IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 June 2025, accepted 16 July 2025, date of publication 21 July 2025, date of current version 25 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3590835

RESEARCH ARTICLE

Dynamic Adaptive Parametric Social Network Analysis Using Reinforcement Learning: A Case Study in Topic-Aware Influence Maximization

MOHAMMAD HOSSEIN AHMADIKIA AND MEHDY ROAYAEI
Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran 14115-111, Iran
Corresponding author: Mehdy Roayaei (mroayaei@modares.ac.ir)

ABSTRACT Current network analysis algorithms often rely on search methods or centrality measures but face challenges such as 1) The solution space is large, resulting in high computational complexity. 2) Algorithms may be instance-dependent, relying considerably on network structure and characteristics, which may result in varying performance across different networks. 3) Most existing centrality measures are inherently static, which fail to capture the dynamic nature inherent in network analysis problems. To address these issues, this paper introduces a dynamic adaptive parametric (DAP) approach using reinforcement learning. As a case study, the method has been applied to the topic-aware influence maximization (TIM) problem, where the objective is to identify k influential nodes that maximize influence spread under a given topic vector and diffusion model. The paper introduces two dynamic centrality measures that capture the evolving importance of nodes during topic propagation in the network. To avoid instance-dependence, an adaptive reinforcement learning technique is used to adjust the significance of each measure based on the current network structure, tailoring solutions to the specific network. The parametric approach further reduces the search space by transforming TIM into a parametric optimization task, where the goal is to determine the optimal importance of each centrality measure. The proposed algorithm is evaluated on both real-world and synthetic networks. Experimental results show that the method outperforms conventional centrality-based greedy algorithms and other existing approaches in terms of solution quality, running time, and scalability. Also, as a part of our research, we propose a topic-aware benchmark dataset by augmenting the Deezer music-based social network with labeled nodes and edges, providing a valuable resource for evaluating future research.

INDEX TERMS Topic-aware influence maximization, reinforcement learning, parametric optimization, adaptive analysis, dynamic centrality measure.

I. INTRODUCTION

Influence Maximization (IM) involves selecting a small group of individuals in a network to maximize the spread of influence, with key applications in viral marketing. The problem is NP-hard, and ongoing research focuses on reducing its computational complexity while improving practical efficiency [1]. The independent cascade model and the linear threshold model [2] are two common diffusion models in IM. The independent cascade model is a probabilistic model where influence spreads through a network based on a series of independent events, while the linear threshold model is a deterministic model where nodes in a network have a threshold of activation and become influenced once the cumulative influence from their neighbors exceeds this threshold.

In most IM literature, topics are treated as identical, ignoring the varying levels of user interest and authority across different topics [3]. Despite the clear correlation between users' authority, expertise, trust, and their influence

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Fiumara.

Q=(<0.2,0.8, 0>,2)

Edge	Topic 1	Topic 2	Topic 3
(1,2)	0.7	0.3	0
(1,3)	0.4	0.8	0.3
(1,5)	0.6	0.8	0.3
(2,3)	0.6	0.7	0.5
(2,6)	0.5	0.1	0.3
(2,7)	0.8	0	0.4
(3,4)	0.1	0.4	0.3
(4,5)	0	0.5	0.6
(4,8)	0	0	0.7
(6,7)	0.6	0	0.2
(7,8)	0.1	0	0.8

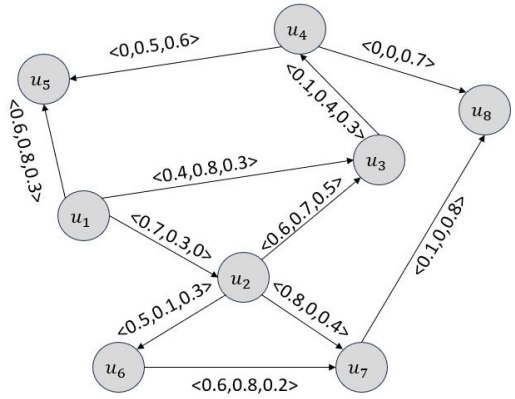

FIGURE 1. An instance of a topic-aware problem [5].

on specific topics, research on social influence has notably neglected this dimension until a decade later [4]. Traditional IM problems assume that the probability of propagating a content in a network is independent of the content itself. However, in real-world scenarios, propagation probability can vary depending on the topic (class and type of content being propagated). This is known as topic-aware influence maximization problem (TIM) in social networks [5].

In general, topic-aware problems in social networks can be formulated as Eq. (1), which specifies the relevance of each topic for query Q (item) to be propagated through network, along with the number of initial influencers. For example, in Fig. 1, the relevance of query Q to three different topics is 20%, 80%, and 0%, respectively, with two initial influencers. The influence of each user u on user v is denoted as a labeled three-dimensional vector on the directed edge (u, v), indicating the extent of u's influence on v across each topic.

$$Q = (\text{topic=importance}, [\text{number of influencers}]) \quad (1)$$

TIM is formally defined on a directed graph, where edges represent topic-specific user-to-user social influence strengths, and budget k is given. The goal is to identify a set of k nodes to maximize the spread of influence for a query (given as a vector denoting the distribution over topics) [3]. TIM has wide applications including viral marketing and targeted advertising [6], and feed ranking [7].

Various algorithms have been developed for IM including greedy algorithms [8], metaheuristics [9], and approximation algorithms [10]. On one hand, metaheuristics search the solution space for optimal solutions, which can be time-consuming, especially for large networks. On the other hand, other methods try to find optimal solution by using measures that indicate the importance of nodes, including centrality metrics [11]. Centrality is a quantitative measure aimed at revealing the importance of nodes. Since the concept of importance can have broad and varied definitions, there are multiple centrality measures as well (including degree, closeness, betweenness, eigenvector, etc.). The problem associated with these methods is that they are instance-dependent approaches which considerably depend on network structure and topic being propagated. Also, most existing centrality measures are inherently static which do not consider the dynamic nature of network analysis problems.

This paper proposes a dynamic adaptive parametric approach using reinforcement learning (RL) to address these issues. RL-based approaches are independent of the environment (in this case, the network instance) and adapt to the environment's characteristics during execution. First, two novel dynamic centrality measures are introduced to assess the importance of nodes by monitoring the propagation of topics within the network. Unlike *static measures*, which remain unchanged during propagation, *dynamic measures* adjust over time, enhancing solution accuracy and relevance in real-world dynamic networks. Second, to avoid dependency on specific network instances, an adaptive RL-based is employed, which updates the significance of each centrality measure during algorithm execution based on the current network structure. This enables the generation of network-specific solutions. Third, the parametric approach narrows the search space by converting TIM into a parametric optimization problem, where the goal is to determine the significance of each centrality measure (both static and dynamic) in the optimal solution. The contributions of the paper can be stated as follows:

- The problem of identifying topic-aware influential nodes using static and dynamic centrality measures is modeled as an RL problem.
- Two novel dynamic centrality measures are introduced to evaluate node importance based on the current state of algorithm execution.
- A parametric approach reduces the solution space by determining the significance of centrality measures in optimal solutions rather than directly identifying influential nodes.
- An RL-based approach dynamically adjusts the significance of centrality measures during the algorithm execution, tailored to the network structure.

- A topic-aware benchmark dataset is proposed by augmenting the Deezer music-based social network with labeled nodes and edges.

The paper is organized as follows. Section II reviews related work, Section III presents the proposed method, Section IV evaluates the proposed method against conventional approaches, and Section V concludes with suggestions for further.

II. RELATED WORK

IM has been extensively studied using four main algorithmic categories: greedy algorithms, such as those by Kempe et al. [12] which are effective but computationally intensive due to the need for repeated influence spread calculations; metaheuristics, like genetic algorithms [13] and simulated annealing [14] enabling exploration beyond local optima for near-optimal solutions in larger networks; approximation algorithms, which utilize sub-modularity to provide computationally feasible solutions with performance guarantees [10] and machine learning-based approaches, which integrate techniques like deep learning to enhance accuracy and efficiency in real-time applications for dynamic social networks [15].

