Applied Intelligence
https://doi.org/10.1007/s10489-020-01685-5

A new irregular cellular learning automata-based evolutionary computation for time series link prediction in social networks

Mozhdeh Khaksar Manshad[1] · Mohammad Reza Meybodi[2] · Afshin Salajegheh[1]

Abstract
Link prediction (LP), as an attempt to predict event-based future connections within a network, is the main task of social network analysis (SNA). Accordingly, common LP approaches to forecast future connections utilize similarity metrics of non-connected links in a static network representation. A general shortcoming of most existing research studies in this field is that they tap the present condition of a system and fail to take any temporal events into account. Moreover; social networks are innately evolutionary since they are assumed to be online, non-deterministic, and unforeseeable in most applications. Consequently, it is not appropriate to employ deterministic models for examining actual social network problems. With regard to time-series LP (TSLP) problems, temporal evolution of connection incidence is correspondingly exploited to predict connection chances at a particular time. In this paper, a new TSLP method based on irregular cellular learning automaton (ICLA) and evolutionary computation (EC) is proposed. In the evolutionary procedure suggested here, each vertex (i.e. cell) includes a genome as well as a set of learning automata (LAs). Accordingly, the genome residing in a cell represents predicted links for the corresponding cell. Local information among cells in successive time 1 to T in the network is then analyzed to predict future connections in time $T+1$. According to the distributed feature of the recommended approach, each genome is locally developed by a local search. The experiments in this study via e-mail and co-authorship networks ultimately show that the proposed algorithm leads to remarkable outcomes in predicting future connections.

Keywords Time-series link prediction · Irregular cellular learning automata · Evolutionary computation

1 Introduction

Since links among individuals on cyberspace are growing daily, useful information is being made available for analyzing social networks. In this respect; link prediction (LP) has been recognized as one of the leading topics in studies of social network analysis (SNA), which can be explained as the evaluation of a possible connection between two separate individuals in the future [1–5]. Typically, such possibilities can be measured through similarities or relative ranks between node pairs. The existing approaches to LP are also classified into two categories of (1) similarity-based and (2) learning-based approaches.

The similarity-based approaches are also grouped into three sub-categories including (1) node-based [6–8], (2) topology-based [9], and (3) social network criteria-based similarity methods [10–14]. The node-based similarity metrics calculate similarity of two nodes according to actions and interests they share [7]. Additionally, the topology-based similarity metrics analyze similarity of two nodes using structural features of a social network such as the shortest path length or the number of common neighborhoods between two nodes [9]. Lastly, the social network criteria-based similarity approaches employ notions of social networks such as community and clustering factor to measure degree of similarity among nodes [10, 11].

Tapping the features provided by internal attributes, similarity-based metrics, and external information; another group of methods called learning-based similarity methods

✉ Mozhdeh Khaksar Manshad
st_m_khaksarmenshad@azad.ac.ir

Mohammad Reza Meybodi
mmeybodi@aut.ac.ir

Afshin Salajegheh
a_salajegheh@azad.ac.ir

[1] Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

[2] Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

Published online: 16 April 2020

has been lately introduced which can be classified into four groups: (1) feature-based classification methods [15–18], (2) probabilistic graph-based models [19, 20], (3) matrix factorization [21], and (4) learning automaton (LA)-based models [22–24].

The model presented in this paper is a learning-based approach that utilizes irregular cellular LA (ICLA) and evolutionary computation (EC), shortened as ICLA-EC.

In accordance with artificial intelligence, an evolutionary algorithm (EA) is a sub-group of ECs, which is known as a meta-heuristic optimization process based on a population. Using a population of candidate solutions, EAs aim to find optimal ones. This population is thus assessed and the best solutions are chosen to mate and to form the next generations. Among several generations, there is also a number of proper trait augments, leading to a growth in quality of solutions. Besides, an EA can be considered as a parallel one. In the majority of parallel EAs (PEAs), tasks are classified into different parts which are then simultaneously solved using various processors. Single-population fine-grained is a kind of PEAs that includes just one population, while containing a spatial structure restricting relations among individuals [25–27]. In this kind of PEA, an individual can compete and mate with its neighbors. However, appropriate results may spread over the whole population, since neighborhoods overlap. In addition, it is common to put individuals of a PEA into a 1, 2, or 3-dimensional regular grid. In [25], it has been accordingly reported that such a system can be modeled via cellular automata (CAs) with stochastic rules. In a CA model, as a non-linear dynamic one, time and space are discrete. Each cell in the CA model can also collect partial information about others and the whole system. Moreover, the cell must decide on the state changing by employing limited information; therefore, uncertainty is entered into decisions. Learning in decisions can also get over the mentioned uncertainty [28]. An adaptive decision-making algorithm i.e. LA can be further used to learn optimal actions among a group of authorized ones via interactions with a random environment [29]. To improve local interactions of LA in a complicated system, CA and LA algorithms are therefore merged and introduced as a cellular learning automaton (CLA) [30]. The CLA is also superior to a CA, since it attempts to learn optimal actions and enhances learning processes via a collection of LAs that interact together. The elimination of the regular structure of the CLA also leads to a new model labeled as irregular CLA (ICLA).

