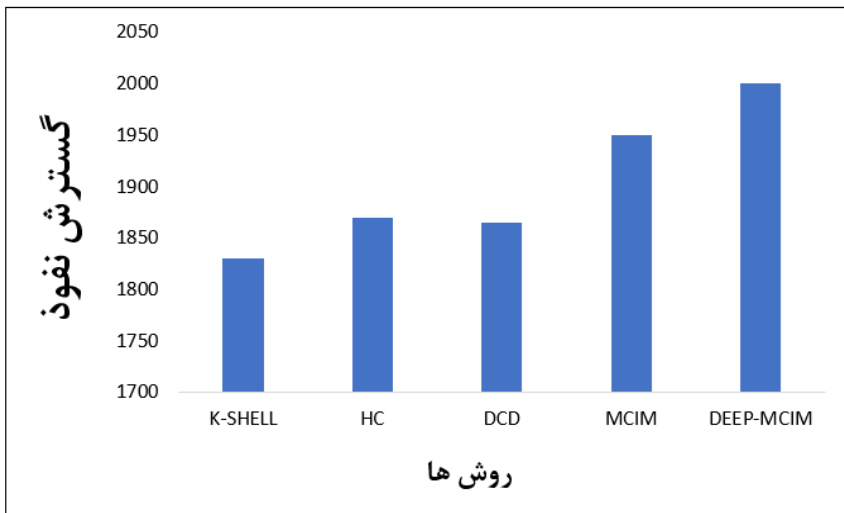
شکل ۴. مقایسه میزان گسترش نفوذ روش‌های مختلف ( $K=40$ ).

<sup>1</sup> Influence spread



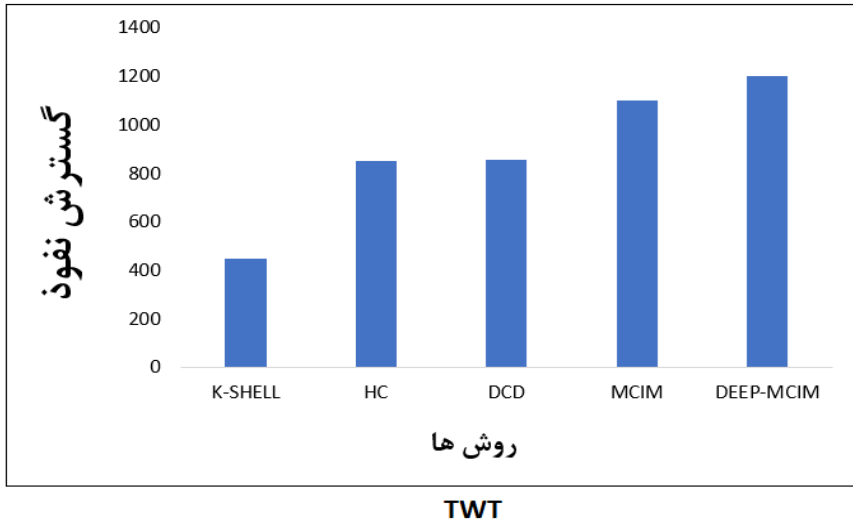
**PGP**

شکل ۵. مقایسه میزان گسترش نفوذ روش‌های مختلف ( $K=40$ ).



**AST**

شکل ۶. مقایسه میزان گسترش نفوذ روش‌های مختلف ( $K=40$ ).



TWT

شکل ۷. مقایسه میزان گسترش نفوذ روش‌های مختلف ( $K=40$ ).

در آزمایش دوم روش پیشنهادی با روش‌های فوق از نظر زمان اجرا مقایسه شده است. نتایج در جدول ۲ قابل مشاهده است.