Recent research on using RL in IM has highlighted innovative methods to optimize influence spread in social networks. Li et al. [16] proposed a deep RL (DRL)-based approach that formulates IM as an RL task, with the network configuration as the state and node selection as the actions. Also, Chen et al. [17] addressed challenges in complex contagion, where influence probability increases with the number of influenced neighbors, by developing an RL algorithm to overcome reward sparsity, achieving state-of-the-art results on real-world networks.

Preliminary studies in TIM have introduced new diffusion models to capture dynamic probabilities across different topics, improving the understanding of information propagation in topic-aware settings [4], [18]. These models address real-world complexities, where individuals have varying probabilities of being activated in different topics, and the influence between individuals changes with topics.

In comparison to the IM problem, much less research has been conducted on the TIM problem. Li et al. [6] addressed the problem of keyword-based targeted IM, focusing on identifying a seed set that maximizes the expected influence over users relevant to a specific advertisement. They proposed a sampling technique based on weighted reverse influence set, achieving an approximation ratio of $1 - 1/e - \epsilon$. Nguyen et al. [19] addressed the challenge of targeting influential users in TIM under budget constraints, introducing algorithms that balance influence spread and marketing costs in large-scale networks.

Tian et al. [18] presented a novel framework for TIM using a learning-based approach, called DIEM. They introduced a meta-learning framework that captures the topic-specific nature of information propagation in networks, aiming to maximize influence under a new diffusion model. They also proposed a deep influence evaluation model to assess user influence under different circumstances, enabling the efficient construction of solutions based on these evaluations.

Compared to our work, DIEM has the following differences. DIEM formulates TIM problem for RL by having the agent learn a function to directly estimate the influence of adding a candidate node to the current partial solution, where the RL state represents the partial solution itself, and actions are selecting nodes. In contrast, DAP-TIM models TIM as a parametric optimization problem, where the RL agent learns the optimal significance weights for a predefined set of centrality measures, and the RL state is a vector representing these weights. Regarding dynamic aspects and adaptability, DIEM incorporates dynamic probabilities into node features via embeddings (Diffusion2Vec) and aims for generalization to different graph instances after training, while DAP-TIM explicitly introduces novel dynamic centrality measures that change during solution generation and uses RL to adaptively adjust the weights of all centrality measures based on the current network structure during execution. Finally, DIEM utilizes DDQN with prioritized experience replay, while DAP-TIM employs the A2C algorithm.

Huang et al. [20] introduced the Comparative Independent Cascade model in TIM, which captures both competitive and complementary interactions among users. Wang et al. [21] tackled IM in online social networks from an opinion-aware perspective, integrating opinion dynamics into the IM framework. Their approach focuses on identifying seed users that maximize influence spread while considering the opinions and sentiments expressed within the network. Teng and Wang [22] focuses on identifying influential nodes within specific topics, employing preprocessing techniques to enhance the efficiency of influence maximization across diverse topic-based communities.

Our paper advances TIM by modeling the identification of influential nodes as an RL problem, diverging from previous studies that primarily relied on static centrality measures. Two dynamic centrality measures are introduced that assess node importance based on the algorithm's current state, enabling real-time adaptability. Additionally, we employ a parametric approach to focus on the significance of various centrality measures, allowing the framework to adjust according to the network's structure.

III. PROPOSED METHOD

In this section, we introduce DAP-TIM, an RL-based algorithm that uses six centrality measures to calculate the node importance (score). The algorithm starts by initializing a vector of size six, representing the significance of each measure in determining the score of each node. These measures are:

- **Degree Centrality**: Measures the number of direct connections a node has.

- **Closeness Centrality**: Reflects how quickly a node can access all other nodes in the network by calculating the average shortest path from that node to all others.
- **Betweenness Centrality**: Quantifies a node's role as a bridge in the network by measuring the number of shortest paths that pass through it.
- **Eigenvector Centrality**: Assesses a node's influence based on the importance of its connections, assigning higher scores to nodes linked to other well-connected nodes.
- **Reachability Factor (RF)**: An approximate measure calculated by summing the probability of influence spread from each node to its t-hop neighbors ($1 \leq t \leq 6$).
- **Crowded Factor (CF)**: An approximate measure indicating the number of nodes within one and two hops in the neighborhood of a given node that have already been selected in the current partial solution.

A key module within this algorithm, known as the Solution Generator (SG), takes this vector as input, and computes the total scores for all nodes based on their centrality scores and the significance of each measures. It then provides the top k nodes as an output solution. Two of these six measures (RF and CF) are new and introduced in this paper, and, due to their dynamic nature, SG selects nodes one by one and adds them to the current partial solution. By altering the significance values of centrality measures, the output of SG will also change. By calculating the influence levels of the outputs of SG and adjusting the significance values of centrality measures using the RL algorithm, an optimized vector is achieved to generate the best solutions by SG. The details of DAP-TIM are provided below.

A. TOPIC-AWARE DIFFUSION MODELING

Each node v is represented as a binary vector $U_{interest}(v)$, where each element (0 or 1) indicates the individual's interest in the topics. The vector is defined as in Eq. (2), where Z denotes the total number of topics.

$$U_{interest}(v) = [v_{q_1}, v_{q_2}, \cdots, v_{qZ}], v_{q_i} \in \{0, 1\}, \ 1 \leq i \leq Z \tag{2}$$

The weight w_{uv} represents the degree of connection between nodes u and v. The probability of influence propagation from u to v on query q is a function of the topic vector during propagation (Topic_q in Eq. (3)), w_{uv}, and the interest vector of u. The query being propagated is a vector of values between 0 and 1, indicating the relevance of the content being propagated to different topics, and is given as a part of the problem input. The diffusion probability for each edge is modeled as in Eq. (4), indicating the propagation probability depends on the direction of propagation. Users with interest vectors more similar to the topic vector disseminate the content more effectively.

$$Topic_q = [q_1, q_2, \cdots, q_Z], q_i \in \mathbb{R}, 0 \leq q_i \leq 1 \ (1 \leq i \leq Z) \tag{3}$$

$$p(u, v, q) = w_{uv} \times \max_{i=1}^{Z}(Topic_q[i] \times U_{interest}(u)[i]) \tag{4}$$

B. PROBLEM FORMULATION

In most RL-based search algorithms for combinatorial optimization problems like TIM, states represent solutions, and actions correspond to altering these solutions, often resulting in large state and action spaces that increase computational complexity. To address this, we formulate the problem to enable the model to learn the characteristics of influential nodes rather than the nodes themselves, significantly reducing the search space. Instead of identifying influential nodes from the entire network, we focus on determining the significance weights of a few centrality measures for selecting influential nodes in the final solution. TIM can thus be modeled as an RL problem as follows:

- **State space** $\mathcal{S}$: The state $s = [f_1, \cdots, f_6]$ comprises the weights of six centrality measures (Degree, Closeness, Betweenness, Eigenvector, Reachability Factor, Crowded Factor), where each weight f_i (ranging from 0 to 100) represents the significance of the corresponding centrality measure in selecting influential nodes. These weights are normalized to ensure $\Sigma_{i=1}^{6} f_i = 100$, and they evolve during the learning process to capture the optimal balance of centrality measures needed for effective influence maximization. These weights contribute directly to selecting influential nodes by guiding the ranking process in the proposed algorithm.
- **Action space** $\mathcal{A}$: Consists of actions defined as an ordered pair of integers (a, b), where a ($1 \leq a \leq 6$) specifies the centrality measure we are trying to improve its weight, and b ($0 \leq b \leq 3$) indicates the extent of increased applied to that weight. Other weights decrease proportionally to maintain a fixed total.
- **Transition function** $\mathcal{T}$: Describes transition from state s to s' based on the action (a, b). For state $s = [f_1, \cdots, f_6]$, $\mathcal{T}$ is defined as in Eq. (5) where $\delta = \max(6 * b - 9, 0)$, determined empirically.

$$f_i' = \begin{cases} \min(f_i + \delta, 100) & i = a \\ \max(f_i - \frac{\delta}{5}, 0) & i \neq a \end{cases} \tag{5}$$

- **Reward function** $\mathcal{R}$: Evaluates the quality of an action by measuring the difference in influence propagation between two consecutive steps. At step i, after applying an action and modifying the weights of the centrality measures, SG generates a k-set of nodes. The reward is calculated as the difference in the number of influenced nodes between steps $i-1$ and i, reflecting the effectiveness of the action in enhancing influence spread.

