Influence Maximization using Time Delay based Harmonic Centrality in Social Networks

Salman Mokhtarzadeh[1], Behzad Zamani Dehkordi[*1,2], Mohammad Mosleh[1], Ali Barati[1]

[1]Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran
[2]Department of Engineering, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran
Email addresses: bzamani@iaushk.ac.ir
[*]Corresponding author

Received: 2021-10-20
Revised: 2022-01-15
Accepted: 2022-01-26

Abstract
With the extension of social networks, research on influence maximization (IM) in time-sensitive graphs has increased in recent years. IM is a problem to find a seed set with k nodes to maximize the information propagation range in the graph. Most of the research in this area consists of greedy, heuristic, meta-heuristic methods. However, most of these methods ignore the time-sensitivity to propagation delay and duration. The preceding time-sensitive centrality measures as a part of heuristic approaches take the propagation delay but only consider the nodes locally so that each graph node considers only the direct neighbors. Based on the above analysis, this article focuses on the time-sensitive IM problem. Here, a propagation value for each path in the graph is defined in terms of the probability of affecting through the edge and freshness amount of the edge. To solve the problem, we propose time-sensitive centrality measures that consider propagation value and both the direct and the indirect neighbors. Therefore, four measures of time-sensitive closeness centrality (TSCloseness), time-sensitive harmonic (TSHarmonic), time-sensitive decay centrality (TSDecay), and time-sensitive eccentricity centrality (TSEccentricity) were proposed. The experiments on five datasets demonstrate the efficiency and influence performance of the TSHarmonic measure on evaluation metrics.

Keywords
Influence Maximization, Propagation Delay, Closeness centrality, Harmonic centrality, Decay centrality, Eccentricity centrality.

1. Introduction

We live in a world where various social networks affect every individual including family, friends, and partners, directing the individual's lifestyle, buying/purchase, traveling and selecting the political party. One of the sciences that has emerged with the development of social networks is social network analysis. In the social network analysis using mathematics and graph theory, a better understanding of the network members is provided. The social network analysis can be used for link prediction, community detection, opinion mining, and influence maximization.

The detection of *k* individuals in a social network with n members, so that these *k* individuals obtain the maximum influence on the other members of the network, is called influence maximization, which is one of the factors of the social network analysis. Among the uses of influence maximization are viral marketing [1] and Influence blocking maximization (IBM) [2, 3]. To detect the influential nodes, Kempe et al. [4] presented a greedy algorithm; since the proposed method is of high time complexity, various researchers have attempted to improve its time complexity based on various methods.

With the development of social networks in human societies, these networks have also evolved and collected additional information, including information transmission location or transmission time. By adding this additional information, new types of influence maximization methods are developed, known as content-sensitive influence maximization. One class of these methods is known as time-based methods which attempt to detect influential nodes considering the time dimension. Mohammadi et al. [5] presented a greedy method for detecting the time-sensitive influential nodes that use delay-sensitive independent cascade propagation instead of independent cascade propagation. Since this method is greedy, it suffers from a low running order. One approach to increase the speed of detecting influential nodes in a social network is to use heuristic methods instead of greedy methods. Although in heuristic methods, the accuracy is not guaranteed [6], they are used as efficient methods due to their high speed of detection of the influential nodes. Using the centrality measures for detecting the influential nodes is an example of the heuristic methods; however, their accuracy in detecting influential nodes is not high, but they are used to detect the influential nodes due to their low running time.

In most centrality measurements, the propagation delay from one node to another has not been considered. Thus, when these measurements are used in a real network, the propagation delay is effective as a hidden parameter, and the result would not be desired. Thus, the purpose of this paper is to introduce several centrality measurements considering the propagation delay in the social networks and using the most valuable path instead of the shortest path, and using the measurements to detect the influential nodes based on the delay-sensitive cascade propagation model[5]. Hence, four centrality methods, including time-sensitive closeness, time-sensitive harmonic, time-sensitive decay and time-sensitive eccentricity are proposed.

The innovations of this paper are as follows:

- Presenting the time-sensitive closeness centrality method
- Presenting the time-sensitive harmonic centrality method
- Presenting the time-sensitive decay centrality method
- Presenting the time-sensitive eccentricity centrality method

In this paper, Section 2 reviews the previous studies. Section 3 presents the primary concepts and the conventional centrality methods. Section 4 introduces the proposed methods. Section 5 presents the simulation results. Finally, the paper is concluded in Section 6, and future suggestions are given.

2. Literature Review

The influence maximization was first presented by Richardson and Domingos [7] and Kempe et al. [4] extended it as an optimization problem. Since the greedy method presented by Kempe et al. has high time complexity, various methods have been presented to improve the execution time and increase the scalability, among which CELF (Cost-Effective Lazy Forward) [8], CELF++ [9], CGA (Community-based Greedy Algorithm) [10], IRIE (Influence Ranking Influence Estimation) [11], INN (Influential Nodes detection according to Neighbors) [12], PMIA (Prefix excluding Maximum Influence Arborescence) [13] and LDAG (Local Directed Acyclic Graph) [14] GWIM (Grey Wolf based Influence Maximization)[15], ECRM (Extended Cluster Coefficient Ranking Measure) [16], MOSI (Multi-Objective algorithm based on Structured Information) [17] can be mentioned.

Most methods of detection of the influential nodes have high time-consuming; thus, various techniques have been presented for this drawback. One class of these techniques uses centrality measures to detect influential nodes that have a higher order than other methods despite their lower accuracy, attracting the attention of the scientists. Among these methods [18], DegreeDiscount (Degree Discount heuristic) [19], DegreePunishment (Degree Punishment heuristic) [20], DegreeDistance (degree distance heuristic) [21], MCIM (Multi-Constraint Influence Maximization) [22], Katz [23], SDC (Shell Degree Centrality) [24], time-sensitive degree [5] and time-sensitive Betweenness [5] can be mentioned.

Since time affects the information transmission from one individual to another in the real world, it has been examined in studies related to influential node detection, developing another type of influential node detection method called time-sensitive influential node detection.

In CTIC (Continuous Time delay Independent Cascade) [25], to consider the time significance, the information propagation delay was considered on each edge in addition to the information propagation probability. In MIA-M (MIA for independent cascade Model) and MIA-C (MIA with Converted propagation probabilities) [26], the time significance was examined by considering a specific time for the information propagation in a network, such that the time delay along a propagation path is considered a Bernoulli function. In time-sensitive greedy [5], a greedy method was presented to detect the influential nodes based on the propagation delay, which is similar to the method presented by Kempe et al. with the difference that in this method, a node is more important if its propagation delay is smaller than the other nodes.

In CT-IPA (Continuously activated and Time-restricted Influence Path Algorithm) [27], two features have been considered. First, each active node can activate its neighbors' multiple times (unlike the independent cascade method where each node can only try one to activate its neighboring nodes). Second, activating the nodes takes a specific time; that is, information propagation in the network is limited to a specific time.