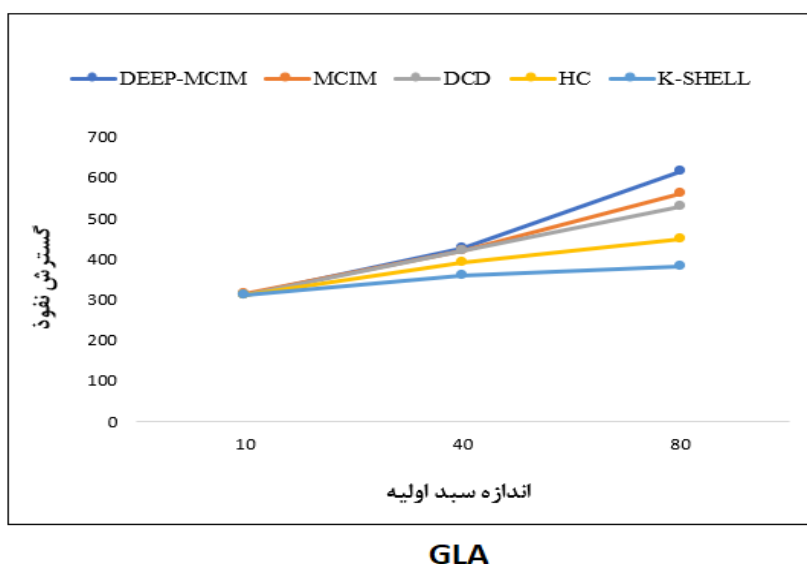
در این آزمایش، برای ارزیابی کارایی روش پیشنهادی، DEEP-MCIM و سایر روش‌ها ۲۰ بار در شبکه‌های مختلف آزمایش شده‌اند و میانگین زمان اجرای هر روش بر روی مجموعه داده‌های واقعی به سمت عدد کوچک‌تر گرد شده و نتایج در جدول ۲ نشان داده شده است.

نتایج در جدول ۲ نشان می‌دهد که بیشترین زمان اجرا متعلق به روش HC است و هرچه اندازه مجموعه داده بزرگ‌تر می‌شود وضعیت به مراتب بدتر می‌شود؛ از این رو استفاده از این روش در مجموعه داده‌های بزرگ عملاً به صرفه نیست. علت آن است که در روش HC گره‌های گراف باید خوشه‌ای باشند و این موضوع منجر به افزایش پیچیدگی زمانی می‌شود. همچنین روش DCD دارای پیچیدگی زمانی کمتری نسبت به MCIM و DEEP-MCIM در مجموعه داده‌های کوچک است، در حالی که با افزایش اندازه شبکه پیچیدگی محاسباتی DCD افزایش می‌یابد. همان‌گونه که در جدول ۲ مشهود است با افزایش اندازه مجموعه داده، کارایی روش پیشنهادی از نظر زمان اجرا نسبت به سایر روش‌ها به خوبی قابل ملاحظه است به ویژه در مجموعه داده‌های AST و TWT. اگر چه تعداد گره‌ها در TWT بیشتر است اما میانگین درجه گره‌ها کمتر می‌باشد و بنابراین گراف خلوت‌تری نسبت به AST خواهیم داشت؛ از این رو روش پیشنهادی با افزایش اندازه گراف و تعداد گره‌ها، کارایی بسیار بهتری نسبت به سایر روش‌ها در این آزمایش دارد در حالی که در مجموعه داده‌های کوچک‌تر، این تفاوت کمتر است. روش K-SHELL دارای پیچیدگی محاسباتی کمتری نسبت به DEEP-MCIM است و از نظر زمان اجرا وضعیت بهتری دارد.

جدول ۲. مقایسه زمان اجرا (برحسب میلی ثانیه) روش پیشنهادی و سایر روش‌ها در مجموعه داده‌های دنیای واقعی.

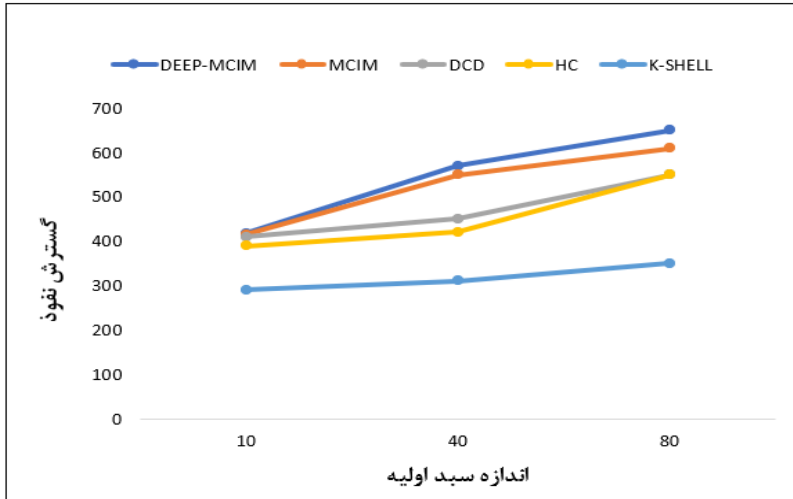
| نام دیتاست                | K-SHELL | HC    | DCD  | MCIM | DEEP-MCIM |
|---------------------------|---------|-------|------|------|-----------|
| Gnutella (GLA)            | ۴۱      | ۹۱۲۵  | ۱۴۹  | ۱۷۱  | ۱۰۲       |
| Pretty Good Privacy (PGP) | ۱۲      | ۷۹۳۶  | ۲۳۶  | ۲۶۲  | ۲۱۰       |
| Astro (AST)               | ۹۸      | ۸۵۴۳۲ | ۹۲۱۶ | ۹۳۷۸ | ۵۹۶۲      |
| Twitter (TWT)             | ۳۶      | ۳۶۲۱۳ | ۸۴۵۶ | ۷۳۶۳ | ۴۲۱۵      |