C. SOLUTION GENERATOR

As can be inferred from Fig. 2, static centrality measures (degree, closeness, betweenness, and eigenvector) are calculated once in the start of the algorithm. However, to capture

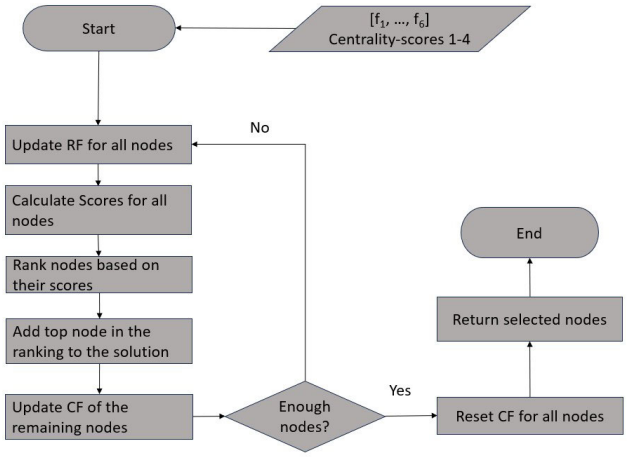

FIGURE 2. Flowchart of SG.

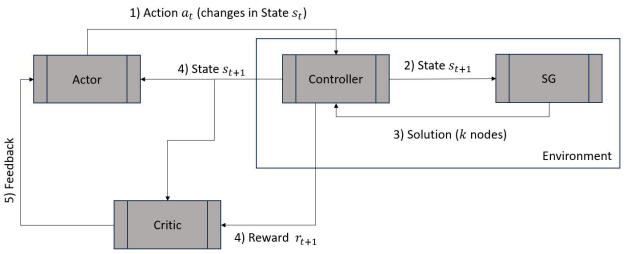

FIGURE 3. Framework of DAP-TIM.

the impact of selecting each node on the propagation, and to dynamically adapt the solution based on this impact, two other measures including CF and RF are updating during generating solutions by SG.

Solution Generator (SG) is introduced as an important part of DAP-TIM to simplify solution space. Given the current state, and the centrality scores of dynamic and static measures for all nodes, SG generates a solution (seed set). SG can be tuned by changing weights of centrality measures $[f_1, f_2, f_3, f_4, f_5, f_6]$.

Alongside four mentioned well-known centrality measures, two dynamic measures have been introduced in this paper, capturing the dynamic evolution of the search algorithm. Reachability Factor (RF) approximates the influence spread from a node to its t-hop neighbors ($1 \leq t \leq 6$. Based on the principle that the average path length in real-world social networks is six, RF provides a network-wide estimate of a node's influence extent. For each node u, $RF_t(u)$ calculates the probability of the spread of influence from u to t-hop neighbors, which is defined as in Eq. (6):

$$RF_t(u) = \sum_{v \in \text{neighbors}(u)} RF_{t-1}(v) \cdot p(u, v, q), \quad 1 \leq t \leq 6 \tag{6}$$

where $RF_{0(u)}$ is considered 1 for all nodes. This equation is calculated six times for each node; Thus, $\Sigma_{t=1}^{6} RF_t(u)$ provides an approximation of the potential number of nodes

Algorithm 1 : A2C-TIM Algorithm

```
for episodes = 0, 1, 2, ..., T do
    Get state s and calculate π(s) to get action (a, b)
    if the episode is not finished then
        Calculate new state s' using Eq. (5)
        Generate seed set using SG
        Calculate reward (influence_{s'} − influence_s)
        Estimate the Q value by Critics
        Update actor's policy π(s) using Critic's feedback
        Update the Critic to reduce the loss
        s = s'
    end if
end for
```

TABLE 1. Dataset characteristics.

Data Set	#Nodes	#Edges	#Topics
Deezer_HR	54573	498202	84
Deezer_HU	47538	222887	84
Deezer_RO	41773	125826	84
Random_Small	20000	59991	84
Random_Medium	54573	491076	84
Random_Large	90000	1079856	84

influenced by u in the network, assuming only node u is active.

Crowded Factor (CF) for node u, defined in Eq. (7), is also an approximate measure indicating the number of nodes within one-hop and two-hop neighborhoods of a given node u that have already been selected in the partial solution in SG.

$$CF(u) = \sum_{v \in \text{selected-neighbors}(u \mid d=1)} -1 + \sum_{v \in \text{selected-neighbors}(u \mid d=2)} -\frac{1}{2} \tag{7}$$

where selected-neighbors$(u|d = i)$ represents the neighboring nodes of u at a distance i, which are already in the solution. Thus, CF indicates a node's proximity to selected nodes, with higher proximity reducing its contribution to influence improvement due to redundancy.

The calculated values for six centrality measures are normalized to [0, 1]. The score of a node u in the ranking process is calculated using Eq. (8):

$$\text{score}(u) = \sum_{i=1}^{6} f_i \times \text{centrality-score}_i \tag{8}$$

Thus, SG ranks network nodes by using the weighted average based on its six input parameters and return the top k nodes as a recommended solution. The flowchart in Fig. 2 illustrates how SG convert an input to a solution.

D. RL ALGORITHM

By transforming the problem into an RL problem in a six-dimensional space, conventional RL methods can be employed to solve it. In this paper, the Advantage