In SA (Scalable Algorithm) [28], a delay-based diffusion model called IDM (Influence Decay Model) is introduced, which instead of considering the delay as an independent parameter, is applied to the probability of impact between nodes, so that over time, the probability of being affected logarithmically between nodes decreases. Naoto et al. [29] also considered the probability of impact between nodes to be variable over time due to propagation delays. In TCIO (Time and Cost constrained Influence model with users' Online patterns) [30], each node propagates information several times to other nodes considering a limited budget, online patterns, and expiry time.

In TTG (Topic-based Time-sensitive Greedy algorithm) and TTH (Topic-based Time-sensitive Heuristic algorithm) [31] for publishing information by each person, a time window is considered so that the end time of each window for that person is determined by considering the history of tweets of that person. In TP-IM (Time-sensitive Positive Influence Maximization) [32], a greedy method to identify the *k* most influential nodes in the signaled social network is presented by considering a specific time frame for information dissemination. The information dissemination model is based on the heat dissipation model. Among other scientists that have investigated the importance of time in information propagation, Goyal et al. [33] and Liu et al. [34, 35] can be mentioned.

As mentioned in [23], one of the approaches for detecting influential nodes is to use the centrality measures. The centrality measures are mathematical tools that can be used to analyze the nodes of a graph and assign a specific order to each graph node. In recent years, various centrality measures, including Degree [36], Betweenness [37], Closeness [36], Decay [38], Harmonic [39], Time-Sensitive Betweenness [5], Eccentricity [40], Time-Sensitive Degree[5], Local-Area [41], Semi-Local [42] have been presented, which are discussed in the following without considering time delay.

2.1. Degree Centrality

In the real world, individuals with many relationships are important people. In social networks, if a node has more connections, it is more important. This is the basis of the degree centrality method. The number of neighbors of a node is called the node's degree that can be used as the significance measure of the node in the degree centrality method, which is represented by $CD(u)$ in Eq. (1) [36].

$$CD(u) = d_u = \sum_{v\epsilon V} a_{uv} \quad (1)$$

where $\mathrm{d_u}$ is the degree of the node u and a_{uv} is the element of the u^{th} row and the v^{th} column of the adjacency matrix A of the graph $G = (V, E)$. This measure is a local assessment with no global vision to define the significance of nodes.

2.2. Betweenness Centrality

Betweenness is the ratio of the times that a node is located on the shortest path between different pairs of nodes in the graph. This measure shows the number of pairs of nodes in the network that depend on this node for a shorter connection with each other. The larger this measurement, the node would be in a more strategic location of the graph. Unlike the degree centrality method, this method uses a greater neighborhood area, including the nodes to which it is not directly connected to calculate the node's Betweenness in addition to its local neighborhood. In $G = (V, E)$, the times that the node u is located on the shortest path between nodes of s and t is called the Betweenness centrality, which is represented by $CB(u)$ in Eq. (2) [37].

$$CB(u) = \sum_{s\neq u\neq t\in V} \frac{dist_{st}(u)}{dist_{st}} \quad (2)$$

Here, $dist_{st}(u)$ is the number of shortest paths between s and t that pass node u, and $dist_{st}$ is the total number of shortest paths between s and t.

2.3. Closeness Centrality

In the $G = (V, E)$, the inverse total shortest distance of the node u from other nodes is called the closeness centrality measures, which is represented by $CC(u)$ in Eq. (3). The shortest path length between two nodes u and v in a graph is represented by σ_{uv} [36].

$$CC(u) = \frac{1}{\sum_{v\neq u\epsilon V} \sigma_{uv}} \quad (3)$$

The node with a larger closeness value has faster access to other nodes in the graph and can propagate information to other nodes in a shorter time. Similar to the Betweenness method, this method calculates the closeness of a node using a larger neighborhood area, including the nodes to which it is not directly connected in addition to the local neighborhood.

2.4. Decay Centrality

Decay centrality is a parameter-oriented centrality measure that evaluates the closeness of a node to other nodes in the network. However, unlike closeness centrality, the importance given to the geodesic distance (usually the number of edges is counted if the edges have no weight) is weighted by a parameter called the decay parameter, δ. The decay centrality measure that is represented by $CDK(u)$ in Eq. (4) uses the length of the shortest path to identify the significance of a node like the closeness centrality measure[38].

$$CDK(u) = \sum_{v\neq u\in V} \delta^{\sigma_{uv}} \quad (4)$$

where $0 < \delta < 1$. In fact, the farther a node is, its impact on the calculation of significance is lower. If δ is smaller, the weight assigned to the distance of the adjacent nodes is higher than the farther nodes. If δ is larger, the distance of each node has almost the same significance. Therefore, if δ is closer to zero, the decay centrality of the node has a strong positive correlation with the degree centrality of the node. If δ is closer to 1, the decay centrality of the node has a strong positive correlation with the closeness centrality of the node.

2.5. Harmonic Centrality

The harmonic centrality, also known as the centrality value, is a type of closeness centrality that has been presented to solve the problem that Eq. (3) encounters in disconnected graphs. In fact, it is a modified version of the closeness centrality with the difference that instead of using the inverse total of the shortest path from node u to other nodes as the value measure of each node, it uses the total inverse of the shortest path from node u to other nodes[39]. This measure is represented by $CH(u)$ in Eq. (5) [39].

$$CH(u) = \sum_{v\neq u\epsilon V} \frac{1}{\sigma_{uv}} \quad (5)$$

Also, if there is no path between the nodes u and v, $\frac{1}{\sigma_{uv}} = 0$. This measure resolves the absence of a path between nodes in the centrality measure.

2.6. Eccentricity Centrality

This measure is the modified version of the closeness centrality such that the inverse of the greatest distance between node u and other nodes are called eccentric, and it is represented by $CEC(u)$ in Eq. (6). In this measure, the node with the minimum greatest distance from other nodes is selected as the central node[40].

$$CEC(i) = \frac{1}{max_{v\in V} \sigma_{uv}} \quad (6)$$

2.7. Time-Sensitive Degree Centrality (TSDegree)

This measure that is represented by $CTSD(u)$ in Eq. (7) is an extended version of the degree measure that uses the total value of a node's neighbors instead of counting the neighbors of a node[5].

$$CTSD(u) = \sum_{j \in N^{out}(u)} ff(d_{uv}) \tag{7}$$

Here, $ff(d_{uv})$ describes the propagation value from node u to node v, and d_{uv} represents the propagation delay from node *u* to node *v*.

2.8. *Time-Sensitive Betweenness Centrality*

This measure that is represented by in Eq. (8) is an extended version of the Betweenness measure in which the propagation delay is considered instead of considering the path length [5].

$$CTSBet(u) = \sum_{x,y \in V} \frac{nsp_u(x,y)}{nsp(x,y)} \tag{8}$$

where $nsp(x,y)$ is the path with minimum delay between x and y, and $nsp_i(x,y)$ is the path with minimum delay between *x* and *y* that pass the node *u*.