The combination of the CLA model and the EAs also results in a model called CLA-based EC (CLA-EC) [31]. Such an algorithm can execute searches in large, complicated, and multimodal environments.

In this research, a TSLP model, called ICLA-based EC for TSLP (ICLA-EC-TSLP), is proposed. The ICLA-EC model is a mixture of ICLA [32] and the evolutionary model [31]. In this algorithm, the entire network is modeled as an ICLA, so that each vertex (i.e. cell) is equipped with a cell of ICLA, and there is a set of LAs in each vertex. For each vertex, the probability of any future connection with a related one is analyzed by each LA. A time sequence from *1* to *T* is considered to predict connection incidence in time $T+1$. Each LA in each cell contains two actions to decide about the presence of the corresponding connection at time $T+1$. Therefore, the ICLA-EC passes through time *1* to *T* to examine the presence of corresponding connections. In each step, under a local rule, a reinforcement signal vector is developed to feed the group of LAs residing in a cell. Based on a learning procedure, each LA in the cell can update the probabilities of its actions using the received signal. The environment of time *t* also form the group of the required predictions for time $t+1$, computed using local information of each cell at time *t*. The explained steps are also repeated up to the time once a termination condition is met. As the algorithm terminates, the genome in each cell represents predicted links for a corresponding cell. The main idea behind using the ICLA-EC in this work is that it utilizes local interactions and learns optimal behaviors to adapt to unknown environments.

According to empirical achievements, ICLA-EC-TSLP surpasses other static LP techniques including Jaccard similarity index (JC) [33], common neighbors (CN) [3], Adamic/Adar index (AA) [34], preferential attachment (PA) [35], quasi-local LP methods like local path [36], global LP methods like Katz index [37], as well as the latest LP techniques such as LA-TSLP [22], covariance matrix adaptive evolution strategy (CMA-ES) [15], interaction prediction (IP) [18], mutual information of network structure LP (MI-LP) [17], autoregressive integrated moving average LP (ARIMA-LP) [16], fuzzy LP based on distributed LA (FLP-DLA) [24], ant colony LP (ACO-LP) [38], and vector auto regression model (VAR) [39] both in precision and performance.

The rest of the paper is organized as follows. Section 2 reviews relevant literature on TSLP. Section 3 describes CLA and ICLA. Section 4 proposes ICLA-EC as a new model. In *section 5,* a new TSLP algorithm is introduced using ICLA-EC. Moreover, the experimental study on predicting e-mail and co-authorship links is presented in section 6. Finally, section 7 summarizes the main conclusions of the paper and discusses future directions.

2 Related works

In this section, a few previous research studies regarding exploitation of temporal information for LP are discussed in brief.

Soares et al. [40] introduced a new proximity measure for LP using the idea of temporal events. They thus used temporal

events like creation, maintenance, or interruption of relations between nodes in successive time periods. In their method, an event-based score is also defined and then modified over time by rewarding the temporal events identified among node pairs and their neighbors. Furthermore, they performed several experiments on co-authorship networks to assess the suggested event-based system in an LP problem.

As another method to organize temporal information, it is possible to consider LP as a time-series prediction problem and set up a time-series for each node pair, in which every series observation is defined as frequency of connection incidence among nodes within a particular time [41]. In this respect, an ARIMA model is used in [42] to produce time-series predictions and to calculate probability of connection incidence in the future. A new LP method utilizing both time-series patterns and similarity algorithms is further introduced by Huang et al. [16]. According to this algorithm, the time-series data are created using LP at different times and the ARIMA model could predict its subsequent value. The final predictions are thus computed according to a mixture of predicted outcomes beside a selected similarity-based method.