در آزمایش سوم، تأثیر اندازه سبد اولیه<sup>۱</sup> (K) بر عملکرد روش‌های مختلف بررسی شده است. در این آزمایش، اندازه سبد اولیه برابر ۱۰، ۴۰ و ۸۰ تنظیم شده و در هر مجموعه داده، عملکرد روش پیشنهادی با سایر روش‌ها ارزیابی شده است و نتایج به دست آمده در شکل‌های ۸ تا ۱۱ نمایش داده شده است.



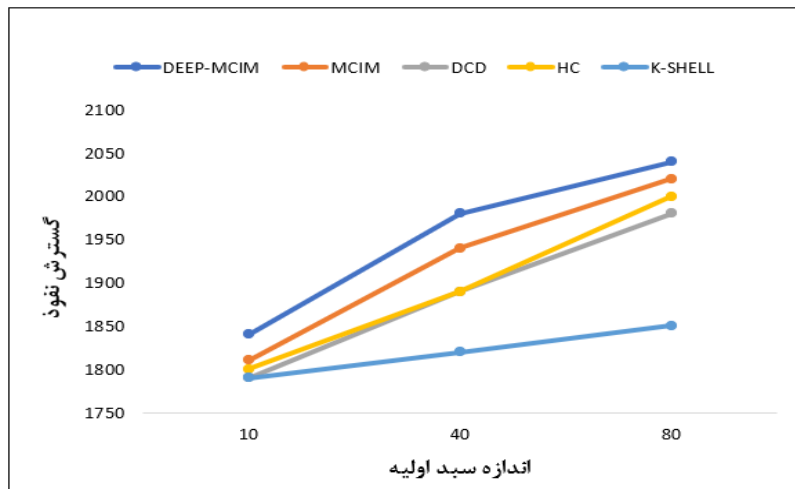
شکل ۸. مقایسه تأثیر اندازه سبد اولیه بر میزان گسترش نفوذ در روش‌های مختلف ( $K=10, 40, 80$ ).

<sup>1</sup> Seed set



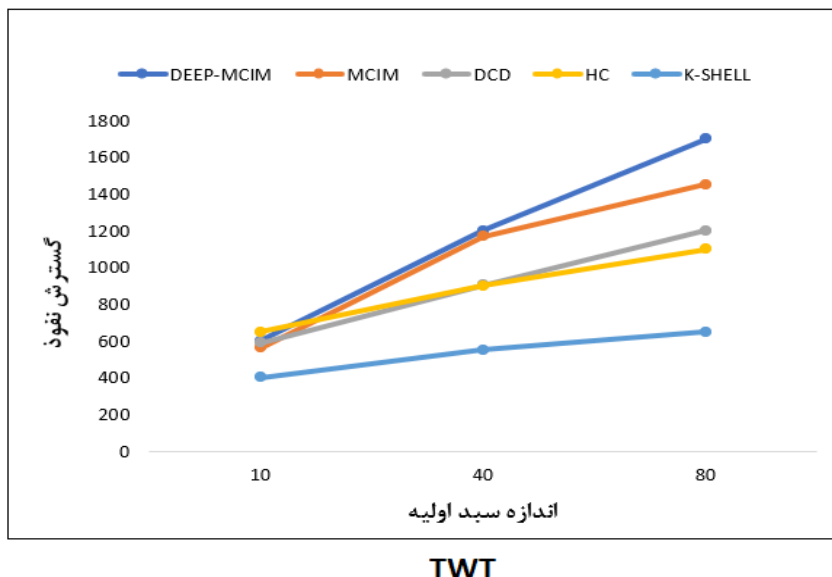
**PGP**

شکل ۹. مقایسه تأثیر اندازه سید اولیه بر میزان گسترش نفوذ در روش‌های مختلف ( $K=10, 40, 80$ ).



**AST**

شکل ۱۰. مقایسه تأثیر اندازه سید اولیه بر میزان گسترش نفوذ در روش‌های مختلف ( $K=10, 40, 80$ ).



شکل ۱۱. مقایسه تأثیر اندازه سبد اولیه بر میزان گسترش نفوذ در روش‌های مختلف ( $K=10, 40, 80$ ).