3. Preliminaries

This section provides the preliminaries to the rest of the paper. We first describe the problem definition and properties and then present the time-sensitive influence maximization (TSIM) problem under DIC model.

3.1. *Problem definition and properties*

In TSIM, with graph *G(V,E)*, diffusion model *M*, positive integer *k* and freshness function *ff(t)*, the goal is to find the set *S* consisting of *k* nodes such that *PV(S)* under Model *M* is maximized; In other words, the goal is to find a set such as S^* that satisfies the condition $S^* = \arg max_{S \subseteq V, |S| \leq k} PV(S)$. In this definition, *PV(S)* is a function that determines the expected value of the nodes activated starting the process from the set *S* [5].

Information propagation value from one individual to another might decrease over time. Considering the type of information and application of its propagation, this decrease might have different speeds. For this purpose, the freshness function is used to determine the propagation value. The higher is the decrease rate of the freshness function, the activation time of the nodes plays a more significant role in determining the final profit of the propagation. According to different applications, different functions can be used as the freshness function; in [5], the function given in Eq. (9) is used, where $ff(0)$ is the initial value of the function at 0s (it is considered to be 1 by default), *t* and λ are the elapsed time and the decay rate. The larger is the decay rate, and the freshness function decreases faster.

$$ff(t) = ff(0)e^{-\lambda t} \tag{9}$$

3.2. *TSIM problem under DIC model*

In order to information Diffusion in the social network, different Diffusion models such as IC (Independent Cascade)[4], LT (Linear Threshold)[4], LAIC (Latency Aware Independent Cascade)[34], iLTM (improved Linear Threshold Model)[43] and DIC (Delay-based Independent Cascade)[5] have been proposed. In this paper, after identifying the most effective nodes, to identify the propagation value of these nodes, the DIC model[5] is used. In this model, as in the independent cascade propagation model, each active node can try to activate each of its neighbors only once. Each node can be active, latent active, or inactive at any one time. If the node *u* can activate the node *v* with probability p_{uv}, if d_{uv} is zero, the node *v* becomes active at the same instant; but if d_{uv} is greater than zero, the node *v* becomes hidden-active and switches to active mode after d_{uv}.

4. Proposed Time-Sensitive Centrality Approaches

As in the real world, the information propagation in social networks also has some delay. Therefore, it is important to detect the individuals that propagate information with a short delay. In the existing literature the algorithms that best fit the desired problem are the ones described by Mohammadi et al. [5]. Although, accuracy of their algorithms, time-sensitive degree centrality and time-sensitive Betweenness centrality are not high, their detection speed is high. The drawback of their methods is the local view, which only consider the direct neighbors of a node to determine its significance. To solve this issue, we first define the path value, and then use the path value to propose four time-sensitive centrality measures, that take into account both direct and indirect neighbors and time delay.

4.1. *Valuable path*

Finding the path between two nodes of a graph such that the number of edges in the path is smaller than all the other paths between these two nodes is called the shortest path problem. But the shortest path is not always the best path, especially when factors other than the path length are important.

In this paper, another type of path, called the valuable path, is introduced. In fact, the valuable path is a path between two nodes of a graph such that the value of information propagation from that path is higher than any other path between these two nodes.

Suppose there exists a set of paths $PATH^{u,v} = \{path_1^{u,v}, \ldots, path_L^{u,v}\}$ between the nodes u and *v* containing *L* paths. That $path_l^{u,v}$ is the *l*th path with length Q between two nodes u and *v* in the form $path_l^{u,v} = \langle e_1^l, \ldots, e_Q^l \rangle$. If $p_q^l \in [0.1]$ is the probability of affecting through the edge e_q^l, d_q^l is the propagation delay of the edge e_q^l and the value of the $ff_{l,q}(d_q^l)$ is freshness amount of the edge e_q^l, then the value of the l^{th} path between the two nodes *u* and *v* is calculated as follows:

$$PV_l^{u,v} = \sum_{q=1}^{Q} ff_{l,q}(d_q^l) * p_q^l \tag{10}$$

Finally, the path with the highest value is obtained in the following form,

$$path^{u,v} = \underset{path_l^{u,v} \in PATH^{u,v}}{argmin} \sum_{q=1}^{Q} d_q^l \tag{11}$$

Hence, first for each edge of the graph, the propagation delay, d_e is considered as the weight of the edge, then using Johnson's algorithm or Dijkstra's algorithm, the shortest path between each pair of nodes of the weighted graph is calculated.

4.2. Time-Sensitive Closeness (TSCloseness)

The Time-Sensitive Closeness method operates like the closeness centrality method with the difference that instead of using the shortest path, the most valuable path is used. In fact, the value of the u^{th} node in the proposed method is calculated using the total product of the propagation probability by the propagation value of each edge as in the Eq. (12).

$$CTSC(u) = \frac{1}{\sum_{v \neq u \epsilon V} p_{uv} * ff_{uv}(t)} \quad (12)$$

where p_{uv} is the influence node of node u on node v and $ff_{uv}(t)$ is the value of the freshness function with delay t from node u to node v. The shortcoming of this method is the weak results while encountering disconnected graphs. Algorithm of the proposed TSDegree criterion is shown in Algorithm 1. In this Algorithm, in lines 2 to 10, $CTSC(u)$ is computed for each node in the graph based on Eq. (12). In line 5, the path with the least propagation delay in the input weighted graph, in which the weight of each edge is the propagation delay between the nodes v and u is calculated using Dijkstra's method. Then, the propagation value between the two nodes v and u is calculated based on Eq. 10 in Line 6. Next, on the basis of the propagation value calculated between node v and each of the graph nodes, the time-sensitive closeness criteria is obtained based on the Eq. 12 in line 9. Then, in lines 11-14, k nodes with the minimum value of $CTSC(u)$ are selected.

Algorithm 1. TSCloseness

Input: *G(V,E)*: Social network; *k*: number of seeds
Output: *S*:seed set

1. $S \leftarrow \emptyset$
2. **for** $v \in V$ **do**
3. *pv(v)* $\leftarrow 0$
4. **for** u $\in$ V/{v} **do**
5. *path* ← Find shortest path between *v* and *u*
6. $PV_{path}^{u,v}$ ← compute propagation value between nodes *v* and *u* using path (via Eq. 10)
7. pv(v)+← $PV_{path}^{u,v}$
8. **end for**
9. FFs(v) ← 1/pv(v)
10. end for
11. **for** *i* ← *1* **to** *k* **do**
12. $u \leftarrow \text{argmin}_{v \in V \setminus S} FFs(v)$
13. $S \leftarrow S \cup \{u\}$
14. end for
15. **return** *S*

Theorem 1: The time complexity of the TSCloseness algorithm is $O(mn + n^2 logn)$ in which n and m refer to number of nodes, and number of edges.