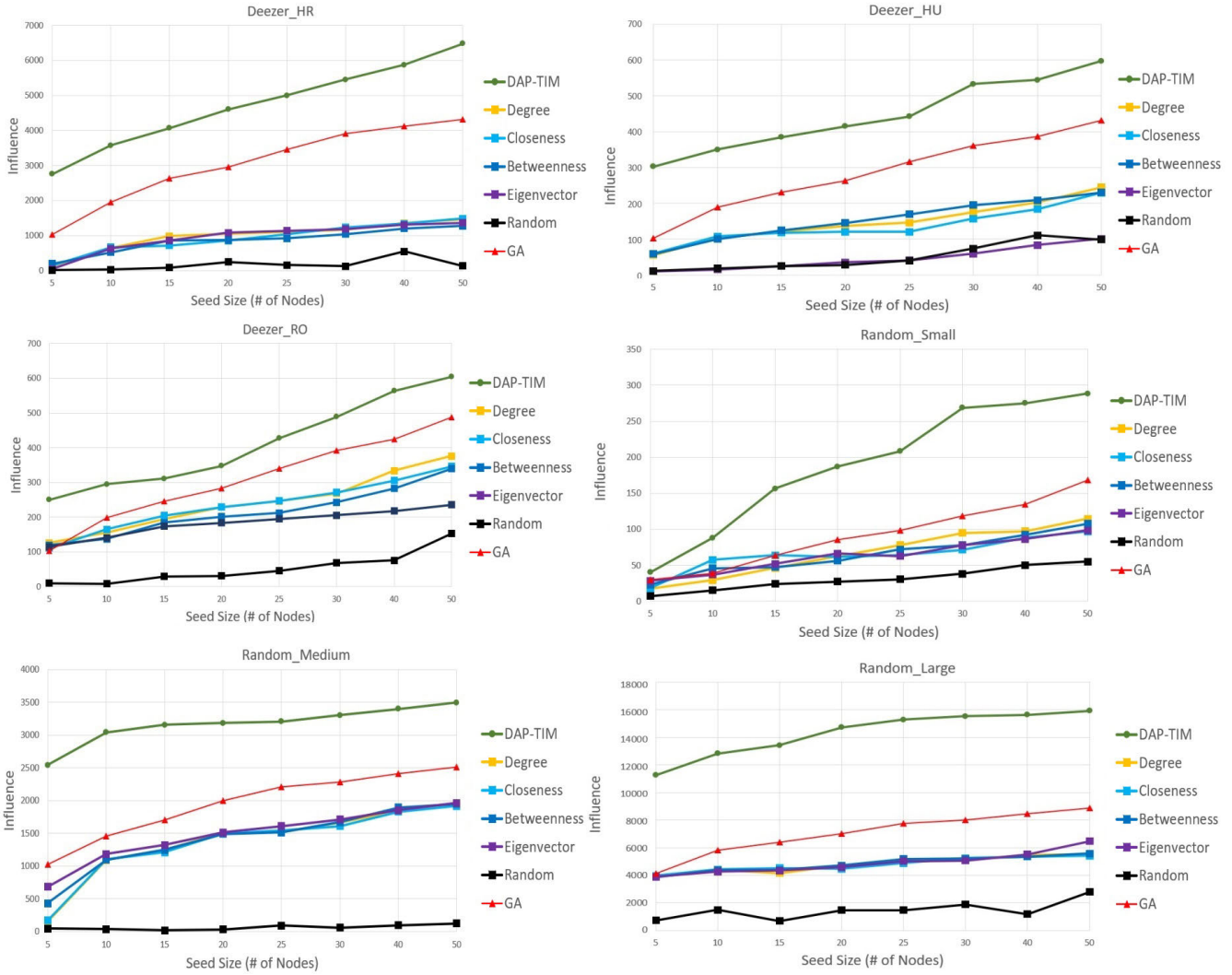

FIGURE 4. Comparison of the quality of algorithms.

Actor-Critic (A2C) algorithm has been utilized, which was introduced by Mnih et al. [23]. A2C is a powerful RL technique that combines the benefits of both actor-critic and asynchronous methods, improving the stability and efficiency of training reinforcement learning models.

A2C consists of two main components: the actor and the critic. The actor selects actions based on the current policy, while the critic evaluates these actions and provides feedback to the actor. This integration allows for simultaneous policy optimization and value function estimation, leading to more stable and faster convergence. A2C's advantages lie in its simplicity, reduced variance, and improved sample efficiency compared to other reinforcement learning algorithms. Additionally, its modular structure and straightforward implementation make it a popular choice for solving complex problems across various domains. The DAP-TIM framework based on A2C is illustrated in Fig. 3.

The steps of the algorithm can be stated as follows. Initially, the actor selects an action based on its current policy π. $\pi(s)$ is approximated using a deep neural network, which takes state s as an input and generates different probabilities for each combination of (a, b), where a indicates the centrality measure whose weight is to be improved, and b indicates the extent of increased applied to that weight. A combination with a higher probability is more likely to be selected. This network is updated according to the changes in influence. The selected action is sent to the coordinator. The coordinator applies this action to the current state s_t, generates a new state s_{t+1}, and delivers it to the SG module. The SG module, according to the Fig. 2, calculates the solution and returns it to the coordinator. The coordinator, based on the reward function defined in section III-B, returns the reward r_{t+1} and new state s_{t+1} to the actor and critic. The critic, based on the received reward, provides feedback to the actor, and the actor updates its policy based on that feedback. Based on the calculated rewards at each step, the algorithm tries to adjust the weights of measures to ultimately achieve the best influence.

Note that each training episode concludes when the reward stabilizes, defined as the absolute difference between

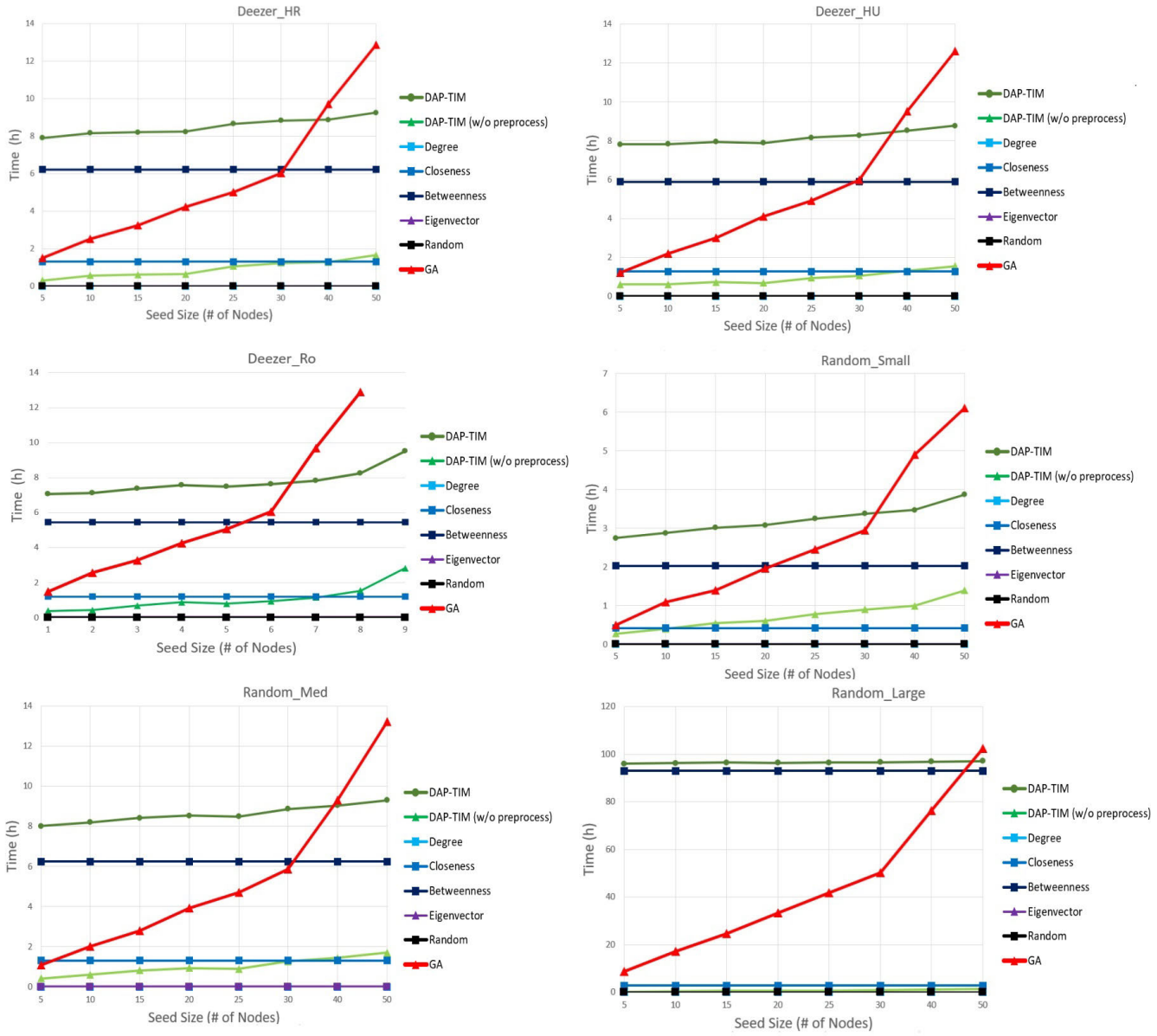

FIGURE 5. Comparison of the running time of algorithms.

consecutive rewards being less than 30 for 10 consecutive steps, indicating convergence. The overall training process is limited to a maximum of T (100 in our case) episodes. The pseudo-code of DAP-TIM algorithm based on A2C is illustrated in **Algorithm 1: A2C-TIM**.

IV. RESULTS AND DISCUSSION

This section evaluates the proposed method on various networks. It begins by introducing the datasets and explaining the preprocessing steps applied to them. Subsequently, the section presents the results and analysis obtained from the algorithm's execution and compares them with baselines. The parameters of the proposed method were set as follows: a learning rate of 0.003, a discount factor of 0.9, and 100 episodes. To validate these settings, hyperparameter testing was conducted using a grid search over ranges such as learning rate {0.001, 0.003, 0.005, 0.01}, discount factor {0.7, 0.8, 0.9, 0.95}, and episodes {50, 100, 150, 200}. Results demonstrated that the selected parameters consistently achieved competitive performance across datasets, with negligible differences compared to other configurations.