همان‌گونه که در شکل ۸ تا ۱۱ مشخص است روش پیشنهادی توانسته است عملکرد بهتری نسبت به سایر روش‌ها در این آزمایش، از خود نشان دهد. برای مقادیر پایین  $K$ ، در مجموعه داده‌های کوچک مانند GLA و PGP میزان گسترش نفوذ روش‌های مختلف، مشابه هم است و به جز روش K-SHELL که عملکرد بسیار ضعیف‌تری نسبت به بقیه دارد، سایر روش‌ها تقریباً به‌طور یکسانی عمل کرده‌اند. اما با افزایش مقدار  $K$ ، روش‌های MCIM و DEEP-MCIM عملکرد بهتری از خود نشان داده‌اند؛ زیرا این روش‌ها علاوه بر معیارهای مرکزیت گره‌ها، عامل هم‌پوشانی با عناصر سبد اولیه را نیز به‌عنوان یک عامل منفی در نظر گرفته‌اند و با حذف گره‌های هم‌پوشان در سبد، با هوشمندی بیشتری نسبت به انتخاب گره‌های پرنفوذ، عمل می‌کنند. هرچه اندازه سبد ( $K$ ) بزرگ‌تر شود و اندازه مجموعه داده افزایش یابد، این بهبود عملکرد، واضح‌تر می‌شود؛ به‌خصوص در مجموعه داده‌های AST و TWT زمانی که تعداد گره‌ها و یال‌ها و میانگین درجه گره‌ها زیاد باشد، روش پیشنهادی با توجه به کاهش ابعاد گراف عملکرد بسیار خوبی دارد؛ به‌ویژه در TWT که نسبت به AST خلوت‌تر است اما تعداد گره‌های بیشتری دارد.

### نتیجه‌گیری

در این مقاله، روش جدیدی برای استخراج ویژگی‌های مهم گراف شبکه‌های اجتماعی ارائه گردید و با استفاده از خودرمن‌نگارهای عمیق، توپولوژی شبکه استخراج شد و عملیات کاهش ابعاد گراف انجام گردید. سپس با در نظر گرفتن همسایگان مستقیم و همسایگان درجه دوم و با استفاده از ماتریس مجاورت گراف و در ادامه با بهره‌گیری از روش MCIM گره‌های پرنفوذ شناسایی شد. در ادامه روش پیشنهادی بر مجموعه داده‌های دنیای واقعی، با چهار روش دیگر (MCIM, DCD, HC, K-SHELL) مقایسه و ارزیابی شد. نتایج حاصل از شبیه‌سازی، مؤید عملکرد بهتر روش

پیشنهادی نسبت به سایر رقبا از نظر زمان اجرا و میزان گسترش نفوذ بود؛ به‌ویژه زمانی که اندازه مجموعه داده بزرگ باشد و تعداد گره‌ها و یال‌های زیادی داشته باشیم و از طرفی اندازه سید اولیه بالا باشد، کارایی روش پیشنهادی به‌مراتب بهتر از سایر روش‌هاست در حالی که برای گراف‌های کوچک و با اندازه سید اولیه کوچک، این تفاوت عملکرد چندان محسوس نیست و این بدان دلیل است که معمولاً در همه این روش‌ها عناصر اولیه سید براساس معیارهای مرکزیت مانند درجه، مرتب و انتخاب می‌شوند و از طرفی در گراف‌های کوچک، عملیات کاهش ابعاد، علاوه بر سرباری که ایجاد می‌کند، تفاوت چندان محسوسی در نتیجه نهایی ایجاد نمی‌کند.