Proof: The Dijkstra's algorithm is used to find the shortest path, which has the time complexity of $O(mn + n^2 logn)$. From Line 2 to Line 10 of Algorithm 1 represent the computation of decay centrality metric which the time complexity is $O(n^2)$. After that, k influential nodes are selected (see Lines 11–14 of Algorithm 1) which time complexity is equal to $O(kn)$. Accordingly, the total complexity of this algorithm is $O(mn + n^2 logn + n^2 + kn) \in O(mn + n^2 logn)$. □

4.3. Time-Sensitive Harmonic (TSHarmonic)

The Time-Sensitive Harmonic method is a modified version of the proposed Time-Sensitive Closeness method, with the difference that instead of using the inverse total of the product of the propagation probability by the propagation value of the most valuable path from node u to other nodes, it uses the total inverse of the product of the propagation probability by the propagation value of the most valuable path from node u to other nodes as in Eq. (13).

$$CTSH(u) = \sum_{v \neq u \epsilon V} \frac{1}{p_{uv} * ff_{uv}(t)} \quad (13)$$

Also, if there is no path between nodes u and v, then $\frac{1}{p_{uv} * ff_{uv}(t)} = 0$. Algorithm of the proposed TSHarmonic criterion is shown in Algorithm 2. In Algorithm 2, $CTSH(u)$ is computed for each node in the graph based on Eq. 13 in lines 2 to 9. In line 5, the path with the least propagation delay between the two nodes v and u is calculated using Dijkstra's method. Then, in Line 6, the propagation value between the two nodes v and u is calculated based on Eq. 10. Next, based on the calculated propagation value between node v and each of the graph nodes, the time-sensitive harmonic criteria is obtained based on Eq. 13 in line 7. Then, in lines 10-13, k nodes with the minimum value of $CTSH(u)$ are selected.

Algorithm 2. TSHarmonic

Input: *G(V,E)*: Social network; *k*:number of seeds
Output: *S*:seed set

1. $S \leftarrow \emptyset$
2. **for** $v \in V$ **do**
3. *FFs(v)* $\leftarrow 0$
4. **for** $u \in V/\{v\}$ **do**
5. *path* ← Find shortest path between *v* and *u*
6. $PV_{path}^{u,v}$ ← compute propagation value between nodes *v* and *u* using path (via Eq. 10) if path is exist
7. $FFs(v) + \leftarrow 1/PV_{path}^{u,v}$
8. **end for**
9. end for
10. **for** *i* ← 1 **to** *k* **do**
11. $u \leftarrow \text{argmin}_{v \in V \setminus S} FFs(v)$
12. $S \leftarrow S \cup \{u\}$
13. end for
14. **return** *S*

Theorem 2: The time complexity of the TSHarmonic algorithm is $O(mn + n^2 logn)$ in which n and m refer to number of nodes, and number of edges.

Proof: The Dijkstra's algorithm is used to find the shortest path, which has the time complexity of $O(mn + n^2 logn)$. From Line 2 to Line 9 of Algorithm 2 represent the computation of decay centrality metric which the time complexity is $O(n^2)$. After that, k

influential nodes are selected (see Lines 10–13 of Algorithm 2) which time complexity is equal to $O(kn)$. Accordingly, the total complexity of this algorithm is $O(mn + n^2 logn + n^2 + kn) \in O(mn + n^2 logn)$. □

4.4. *Time-Sensitive Decay (TSDecay)*

Similar to the Time-Sensitive Closeness method, The Time-Sensitive Decay method employs the most valuable path such that the product of the propagation probability by the propagation value of the most valuable path is used to detect the importance of a node as given in the Eq. (14).

$$CTSD(u) = \sum_{j \neq i \in V} \delta^{p_{uv} * ff_{uv}(t)} \quad (14)$$

where $0 < \delta < 1$. Algorithm of the proposed TSDecay criterion is shown in Algorithm 3.

Algorithm 3. TSDecay

Input: *G(V,E)*: Social network; *k*: number of seeds
Output: *S*:seed set

1. $S \leftarrow \emptyset$
2. **for** $v \in V$ **do**
3. $FFs(v) \leftarrow 0$
4.   **for** $u \in V/\{v\}$ **do**
5.   *path*←Find shortest path between *v* and *u*
6. $PV_{path}^{u,v}$←compute propagation value between nodes *v* and *u* using path (via Eq. 10) if path is exist
7. $FFs(v) +\leftarrow \delta^{PV_{path}^{u,v}}$
8.   **end for**
9. end for
10. **for** $i \leftarrow 1$ **to** *k* **do**
11. $u \leftarrow \text{argmin}_{v \in V \setminus S} FFs(v)$
12. $S \leftarrow S \cup \{u\}$
13. end for
14. **return** *S*

In lines 2 to 9 of Algorithm 3, $CTSD(u)$ is computed for each node in the graph based on Eq. (14). In line 5, the path with the least propagation delay between the two nodes *v* and *u* is calculated using Dijkstra's method. Then, in Line 6, the propagation value between the two nodes *v* and *u* is calculated based on Eq. 10. Next, based on the calculated propagation value between node *v* and each of the graph nodes, the time-sensitive decay criteria is obtained based on Eq. 14 in line 7. Finally, in lines 10-13, *k* nodes with the minimum value of $CTSD(u)$ are selected.

Theorem 3: The time complexity of the TSDecay algorithm is $O(mn + n^2 logn)$ in which n and m refer to number of nodes, and number of edges.

Proof: The Dijkstra's algorithm is used to find the shortest path, which has the time complexity of $O(mn + n^2 logn)$. From Line 2 to Line 9 of Algorithm 3 represent the computation of decay centrality metric which the time complexity is $O(n^2)$. After that, *k* influential nodes are selected (see Lines 10–13 of Algorithm 3) which time complexity is equal to $O(kn)$. Accordingly, the total complexity of this algorithm is $O(mn + n^2 logn + n^2 + kn) \in O(mn + n^2 logn)$. □

4.5. *Time-Sensitive Eccentricity (TSEccentricity)*

The Time-Sensitive Eccentricity method first identifies the valuable paths from one node to other nodes, and then calculates the product of the propagation probability by the propagation value. Finally, the node which has the path with the maximum product of the propagation probability by the propagation value is selected as the most significant node as in the Eq. (15).

$$CTSE(u) = max_{v \in V} p_{uv} * ff_{uv}(t) \quad (15)$$

Algorithm 4 shows the identification of high-performance nodes by this criterion. $CTSE(u)$ is computed for each node in the graph based on Eq. 15 in lines 2 to 12. In line 6, the path with the least propagation delay between the two nodes *v* and *u* is calculated using Dijkstra's method. Then, in Line 7, the propagation value between the two nodes *v* and *u* is calculated according to Eq. 10. Next, based on the calculated propagation value between node *v* and each of the graph nodes, the time-sensitive eccentricity criteria is obtained based on Eq. 15 in line 9. Then, in lines 13-16, *k* nodes with the minimum value of $CTSE(u)$ are selected.