A. DATASET

One challenge in evaluating the proposed algorithm is the lack of a public dataset for TIM. While several social networks datasets exist, they are not weighted by topics. Therefore, we conducted this task ourselves. To do this, six datasets have been utilized, half of which are artificially generated to mimic the structure of real-world social networks. The distribution of user interests in these network is designed to be similar to that of three

other real-world networks. Following introduces the six datasets:

- **Real-world Datasets (Deezer Social Network)**: Deezer is a music-based social network where users share their favorite genre playlists. The dataset [24] contains nodes and edges from Deezer social network in three European countries: Hungary, Romania, and Croatia. For each individual, their favorite music genres are stored as an array of genre names. In total, there are 84 different music genres in all three datasets. The reason for choosing this network is the availability of sufficient data to construct an appropriate graph for TIM.
- **Synthetic Datasets**: To further investigate the behavior of the proposed method, three synthetic graph with different sizes are randomly generated using the algorithm introduced by Barabási and Albert [25]. This algorithm allows the creation of a graph with a specified number of nodes and edges, possessing a normal degree distribution, resembling real-world social networks. User interest vectors were sampled from the Deezer Hungarian network to match the interest distributions of other datasets.

Table 1 provides a summary of the characteristics of the networks. The datasets related to these six networks are publicly available at [26].

B. DATA PREPARATION

To implement the proposed method, two key parameters need to be calculated from the information available in the Deezer dataset: the interest vector of each individual ($U_{interest}(u)$) and the weight of each edge (w_{uv}).

- **User Interest**: Since the number of topics in the network is 84, the interest of each individual in various music genres is defined as an 84-dimensional vector of 0s and 1s (as in Eq. (2)). For each user, i-th index is set to 1 if topic i is in the user's list of favorite topics, and 0 otherwise.
- **Edge Weight**: The similarity between two users u_1 and u_2 is calculated as the dot product of their interest vectors as in Eq. (9):

$$Similarity(u_1, u_2) = U_{interest}(u_1) \cdot U_{interest}(u_2) \quad (9)$$

The edge weight $w_{u_1u_2}$ is calculated by normalizing the similarity score using the total number of unique genres that either u_1 or u_2 is interested in. That is, $w_{u_1u_2}$ is defined as the similarity between u_1 and u_2 divided by the total number of unique topics interested by two users, and is calculated as in Eq. (10):

$$w_{u_1u_2} = \frac{Similarity(u_1, u_2)}{\sum_{i=1}^{Z}(U_{interest}(u_1)[i] \vee U_{interest}(u_2)[i])}, \quad Z = 84 \quad (10)$$

where $\vee$ denotes the logical OR operation. For instance, if u_1 is interested in rock and hip-hop, and u_2 is interested in pop and rock, the edge weight would be $\frac{1}{3}$, representing the overlap of one genre (rock) between the two users. If u_2 is only interested in rock, the edge weight would be $\frac{1}{2}$, indicating a higher correlation due to the shared interest in rock. To determine the probability of influencing a neighbor, Eq. (4) is used. Thus, due to the differences in the interests of each individual and the attractiveness of a song for each person, the probability of diffusion from u to v and v to u will be different, aligning more closely with real-world scenarios.

C. EVALUATION DETAILS

The proposed method was evaluated on a system running Windows with a 12-core processor and 20 GB of RAM, with all implementations carried out in Python. The source code is publicly available in [26].

In each network, different music tracks with different topic vectors were propagated (generated randomly), and the results, in the form of average influence, are presented. For a more comprehensive evaluation of the proposed method, it has been compared with the following methods:

- **Degree-based Greedy (Degree)**: Nodes are ranked by degree centrality, prioritizing those with the highest number of direct connections. The top k nodes with the highest degree are selected.
- **Closeness-based Greedy (Closeness)**: Nodes are ranked by closeness centrality, which measures the average shortest path from a node to all others. The top k nodes with the lowest average path lengths are selected.
- **Betweenness-based Greedy (Betweenness)**: Nodes are ranked by betweenness centrality, which quantifies the number of shortest paths passing through a node. The top k nodes with the highest betweenness scores are selected.
- **Eigenvector-based Greedy (Eigenvector)**: Nodes are ranked by eigenvector centrality, which measures a node's influence based on the importance of its neighbors. The top k nodes with the highest eigenvector scores are selected.
- **Random Selection Method (Random)**: In this method, k random nodes are selected.
- **Genetic Algorithm (GA)**: Inspired by the research of Bucur and Iacca [27], where genetic algorithm was employed to solve IM.
- **DAP-TIM**: The proposed method which employs SG, reinforcement learning, and six centrality measures.

The selected algorithms are compared to highlight the advantages of DAP-TIM's dynamic, adaptive, and parametric features in reducing search time and enhancing solution quality, particularly compared to search-based methods (e.g., genetic algorithms) and static centrality-based algorithms.

For a more comprehensive evaluation, topics in each network are sorted by popularity, and the following conditions are used to select a propagating topic:

TABLE 2. Statistical measures (standard deviation and p-value) on Deezer_RO dataset with $k = 50$.

DAP-TIM vs.	Mean ± Std Dev	p-value
DAP-TIM	604.2 ± 36.2	-
Degree	376.3 ± 7.5	2.87E-11
Closeness	346.2 ± 6.9	2.34E-12
Betweenness	339.8 ± 6.7	2.93E-11
Eigenvector	236.0 ± 4.7	3.21E-10
Random	153.2 ± 22.9	2.17E-13
GA	488.2 ± 6.9	1.83E-10
Modularity Vitality	508.1 ± 37.5	4.01E-10
Map Equation	493.9 ± 39.5	7.03E-11
Overlapping Modular	493.2 ± 34.5	7.27E-11
DIEM	540.5 ± 32.4	1.42E-7
TopicSample	528.3 ± 36.4	3.66E-10

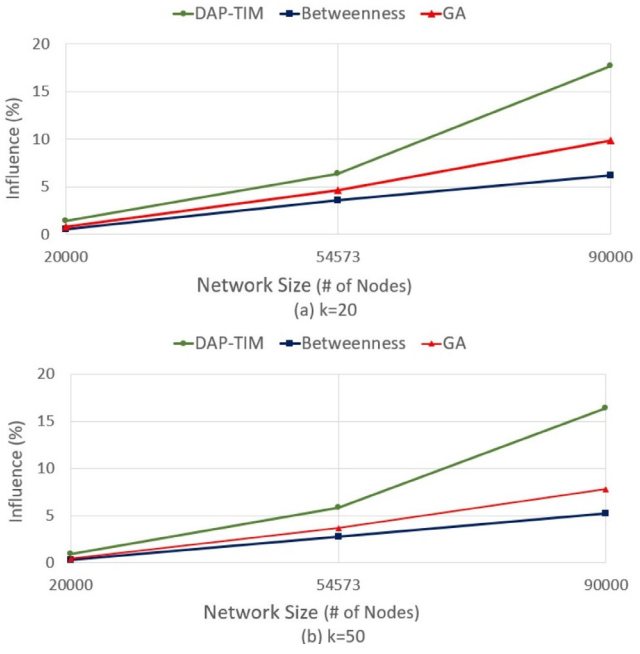

FIGURE 6. Comparison of the scalability of algorithms w.r.t. output quality (a) k=20, (b) k=50.