## References

- [1] Centola, D. (2010). The spread of behavior in an online social network experiment. *science*, 329(5996), 1194-1197. <https://doi.org/10.1126/science.1185231>
- [2] Chen, W., Wang, C., & Wang, Y. (2010, July 25-28). *Scalable influence maximization for prevalent viral marketing in large-scale social networks*. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, DC, USA. <https://doi.org/10.1145/1835804.1835934>
- [3] Bakshy, E., Hofman, J. M., Mason, W. A., & Watts, D. J. (2011, February 9-12). *Everyone's an influencer: quantifying influence on twitter*. Proceedings of the fourth ACM international conference on Web search and data mining, Hong Kong, China. <https://doi.org/10.1145/1935826.1935845>
- [4] Banerjee, S., Jenamani, M., & Pratihar, D. K. (2020). A survey on influence maximization in a social network. *Knowledge and Information Systems*, 62(9), 3417-3455. <https://doi.org/10.1007/s10115-020-01461-4>
- [5] Li, Y., Fan, J., Wang, Y., & Tan, K. L. (2018). Influence Maximization on Social Graphs: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10), 1852-1872. <https://doi.org/10.1109/TKDE.2018.2807843>
- [6] Zareie, A., Sheikhhahmadi, A., & Khamforoosh, K. (2018). Influence maximization in social networks based on TOPSIS. *Expert Systems with Applications*, 108, 96-107. <https://doi.org/10.1016/j.eswa.2018.05.001>
- [7] Chandran, J., & Viswanatham, V. M. (2021, February 19-20). *Evaluating the Effectiveness of Community Detection Algorithms for Influence Maximization in Social Networks*. 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies Bhilai, India <https://doi.org/10.1109/ICAECT49130.2021.9392387>
- [8] Kumar, S., Singhla, L., Jindal, K., Grover, K., & Panda, B. S. (2021). IM-ELPR: Influence maximization in social networks using label propagation based community structure. *Applied Intelligence*, 51(11), 7647-7665. <https://doi.org/10.1007/s10489-021-02266-w>
- [9] Wang, Z., Sun, C., Xi, J., & Li, X. (2021). Influence maximization in social graphs based on community structure and node coverage gain. *Future Generation Computer Systems*, 118, 327-338. <https://doi.org/10.1016/j.future.2021.01.025>
- [10] Statista. (2022). *Leading countries based on Facebook audience size as of January 2022*. Statista. <https://www.statista.com/statistics/268136/top-15-countries-based-on-number-of-facebook-users/>
- [11] Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4), 1118-1123. <https://doi.org/10.1073/pnas.0706851105>

- [12] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>
- [13] Zhang, Y., Lyu, T., & Zhang, Y. (2018, February 2-7). *COSINE: Community-Preserving Social Network Embedding From Information Diffusion Cascades*. The Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana USA. <https://ojs.aaai.org/index.php/AAAI/article/view/11856>
- [14] Wikipedia. (2022, February 28). *Complex network*. the Creative Commons Attribution-ShareAlike License 3.0. [https://en.wikipedia.org/wiki/Complex\\_network](https://en.wikipedia.org/wiki/Complex_network)
- [15] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70-90. <https://doi.org/10.1016/j.compag.2018.02.016>
- [16] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., & ... (2016, June 19-24). *Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin*. Proceedings of The 33rd International Conference on Machine Learning, New York, United States. <https://proceedings.mlr.press/v48/amodei16.html>
- [17] Yang, H., Li, S., Wu, X., Lu, H., & Han, W. (2019). A Novel Solutions for Malicious Code Detection and Family Clustering Based on Machine Learning. *IEEE Access*, 7, 148853-148860. <https://doi.org/10.1109/ACCESS.2019.2946482>
- [18] Zhao, D., Gao, B., Wang, Y., Wang, L., & Wang, Z. (2018). Optimal Dismantling of Interdependent Networks Based on Inverse Explosive Percolation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(7), 953-957. <https://doi.org/10.1109/TCSII.2018.2793257>
- [19] Collobert, R., & Weston, J. (2008, July 5-9). *A unified architecture for natural language processing: deep neural networks with multitask learning*. Proceedings of the 25th international conference on Machine learning, Helsinki, Finland. <https://doi.org/10.1145/1390156.1390177>
- [20] Li, S., Jiang, L., Wu, X., Han, W., Zhao, D., & Wang, Z. (2021). A weighted network community detection algorithm based on deep learning. *Applied Mathematics and Computation*, 401(7), 126012. <https://doi.org/10.1016/j.amc.2021.126012>
- [21] Keikha, M. M., Rahgozar, M., Asadpour, M., & Abdollahi, M. F. (2020). Influence maximization across heterogeneous interconnected networks based on deep learning. *Expert Systems with Applications*, 140(10), 112905. <https://doi.org/10.1016/j.eswa.2019.112905>
- [22] Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., & Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature Physics*, 6(11), 888-893. <https://doi.org/10.1038/nphys1746>
- [23] Bao, Z-K., Liu, J-G., & Zhang, H-F. (2017). Identifying multiple influential spreaders by a heuristic clustering algorithm. *Physics Letters A*, 381(11), 976-983. <https://doi.org/10.1016/j.physleta.2017.01.043>
- [24] Guo, L., Lin, J-H., Guo, Q., & Liu, J-G. (2016). Identifying multiple influential spreaders in term of the distance-based coloring. *Physics Letters A*, 380(7-8), 837-842. <https://doi.org/10.1016/j.physleta.2015.12.031>