Algorithm 4. TSEccentricity

Input: *G(V,E)*: Social network; *k*:number of seeds
Output: *S*:seed set

1. $S \leftarrow \emptyset$
2. **for** $v \in V$ **do**
3. $pv(v) \leftarrow 0$
4. $FFs(v) \leftarrow 0$
5.   **for** $u \in V/\{v\}$ **do**
6.   *path*←Find shortest path between *v* and *u*
7. $PV_{path}^{u,v}$←compute propagation value between nodes *v* and *u* using path (via Eq. 10) if path is exist
8.   **if** $PV_{path}^{u,v} > FFs(v)$ **then**
9. $FFs(v) \leftarrow PV_{path}^{u,v}$
10.   **end if**
11.   **end for**
12. end for
13. **for** $i \leftarrow 1$ **to** *k* **do**
14. $u \leftarrow \text{argmax}_{v \in V \setminus S} FFs(v)$
15. $S \leftarrow S \cup \{u\}$
16. end for
17. **return** *S*

Theorem 4: The time complexity of the TSEccentricity algorithm is $O(mn + n^2 logn)$ in which n and m refer to number of nodes, and number of edges.

Proof: The Dijkstra's algorithm is used to find the shortest path, which has the time complexity of $O(mn + n^2 logn)$. From Line 2 to Line 13 of Algorithm 4 represent the computation of decay centrality metric which the time complexity is $O(n^2)$. After that, *k* influential nodes are selected (see Lines 14–17 of Algorithm 4) which time complexity is equal to $O(kn)$. Accordingly, the total complexity of this algorithm is $O(mn + n^2 logn + n^2 + kn) \in O(mn + n^2 logn)$. □

5. Simulation Results

As discussed in this paper, using centrality measures to detect the influential nodes in a social network is a heuristic approach that is used instead of greedy approaches, and its running time is better than the greedy algorithms. The propagation delay is a hidden and effective parameter in the influence and inspiration of the nodes of a social network. In this paper, considering this parameter, first the concept of valuable path is expressed and then based on it, Time-Sensitive Closeness, Time-

Sensitive Harmonic, Time-Sensitive Decay, and Time-Sensitive Eccentricity measures It was introduced, which are evaluated in the following.

As shown in[5], the Time-Sensitive Degree has higher accuracy compared to Time-Sensitive Betweenness. Thus, in this paper, the proposed measures are compared with the Time-Sensitive Degree method. Table I represents the names of the methods evaluated in this Section.

Table I. Naming of the evaluated methods

Method	Description
Deg	Degree centrality [36]
TSDeg	Time-sensitive degree centrality [2]
TSCloseness	Time-sensitive closeness centrality (sub-section 4.2)
TSHarmonic	Time-sensitive harmonic centrality (sub-section 4.3)
TSDecay	Time-sensitive decay centrality (sub-section 4.4)
TSEccentricity	Time-sensitive eccentricity centrality (sub-section 4.5)

5.1. *Experimental setups*

In this section, used datasets and configurations of the conducted experiments are defined.

5.1.1. Datasets

The proposed centrality methods are evaluated using five real datasets given in Table II.

Dolphine dataset is a directed social network of bottlenose dolphins. The nodes are the bottlenose dolphins and edges indicate frequent association[44]. HighSchool dataset is directed network contains friendships between boys in a small high school in Illinois[44]. Florida ecosystem wet contains the carbon exchanges in the cypress wetlands of South Florida during the wet season. Nodes represent taxa and an edge denotes that a taxon uses another taxon as food with a given trophic factor[44]. FilmTrust trust dataset is the user–user trust network of the FilmTrust project[44]. Human proteins dataset is a network of interactions between proteins in Humans (Homo sapiens), from the first large-scale study of protein–protein interactions in Human cells using a mass spectrometry-based approach[44]. Wiki-vote dataset is directed/unweighted network from [45].

Table II. Datasets

Datasets	Nodes	Edges
Dolphin	62	159
HighSchool	70	366
Floryda Ecosystem Wet (Floryda)	128	2106
Film Trust	874	1853
Human proteins (Figeys)	2239	6452
Wiki-vote	7115	103689

5.1.2. Configuration

The propagation value is calculated using the freshness function, $ff(t) = ff(0)e^{-\lambda t}$, presented in[5]; by increasing the value of λ, the decay rate also increases. Thus, to change the information decay rate, two values of 2 and 0.2 are considered for λ.

The influence probability of each edge is calculated using $p_{u,v} = \frac{1}{|N^{in}(v)|}$, such that $|N^{in}(v)|$ is the input degree of node v; the propagation delay of each edge is a random number in the range of [0-20]. To specify the total propagation value of the k identified nodes using any of the proposed centrality measures, the delayed independent cascade propagation method presented in[5] is used; finally, to increase the accuracy of the results, 10000 Monte-Carlo iterations are used.

5.2. *Experimental Results*

As mentioned in Section 3.1, the information propagation delay among people of a society or a social network plays an important role in information propagation among people that can change the individuals' influence. The higher is the decay rate of the information, the influence of the information propagation delay would also be greater. So, in this Section, the freshness function with two different decay rates is studied, and their results are compared. The propagation value of a seed set is a measure of its quality. In fact, it is more desired that a seed set (k selected nodes) generates a higher propagation value. To this end, the delay-sensitive, independent propagation value is estimated for initial seed sets of different sizes.