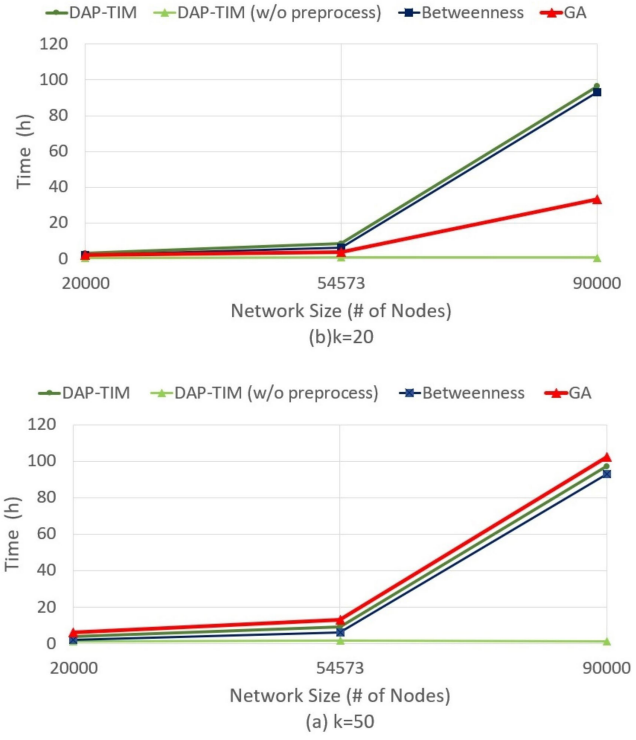

FIGURE 7. Comparison of the scalability of algorithms w.r.t. running time (a) k=20, (b) k=50.

- 10% of the topic is related to the most popular topic in the network.
- 30% of the topic is related to the median topic in the network.
- 60% of the topic is related to the least popular topic in the network.

The final results of each algorithm are averaged over 30 independent runs and analyzed on three metrics: output quality, running time, and scalability.

D. OUTPUT QUALITY

The results for influence metric for different sizes of k, are depicted in Fig. 4. As observed, DAP-TIM has significantly outperformed the baseline methods. The results indicate that two dynamic centrality measures, RF and CF, along with adaptively adjusting the weights of centrality measures during the course of algorithm significantly improve the exploration capability of the algorithm, and impact the quality of the final result.

Note that the quality of the output of greedy methods compared to each other does not show a significant difference. By analyzing the output of these algorithms, it was determined that this arises from the fact that the use of different static centrality measures in ranking nodes often leads to the selection of the same set of k nodes, although the order of nodes may differ. However, by incorporating the two dynamic measures along with static centrality measures, into the node selection process during the generation of the solution set, the order of node selection becomes important. Since SG selects nodes one by one, the output solution generated by SG will differ significantly from greedy algorithms.

E. RUNNING TIME

The results for running time metric for all algorithms on all networks are depicted in Fig. 5.

After analyzing the running time results, it was observed that RL algorithms require less time compared to the GA. This is due to smaller search space of the RL algorithm, with SG guiding it towards better solutions, allowing it to converge with fewer trials and errors than GA.

It is evident that the time complexity of greedy algorithms is independent of the size of k, as nodes are sorted based on their centrality score only once. In contrast, the running time of DAP-TIM and GA methods depends on the size of k, as they primarily rely on calculating influence of k-node solutions, which increases with k. GA, compared to other methods, requires more influence calculations for potential solutions, leading to a faster increase in running time with respect to k.

In Fig. 5, DAP-TIM (w/o preprocess) refers to the running time for DAP-TIM excluding the time spent on calculating the four static centrality measures. In a static network, these centrality measures are only calculated once, and for changes in k, only retraining the RL model is necessary. The results show that the proposed method is significantly more efficient when excluding pre-processing time. This also indicates that the two dynamic measures and the RL approach have significantly lower computational complexity compared to static measures, making them suitable for use in complex networks.

F. SCALABILITY

To assess the impact of network size on the output quality and running time of algorithms, the results for k=20 and k=50 on the three random networks are presented. The scalability analysis results for output quality and running time are shown in Fig. 6 and Fig. 7, respectively. For simplicity, and since all greedy algorithms had the same behaviors, only the results of the Betweenness algorithm have been illustrated as a representative example.

As results show, the proposed algorithm maintains its performance in larger network sizes. It even achieves a higher quality of results compared to other methods, as the network size increases. It suggests that DAP-TIM may be a highly suitable option for analyzing large and complex networks.

Regarding running time, as the network size increases, the running time of the greedy algorithm also increases. Furthermore, since SG in DAP-TIM utilizes centrality scores, the running time of DAP-TIM also increases. According to the scalability analysis results, the overhead of using RL in DAP-TIM is negligible. Therefore, the complexity of DAP-TIM mostly depends on calculating centrality scores. Thus, by employing less computationally intensive measures, DAP-TIM would be more scalable in terms of running time.

G. COMPARISON WITH OTHER METHODS

Although the primary aim of this paper is to illustrate how dynamically adjusting the importance of various centrality measures can enhance the effectiveness of traditional approaches, we extend our analysis by comparing DAP-TIM with three community-aware centrality measures and two TIM-based methods for a more comprehensive evaluation. Community-aware centrality methods emphasize the influence of nodes within specific communities or clusters, offering insights into local network structures. Since community-aware centrality measures have gained much attention recently [28], [29], [30], this comparison is useful for illustrating the strengths of proposed approach in identifying influential nodes, while also positioning it within the broader landscape of centrality measures. Furthermore, comparing with TIM-based methods allows us to evaluate the performance of proposed method more fairly. Thus, for comparison, we selected five recent and well-known methods: the first three are community-based, and the last two are TIM-specific methods.

- Modularity Vitality [31]
- Map Equation [32]
- Overlapping Modular [33]
- DIEM [18]
- TopicSample [5]

The results for output quality and running time for all algorithms on all networks are depicted in Fig. 8 and Fig. 9, respectively.

As can be seen in Fig. 8, TIM-specific methods outperform the community-aware algorithms on influence since they dynamically adapts parameters based on topic relevance, potentially allowing it to target more influential nodes within topic clusters. Also, DAP-TIM outperforms other TIM-specific methods because it dynamically change the importance of centrality measures according to the current state of the propagation.

Also, according to Fig. 9, DAP-TIM has a longer runtime compared to other methods. However, excluding the preprocessing phase and centrality measure calculations, makes its runtime comparable to the others. Therefore, it appears that using alternative centrality measures instead of the four static measures applied in this paper could significantly improve the runtime of the proposed algorithm.

H. STATISTICAL ANALYSIS

As stated before, the final results of each algorithm are averaged over 30 independent runs and analyzed. To ensure the reliability and validity of our experimental findings, we report detailed statistical measures for all evaluated algorithms. The statistical results for all algorithms on Deezer_RO dataset with $k = 50$ has been shown in Table 2. Similar trends were observed across other networks and different seed sizes, highlighting the generality of the findings. The variance of influence across these runs is quantified using standard deviation, which was less than 5.8% for DAP-TIM (for all datasets and different values of k). This low variance reflects the inherent randomness in propagation models but remains within an acceptable range for reliable comparisons.

Greedy algorithms that rely on centrality measures have the lowest variance, with a maximum standard deviation of