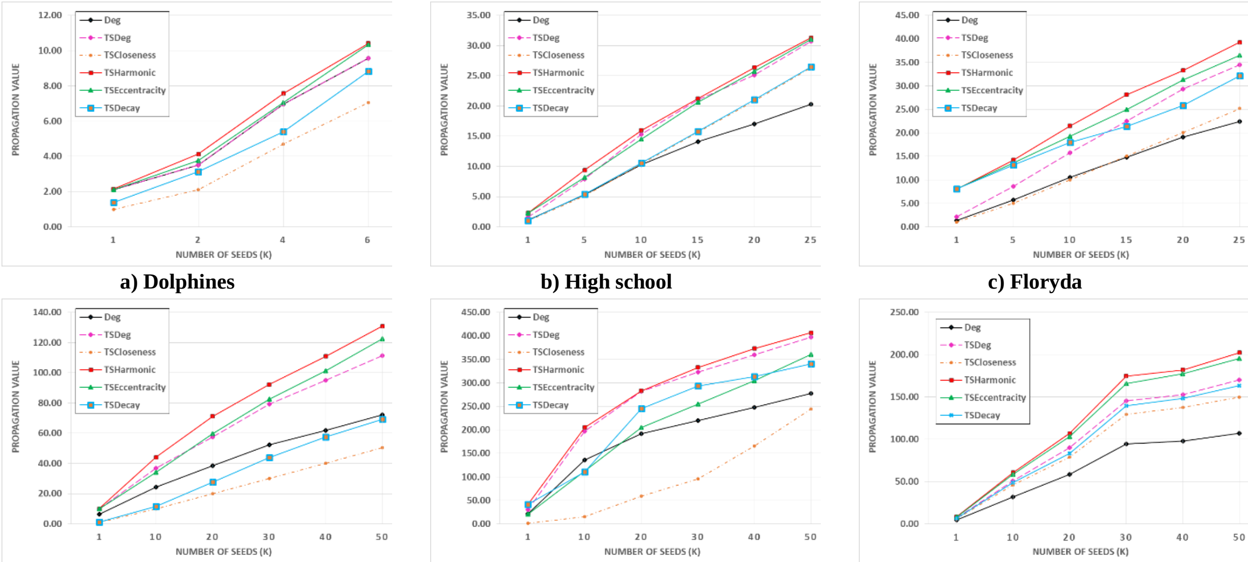
5.2.1. Propagation Value

In this sub-Section, the propagation value measure of different methods is studied. First, the performance of the proposed centrality measures for the freshness function of $ff(t) = ff(0)e^{-0.2t}$ is examined, where the initial value of the function at 0s is 1, and as time passes, the information value decreases with a decay rate of 0.2. Fig. 1 shows the propagation value for the five datasets; the horizontal axis represents the number of seed sets (k), and the vertical axis represents the propagation value.

According to the results given in Fig. 1, first, for all measures, as the number of seed sets increases, the propagation value also increases. Second, in all of the examined datasets, the TSHarmonic measure has achieved the maximum propagation value such that, for example, in the FilmTrust dataset, the propagation value has increased about 23% compared to the TSDegree measure. The TSEccentricity measure outperformed the TSDegree in most datasets.

In the following, the decay rate is increased, and the performance of the centrality measures for the freshness function of $ff(t) = ff(0)e^{-2t}$ is examined. The initial value of the function at 0s is 1, and as time passes, the information value decreases with a decay rate of 2. The propagation value for the five introduced datasets is shown in Fig. 2. Where the horizontal axis represents the number of seed sets, and the vertical axis represents the propagation value.

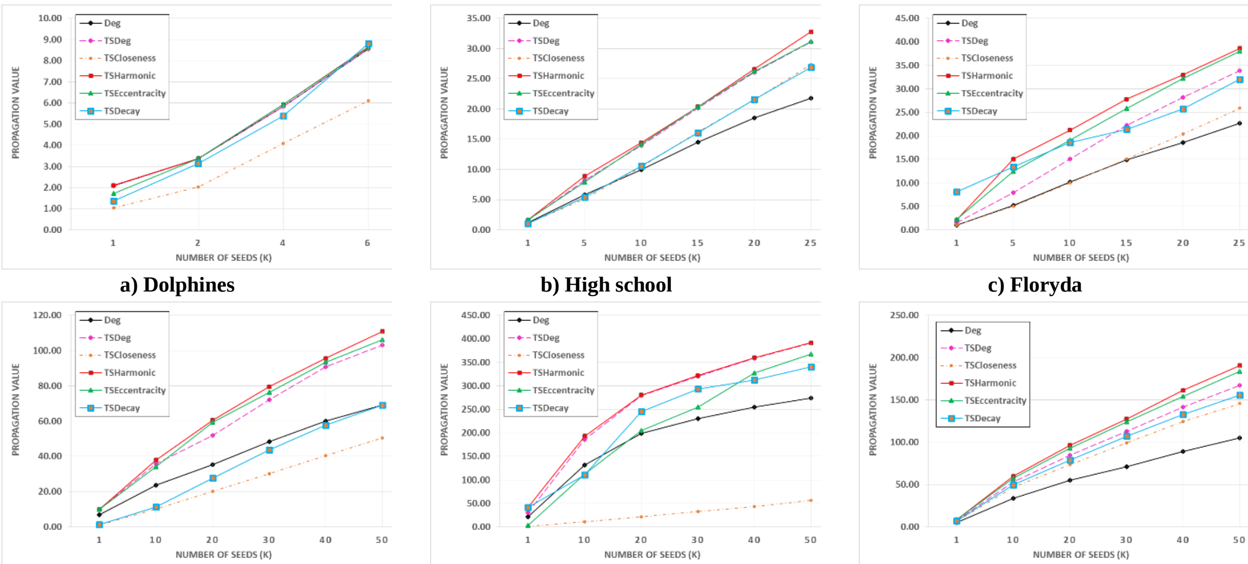
As shown in Fig. 2, as the information decay rate increases, the distance of the diagrams in most datasets decreases. Because as the decay rate increases, the information that reaches the user with more intermediates, its value is lower. Thus, the neighbors with fewer intermediates are more significant than the neighbors with more intermediates. As the decay rate increases, the TSHarmonic measure obtains a higher value compared to the other measures, such that its propagation value is 16% higher than the TSDegree. Due to the fact that the weaknesses of other criteria in the TSHarmonic criterion have been eliminated, it has shown higher accuracy in the simulation results. For example, considering the global view to the node's position in the graph, all of the direct and indirect neighbors of each node affect its ranking in the TSHarmonic measure. But in local methods like TSDegree, only the direct neighbors are influential.



a) Dolphines b) High school c) Floryda

d) Film Trust e) Human proteins f) Wiki-vote

Fig. 1. The propagation value of different methods, independent cascade propagation model with the freshness function of $ff(t) = e^{-0.2t}$.



a) Dolphines b) High school c) Floryda

d) Film Trust e) Human proteins f) Wiki-vote

Fig. 2. The propagation value for different methods, independent cascade propagation model for the freshness function of $ff(t) = e^{-2t}$.

Thus, considering the global position of each node in the network, its accuracy is higher than the degree-based method.Since in the TSEccentricity method, unlike TSHarmonic method, only the most valuable path is considered for each node (only one path), and it might have paths of lower value compared to other nodes. Thus, its accuracy would not be proper.

The TSEccentricity criterion is also close to TSHarmonic, but since the criterion for ranking graph nodes in TSEccentricity is only the most valuable path of each node to other nodes, a node like u may have a more valuable path than node v. But the sum of the path values of node u is less than v, which is not taken into account in TSEccentricity; So, in theory it is a weakness that reduces accuracy. In the simulated results, this decrease in accuracy compared to TSHarmonic can be seen.

5.2.2. Computational complexity

In this sub-Section, the computational complexity and running time of the measures presented in this study are examined. The running time of various methods on different datasets for k=20 and freshness function of $ff(t) = ff(0)e^{-0.2t}$ is represented in Table III.

Since all methods, unlike the TSDegree method, should examine all paths of each node to its direct and indirect neighbors, which takes more time. The running time of TSHarmonic and TSDecay methods is similar and shorter than TSCloseness.