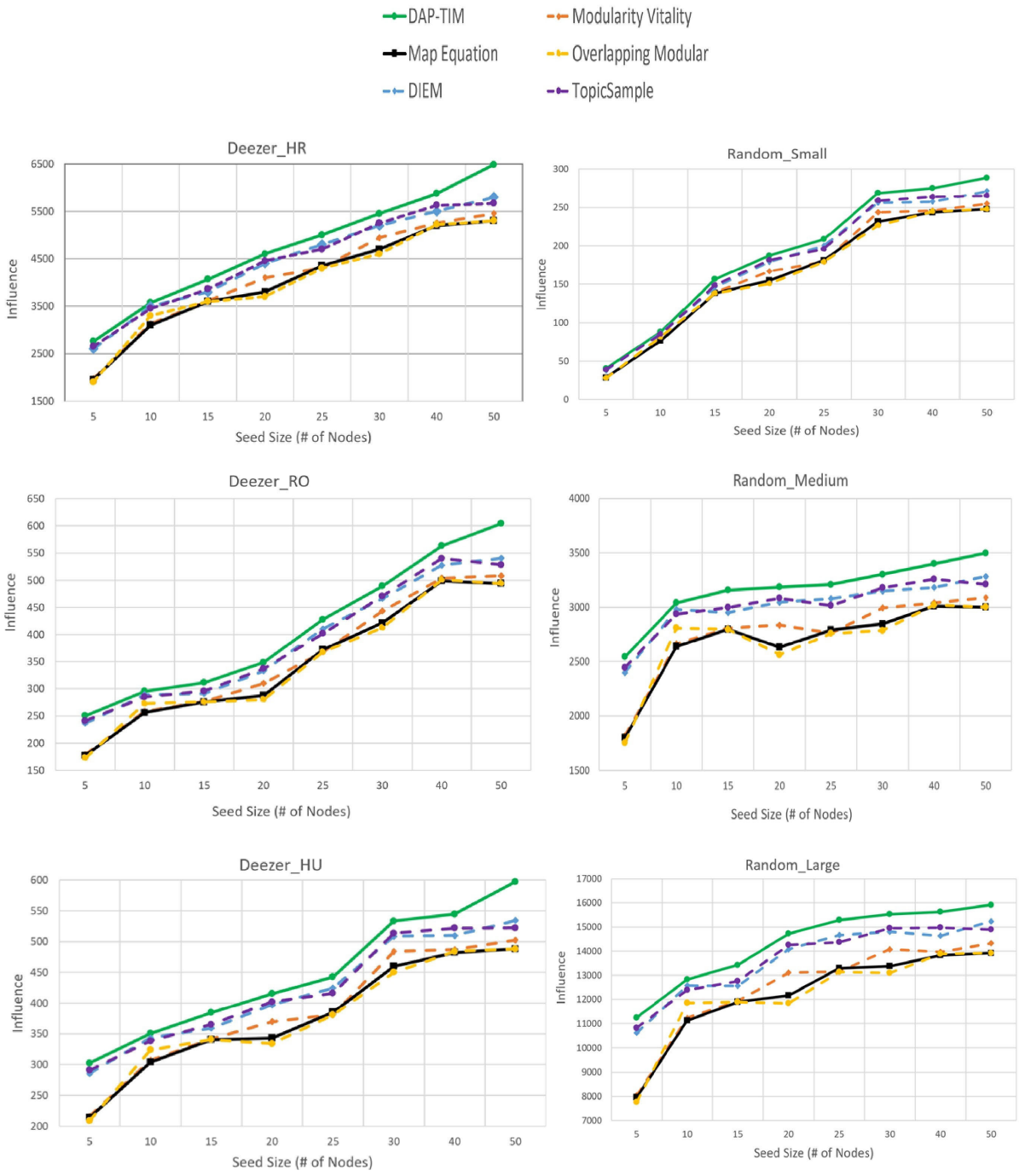

FIGURE 8. Comparison of influence of algorithms.

5%. This minimal variability is due to their deterministic selection process, which ensures consistent results regardless of random factors in the propagation model. In contrast, the random algorithm shows the highest variance at 23.1% (for $k = 10$), stemming from its purely stochastic node selection.

To evaluate the statistical significance of performance differences, we conducted a two-tailed t-test to calculate the p-value between DAP-TIM and each of the competing algorithms. The results, presented in Table 2, show that all p-values are below the commonly accepted threshold of 0.05 ($p < 0.05$). This indicates that the observed differences in influence spread between DAP-TIM and other algorithms are statistically significant. The consistently low p-values validate the superiority of DAP-TIM in achieving higher influence spread compared to alternative methods. These results provide strong statistical evidence supporting the robustness and effectiveness of the proposed algorithm,

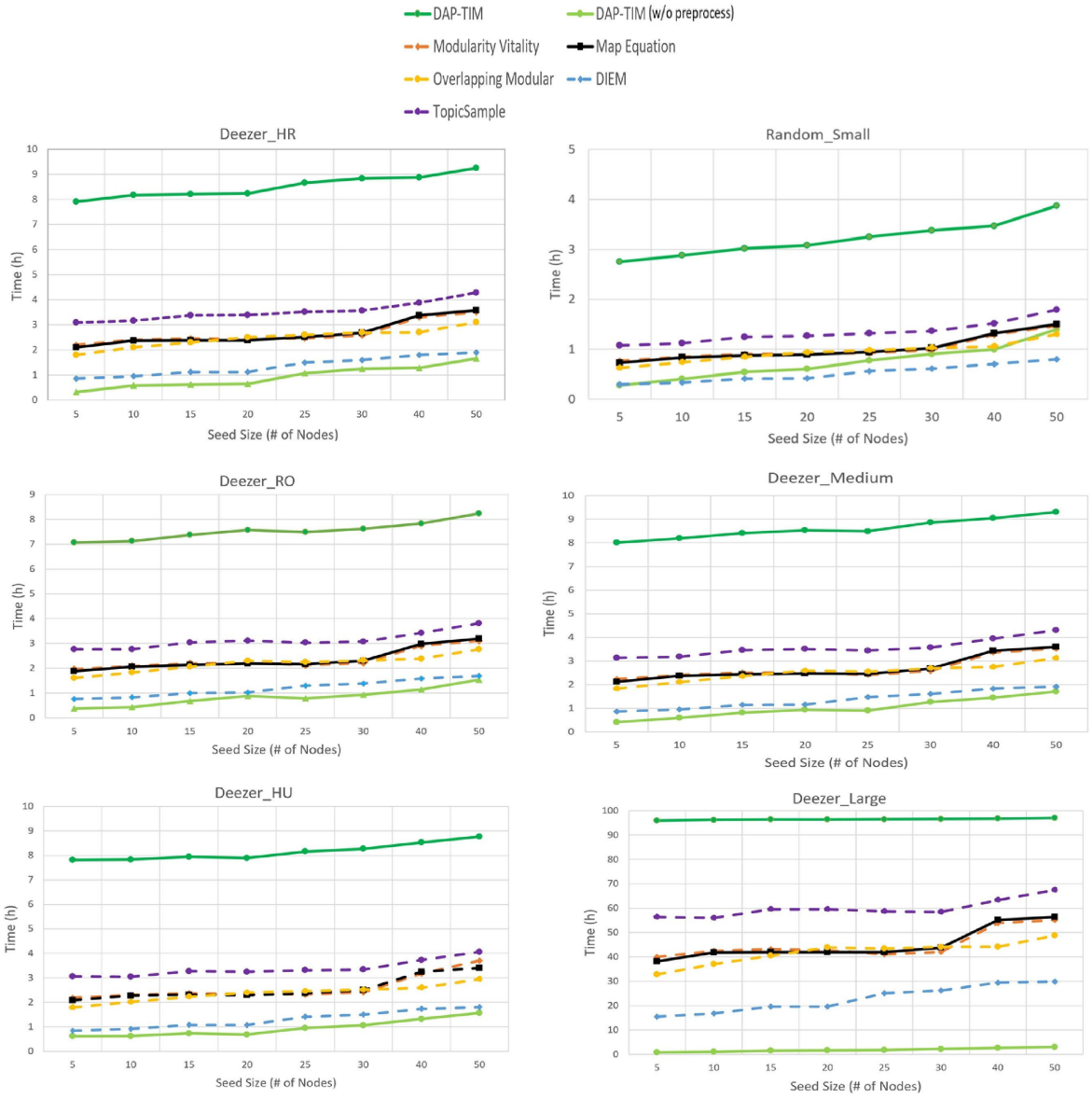

FIGURE 9. Comparison of running time of algorithms.

emphasizing its ability to outperform other state-of-the-art approaches across various metrics and datasets.

V. CONCLUSION AND FUTURE WORK

We introduced a dynamic adaptive parametric network analysis approach based on reinforcement learning. The parametric strategy reduces the exploration space by converting TIM into a parametric optimization problem, with the goal of determining the significance of each centrality measure for optimal results. We introduced two dynamic centrality measures that dynamically capture the significance of nodes during the topic propagation in the network. Furthermore, to ensure that the method is instance-independent, we've employed an adaptive technique utilizing reinforcement learning, which adjusts the significance of each measure throughout algorithm execution, considering the current network structure, and generates solutions tailored to the specific network.

Due to lack of dataset availability for topic-aware IM, the social network data from Deezer has been created and is publicly available in [26], making it ready for use in future research and similar studies.