Since all measures introduced in this paper are evolved versions of basic measures like Degree, Closeness, Decay, Harmonic, and Eccentricity, and some additional calculations are added to calculate the propagation path that does not affect the time complexity of the methods. Therefore, the time complexity of these new measures is the same as the basic measures, which are given in Theorem 1-4.

Table III. Running time of various measures on a different network in seconds ($ff(t) = e^{-0.2t}$, k=20)

Datasets\Methods	Deg	TSDeg	TSCloseness	TSHarmonic	TSDecay	TSEccentricity
HighSchool	0.00	0.00	2.53	1.38	1.58	1.70
Floryda	0.00	0.00	22.52	17.72	19.92	14.32
Film Trust	0.00	0.01	748.18	523.57	521.29	279.26
Human Proteins	0.00	0.03	278.16	232.06	230.24	25.20
Wiki-vote	0.12	0.28	3095.85	1869.01	2378.65	462.37

5.2.3. Effect of freshness function

The impact of the freshness function on the TSIM problem is examined. Therefore, the influence propagation of TSHarmonic under the DIC model on different datasets with three different freshness functions is evaluated. $ff(t) = 1$, $ff(t) = e^{-0.2t}$ and $ff(t) = e^{-2t}$ are used as freshness functions.

As mentioned, when the freshness function decreases faster, the total spread value will decrease more. For example, the propagation value of TSHarmonic on Human proteins dataset with k=50 is equal to 130.93, 115.87, and 110.98, for $ff(t) = 1$, $ff(t) = e^{-0.2t}$ and $ff(t) = e^{-2t}$, respectively. As the outcomes show, the highest propagation value is obtained when the freshness function is constant. Indeed, when $ff(t) = 1$ is used as a freshness function, there is no decay in the information value and all the activated nodes have a value of 1. However, in $ff(t) = e^{-0.2t}$ and $ff(t) = e^{-2t}$ the value of activated nodes depends on their activation time and decreases over time. The decay rate in $ff(t) = e^{-0.2t}$ is lower than $ff(t) = e^{-2t}$.

Furthermore, the overlaps of seed sets returned by TSHarmonic for different freshness functions are evaluated. Table IV shows the results for k=50 in two datasets. Each entry within the table shows the quantity of common seeds returned by TSHarmonic for the two freshness functions in corresponding columns and rows.

Table IV. The overlaps of seed sets returned by TSHarmonic with different freshness function when k=50.

	freshness function	$ff(t) = 1$	$ff(t) = e^{-0.2t}$	$ff(t) = e^{-2t}$
Human proteins	$ff(t) = 1$	50	40	28
	$ff(t) = e^{-0.2t}$		50	32
	$ff(t) = e^{-2t}$			50
Film Trust	$ff(t) = 1$	50	32	14
	$ff(t) = e^{-0.2t}$		50	24
	$ff(t) = e^{-2t}$			50

According to the results, the seed sets that maximizes the propagation value, differs significantly for different freshness functions. For example, the set of nodes that maximizes the propagation value of a given freshness function does not necessarily maximize the propagation values of different decay amounts. This result represents the time sensitivity of propagation value has an important role in the problem of the influence maximization.

6. Conclusion

In this paper, we investigate the problem of influence maximization with time-sensitive. First, we define a time-sensitive influence maximization problem. Second, we define propagation value for each edge between two graph nodes according to the freshness function and the probability of affecting one node on another node. Third, we propose four centrality metric to estimate the most influential nodes. In TSDecay, we propose an exponential function based on the sum of the most valuable paths of each node as a measure of centrality. We propose TSCloseness method which detects the proximity of each node in the graph based on the most valuable path to the other nodes, but in cases where there is no path between the two nodes, the standard performance decreases. In TSEccentricity, the Maximum value path of each node is used as a criterion for identifying the centrality of the nodes, but due to the fact that for each node is only one measurement path, it is possible to compare two nodes, one of the two It has a larger value path but less total value paths, which also reduces standard accuracy. In TSHarmonic, which is an enhanced TSCloseness, the sum of the value paths of each node to the other nodes is used as a measure of the centrality of each node. Experimental results based on five datasets demonstrate the performance of our algorithms compared to baseline models. Among the four metrics introduced, the time-sensitive harmonic measurement has achieved higher accuracy compared to other metrics, such that its accuracy is increased by about 23% compared to the time-sensitive degree metric. Because of the improvement in accuracy and time complexity, the time-sensitive harmonic can be used as an appropriate time-sensitive centrality measure.

7. References

[1] H. Huang, H. Shen, Z. Meng, H. Chang, H. He, “Community-based influence maximization for viral marketing”, *Applied Intelligence*, vol. 49, pp. 2137-2150, 2018.

[2] C. Budak, D. Agrawal, A. El Abbadi, “Limiting the spread of misinformation in social networks”, In the Proceedings of the 20th international conference on World Wide Web, 2011, pp. 665-674.

[3] M. A. Manouchehri, M. S. Helfroush, H. Danyali, “A Theoretically Guaranteed Approach to Efficiently Block the Influence of Misinformation in Social Networks”, *IEEE Transactions on Computational Social Systems*, vol. 8, pp. 716-727, 2021.

[4] D. Kempe, J. Kleinberg, É. Tardos, “Maximizing the spread of influence through a social network”, In the Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, Washington, DC, USA, pp. 137-146.

[5] A. Mohammadi, M. Saraee, A. Mirzaei, “Time-sensitive influence maximization in social networks”, *Journal of Information Science*, vol. 41, no. 6, pp. 765-778, 2015.

[6] M. Adineh, M. Nouri-Baygi, “High Quality Degree Based Heuristics for the Influence Maximization Problem”, *arXiv preprint arXiv:1904.12164,* 2019.

[7] P. Domingos, M. Richardson, “Mining the network value of customers”, In the Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining, 2001, San Francisco, California, pp. 57-66.

[8] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, “Cost-effective outbreak detection in networks”, In the Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, San Jose, California, USA, pp. 420-429.

[9] A. Goyal, W. Lu, L.V.S. Lakshmanan, “CELF++: optimizing the greedy algorithm for influence maximization in social networks”, In the Proceedings of the 20th international conference companion on World wide web, 2011, Hyderabad, India, pp. 47-48.

[10] Y. Wang, G. Cong, G. Song, K. Xie, “Community-based greedy algorithm for mining top-K influential nodes in mobile social networks”, In the Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010, Washington, DC, USA, pp. 1039-1048.

[11] K. Jung, W. Heo, and W. Chen, “IRIE: Scalable and Robust Influence Maximization in Social Networks”, In 2010 IEEE 12th International Conference on Data Mining, 2012, Brussels, Belgium, pp. 918-923,.

[12] A. Sheikhahmadi, M. A. Nematbakhsh, A. Zareie, “Identification of influential users by neighbors in online social networks”, *Physica A: Statistical Mechanics and its Applications*, vol. 486, pp. 517-534, 2017.

[13] W. Chen, C. Wang, Y. Wang, “Scalable influence maximization for prevalent viral marketing in large-scale social networks”, In the Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010, Washington, DC, USA, pp. 1029-1038.

[14] W. Chen, Y. Yuan, L. Zhang, “Scalable Influence Maximization in Social Networks under the Linear Threshold Model”, In 2010 IEEE International Conference on Data Mining, 2010, Sydney, Australia, pp. 88-97.

[15] A. Zareie, A. Sheikhahmadi, M. Jalili, “Identification of influential users in social network using gray wolf optimization algorithm”, *Expert Systems with Applications*, vol. 142, 2020.

[16] A. Zareie, A. Sheikhahmadi, M. Jalili, M.S.K. Fasaei, “Finding influential nodes in social networks based on neighborhood correlation coefficient”, *Knowledge-based systems*, vol. 194, 2020.

[17] Ch. Salavati, A. Abdollahpouri, Zh. Manbari, “A Multi-objective Algorithm for Identifying Influential Nodes in Social Networks”, *Tabriz Journal of Electrical Engineering*, vol. 50, no. 3, pp. 1293-1304, 2020 (in persian).

[18] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, “Identification of influential spreaders in complex networks”, *Nature physics*, vol. 6, no. 11, pp. 888-893, 2010.

[19] W. Chen, Y. Wang, S. Yang, “Efficient influence maximization in social networks”, In the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, Paris, pp. 199-208.

[20] X. Wang, Y. Su, C. Zhao, D. Yi, “Effective identification of multiple influential spreaders by DegreePunishment”, *Physica A: Statistical Mechanics and its Applications*, vol. 461, pp. 238-247, 2016.

[21] A. Sheikhahmadi, M. A. Nematbakhsh, and A. Shokrollahi, “Improving detection of influential nodes in complex networks”, *Physica A: Statistical Mechanics and its Applications*, vol. 436, pp. 833-845, 2015.

[22] A. Zareie, A. Sheikhahmadi, K. Khamforoosh, “Influence maximization in social networks based on TOPSIS”, *Expert Systems with Applications*, vol. 108, pp. 96-107, 2018.

[23] L. Katz, “A new status index derived from sociometric analysis”, *Psychometrika*, vol. 18, no. 1, pp. 39-43, 1953.

[24] A. Zareie, A. Sheikhahmadi, and R. Sakellariou, “A composite centrality measure for improved identification of influential users”, *arXiv preprint arXiv:2111.04529*, 2021.

[25] K. Saito, M. Kimura, K. Ohara, H. Motoda, “Learning continuous-time information diffusion model for social behavioral data analysis”, In Asian Conference on Machine Learning, 2009, Berlin, Heidelberg, pp. 322-337.

[26] W. Chen, W. Lu, N. Zhang, “Time-critical influence maximization in social networks with time-delayed diffusion process”, In the Proceedings of the 26th AAAI Conference on Artificial Intelligence, 2012, Toronto, Ontario, Canada.

[27] J. Kim, W. Lee, H. Yu, “CT-IC: Continuously activated and time-restricted independent cascade model for viral marketing”, *Knowledge-Based Systems*, vol. 62, pp. 57-68, 2014.

[28] R. Yan, Y. Li, D. Li, Y. Zhu, Y. Wang, H. Du, “Activation probability maximization for target users under influence decay model”, in International Computing and Combinatorics Conference, 2019, Springer, pp. 603-614.

[29] N. Ohsaka, Y. Yamaguchi, N. Kakimura, K.I. Kawarabayashi, “Maximizing time-decaying influence in social networks”, in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2016, Springer, Cham, pp. 132-147.

[30] M. Hu, Q. Liu, H. Huang, X. Jia, “Time-sensitive influence maximization in social networks”, in 18th International Conference on Communication Technology (ICCT), 2018, IEEE, pp. 1351-1356.

[31] M. Huiyu, C. Jiuxin, Y. Tangfei, B. Liu, “Topic based time-sensitive influence maximization in online social networks”, *World Wide Web*, vol. 23, no. 3, pp. 1831-1859, 2020.

[32] Y. Wang, Y. Zhang, F. Yang, D. Li, X. Sun, J. Ma, “Time-sensitive positive influence maximization in signed social networks”, *Physica A: Statistical Mechanics and its Applications*, vol. 584, pp. 126353, 2021.

[33] A. Goyal, F. Bonchi, L.V.S. Lakshmanan, "Learning influence probabilities in social networks", In the Proceedings of the 3rd ACM international conference on web search and data mining, 2010, New York, USA, pp. 241-250.

[34] B. Liu, G. Cong, D. Xu, Y. Zeng, "Time constrained influence maximization in social networks", in 12th international conference on data mining, 2012, IEEE, Brussels, Belgium, pp. 439-448.

[35] B. Liu, G. Cong, Y. Zeng, D. Xu, Y.M. Chee, "Influence spreading path and its application to the time constrained social influence maximization problem and beyond", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1904-1917, 2013.

[36] L.C. Freeman, "Centrality in social networks conceptual clarification", *Social Networks*, vol. 1, no. 3, pp. 215-239, 1978.

[37] L.C. Freeman, "A set of measures of centrality based on Betweenness", *Sociometry,* pp. 35-41, 1977.

[38] M.O. Jackson, "Social and economic networks", Princeton university press, 2010.

[39] P. Crescenzi, G. d'Angelo, L. Severini, Y. Velaj, "Greedily improving our own centrality in a network", in International Symposium on Experimental Algorithms, 2015, Springer, pp. 43-55.

[40] U. Brandes, "A faster algorithm for Betweenness centrality", J*ournal of mathematical sociology*, vol. 25, no. 2, pp. 163-177, 2001.

[41] A. Raychaudhuri, S. Mallick, A. Sircar, S. Singh, "Identifying Influential Nodes Based on Network Topology: A Comparative Study", in Information, Photonics and Communication, 2020, Springer, pp. 65-76.

[42] P. Jia, J. Liu, C. Huang, L. Liu, C. Xu, "An improvement method for degree and its extending centralities in directed networks", *Physica A: Statistical Mechanics and its Applications,* vol. 532, pp. 121891, 2019.

[43] S. Babaei S. Molaei M. Salehi, "Modeling Information Diffusion in Bibliographic Multilayer Networks", *Tabriz Journal of Electrical Engineering,* vol. 49, no. 2, pp. 503-515, 2019 (in persian).

[44] J. Kunegis, "Konect: the koblenz network collection", in Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 1343-1350.

[45] J. Leskovec, D. Huttenlocher, J. Kleinberg, "Predicting Positive and Negative Links in Online Social Networks", In Proceedings of the 19th international conference on World Wide Web, 2010, North Carolina, USA, pp. 641-650.