The results of the DAP-TIM algorithm indicate its effectiveness in large and complex networks. This conclusion is drawn from the algorithm's performance, which demonstrates promising results in terms of both output quality and running time. Despite the challenges posed by the computational complexity of calculating centrality scores, the algorithm is scalable, especially when using less computationally intensive measures or their approximations. Overall, the findings suggest that DAP-TIM holds promise for addressing the challenges of network analysis

in real-world scenarios, particularly in large and complex networks.

Future research can explore several directions, such as investigating alternative centrality metrics or approximations to reduce computational complexity, enhancing performance for real-time and online scenarios. Another avenue is studying TIM in scenarios where topic propagation significance changes over time—like how news relevance spikes closer to an event or how interest in movie campaigns wanes after release. Additionally, due to the lack of large datasets, evaluating DAP-TIM on larger networks is a key priority for future work. Comparative studies between DAP-TIM and other contemporary algorithms could further clarify its advantages and performance.

REFERENCES

[1] S. S. Singh, D. Srivastva, M. Verma, and J. Singh, "Influence maximization frameworks, performance, challenges and directions on social network: A theoretical study," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7570–7603, Oct. 2022.

[2] K. Rahimkhani, A. Aleahmad, M. Rahgozar, and A. Moeini, "A fast algorithm for finding most influential people based on the linear threshold model," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1353–1361, Feb. 2015.

[3] Ç. Aslay, N. Barbieri, F. Bonchi, and R. Baeza-Yates, "Online topic-aware influence maximization queries," in *Proc. EDBT*, Mar. 2014, pp. 295–306.

[4] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," *Knowl. Inf. Syst.*, vol. 37, no. 3, pp. 555–584, Dec. 2013.

[5] S. Chen, J. Fan, G. Li, J. Feng, K.-L. Tan, and J. Tang, "Online topic-aware influence maximization," *Proc. VLDB Endowment*, vol. 8, no. 6, pp. 666–677, Feb. 2015.

[6] Y. Li, D. Zhang, and K.-L. Tan, "Real-time targeted influence maximization for online advertisements," *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 1070–1081, Jun. 2015.

[7] F. Bonchi, C. Castillo, and D. Ienco, "The meme ranking problem: Maximizing microblogging virality," *J. Intell. Inf. Syst.*, vol. 40, no. 2, 2013, Art. no. 211239.

[8] W. Chen, B. Peng, G. Schoenebeck, and B. Tao, "Adaptive greedy versus non-adaptive greedy for influence maximization," *J. Artif. Intell. Res.*, vol. 74, pp. 303–351, May 2022.

[9] B. Razaghi, M. Roayaei, and N. M. Charkari, "On the group-fairness-aware influence maximization in social networks," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 6, pp. 3406–3414, Dec. 2023.

[10] K. Huang, J. Tang, K. Han, X. Xiao, W. Chen, A. Sun, X. Tang, and A. Lim, "Efficient approximation algorithms for adaptive influence maximization," *VLDB J.*, vol. 29, no. 6, pp. 1385–1406, Nov. 2020.

[11] M. Hosseini-Pozveh, "New influence-aware centrality measures for influence maximization in social networks," *J. Comput. Secur.*, vol. 9, no. 2, pp. 23–38, 2022.

[12] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. discovery data mining*, Aug. 2003, pp. 137–146.

[13] J. Jabari Lotf, M. Abdollahi Azgomi, and M. R. Ebrahimi Dishabi, "An improved influence maximization method for social networks based on genetic algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 586, Jan. 2022, Art. no. 126480.

[14] T. K. Biswas, A. Abbasi, and R. K. Chakrabortty, "An MCDM integrated adaptive simulated annealing approach for influence maximization in social networks," *Inf. Sci.*, vol. 556, pp. 27–48, May 2021.

[15] Y. Li, H. Gao, Y. Gao, J. Guo, and W. Wu, "A survey on influence maximization: From an ML-based combinatorial optimization," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 9, pp. 1–50, Nov. 2023.

[16] H. Li, M. Xu, S. S. Bhowmick, J. S. Rayhan, C. Sun, and J. Cui, "PIANO: Influence maximization meets deep reinforcement learning," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 3, pp. 1288–1300, Jun. 2023.

[17] H. Chen, B. Wilder, W. Qiu, B. An, E. Rice, and M. Tambe, "Complex contagion influence maximization: A reinforcement learning approach," in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, Aug. 2023, pp. 5531–5540.

[18] S. Tian, S. Mo, L. Wang, and Z. Peng, "Deep reinforcement learning-based approach to tackle topic-aware influence maximization," *Data Sci. Eng.*, vol. 5, no. 1, pp. 1–11, Mar. 2020.

[19] H. T. Nguyen, T. N. Dinh, and M. T. Thai, "Cost-aware targeted viral marketing in billion-scale networks," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2016, pp. 1–9.

[20] H. Huang, Z. Meng, and H. Shen, "Competitive and complementary influence maximization in social network: A follower's perspective," *Knowl.-Based Syst.*, vol. 213, Feb. 2021, Art. no. 106600.

[21] Y. Wang and Y. Wang, "Opinion-aware influence maximization in online social networks," 2023, *arXiv:2301.11527*.

[22] L. Teng, Y. Wang, Z. Lin, and F. Yu, "Topic-aware most influential community search in social networks," 2024, *arXiv:2402.07601*.

[23] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. Lillicrap, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 1928–1937.

[24] A. Garritano. *Deezer Social Network*. Accessed: Apr. 17, 2025. [Online]. Available: https://www.kaggle.com/datasets/andreagarritano/deezer-social-network

[25] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[26] *DAP-TIM*. Accessed: Apr. 17, 2025. [Online]. Available: https://github.com/mroayaei/DAP-TIM

[27] D. Bucur and G. Iacca, "Influence maximization in social networks with genetic algorithms," in *Proc. 19th Eur. Conf. EvoApplications Appl. Evol. Comput.*, Porto, Portugal. Cham, Switzerland: Springer, Mar. 2016, pp. 379–392.

[28] H. Cherifi, G. Palla, B. K. Szymanski, and X. Lu, "On community structure in complex networks: Challenges and opportunities," *Appl. Netw. Sci.*, vol. 4, no. 1, pp. 1–35, Dec. 2019.

[29] S. Rajeh, M. Savonnet, E. Leclercq, and H. Cherifi, "Comparative evaluation of community-aware centrality measures," *Qual. Quantity*, vol. 57, no. 2, pp. 1273–1302, Apr. 2023.

[30] S. Rajeh, M. Savonnet, E. Leclercq, and H. Cherifi, "Characterizing the interactions between classical and community-aware centrality measures in complex networks," *Sci. Rep.*, vol. 11, no. 1, p. 10088, May 2021.

[31] T. Magelinski, M. Bartulovic, and K. M. Carley, "Measuring node contribution to community structure with modularity vitality," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 707–723, Jan. 2021.

[32] C. Blöcker, J. C. Nieves, and M. Rosvall, "Map equation centrality: Community-aware centrality based on the map equation," *Appl. Netw. Sci.*, vol. 7, no. 1, p. 56, Aug. 2022.

[33] Z. Ghalmane, C. Cherifi, H. Cherifi, and M. E. Hassouni, "Centrality in complex networks with overlapping community structure," *Sci. Rep.*, vol. 9, no. 1, p. 10133, Jul. 2019.

MOHAMMAD HOSSEIN AHMADIKIA received the M.S. degree in computer engineering from Tarbiat Modares University, Tehran, Iran, in 2023. His research interests include artificial intelligence, with a focus on applying reinforcement learning to real-world problems.

MEHDY ROAYAEI received the B.S., M.S., and Ph.D. degrees in computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2008, 2010, and 2016, respectively. Since 2018, he has been an Assistant Professor with Tarbiat Modares University. He has published more than 40 works, including journal articles, conference papers, and book chapters. His research interests include artificial intelligence, with a focus on applying reinforcement learning to real-world problems. He is actively engaged in the research community, contributing as a peer reviewer of various prestigious academic journals.