Considering temporal similarity metrics and continuous action-set learning automata (CALA), Moradabadi et al. [23] introduced a new algorithm using various similarity metrics and time periods. In this method, an LP problem is regarded as a noisy optimization one and a group of CALA is applied to find a solution to this problem. The acquired LP outcomes also demonstrated that the suggested algorithm is appropriate for some social network datasets. Similarly, another time-series LP algorithm is proposed in [22], employing LAs to predict existence or non-existence of connections at time $T+1$ via some similarity metrics from time *1* to *T*. Also, in [24], an LP method based on FLP-DLA is proposed which could estimate the strength of test links using the network information. They also employed the strength of the test link as the output of the LP method.

For LP in evolving networks, a new algorithm called vector auto regression (VAR) for multivariate TSLP is introduced by Özcan et al. [39], combining (1) temporal network evolution, (2) node similarities, and (3) node connectivity information. In this algorithm, available connections and a computed similarity metrics are used for each time. They also performed several experiments to compare various similarity metrics.

In [15], a method is proposed to predict future connections employing CMA-ES to enhance weights used in a linear mixture of sixteen neighbors and node similarity scores. They further analyzed a dynamic social network with more than 10^6 nodes. Ultimately, their proposed approach demonstrated rapid convergence and high accuracy for the first twenty predicted connections.

In [18], a new LP problem is proposed for dynamic networks called interaction prediction (IP). The proposed method has accordingly used feature selection and time-series forecast to predict new links in the near future.

Employing information theory and mutual information of social structure, a novel LP approach, which is called (MI-LP), has been also introduced in [17]. They thus used ten networks to analyze their method and compared it with six common prediction algorithms. As well, the achieved outcomes demonstrated an improvement in LP precision and a logical computational complexity.

In [38] an unsupervised link prediction algorithm is introduced based on an ant colony system. In the proposed approach, initially special subgraphs are identified as triangular triads in the network. Then, their evolution is assessed so as to predict new edges in the network. The ant colony system is used to discover triangular subgraphs. Not only does the introduced algorithm improve the execution time, the achieved results for some datasets are better compared to those of other unsupervised link prediction methods.

3 Background

In this section; problem formulation, LA, and CLA are briefly reviewed.

3.1 Problem formulation

In this paper, undirected unipartite graphs are studied entirely. Following [16, 40], a TSLP problem is formally introduced. At first, the temporal network dataset is separated into time-sliced snapshots from time *1* to *T*. Considering *V* as a set of vertices, $V=(1,2,\ldots,N)$. $G=(G_1,G_2,\ldots,G_T)$ is a graph series related to a group of symmetric adjacency matrices $M=(M_1,M_2,\ldots,M_T)$, where G_t is the social network structure at time t with a vertex set V_t and a connection set E_t.At times t and $t+1$, the graphs include the same set of vertices; however, there is a various set of connections. Each M_t is a $N\times N$ matrix with nonzero elements $M_t(i,j)$ which correspond with edges $E_t(i,j)$ in E_t. The value of $M_t(i,j)$ is also from the set $\{0,1\}$, indicating the existence or otherwise non-existence of edge (i,j) through time period (t). Considering such a graph, the goal of the proposed LP method is to forecast the incidence of probabilities of edges in time $T+1$ by employing adjacency matrices $M=(M_1,M_2,\ldots,M_T)$.

3.2 LA

An LA [29, 43] is an adaptive decision-making tool that learns an optimal action through repeated interactions with an unknown random environment. At each stage, LA chooses an action out of a set of finite ones based on a probability distribution of the action-set. The selected action is then applied to the environment. After that, the environment evaluates the

selected action and sends feedback to the automaton, which is used to update the probabilities of actions [43]. Figure 1 indicates how an automaton interacts with its environment.

A linear formula can be thus used for updating action probabilities, once the action i is performed. When the positive response $(\beta = 0)$ is received, the action probabilities are updated through Eq. (1):

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1-p_i(n)] \\ p_{j\neq i}(n+1) &= (1-a)p_j(n) \end{aligned} \tag{1}$$

When the negative response $(\beta = 1)$ is received from the environment, the action probabilities are also updated via Eq. (2):

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_{j\neq i}(n+1) &= \left({}^{b}/_{r-1}\right) + (1-b)p_j(n) \end{aligned} \tag{2}$$

Wherein a and b are reward and penalty parameters. When $a = b$, the above equation called linear reward–penalty L_{R-P}. If $b = 0$, the presented equation is labeled as linear reward-Inaction L_{R-I}; and if $0 < b << a < 1$, it is called linear reward-ε penalty $L_{R-\varepsilon P}$.

3.3 CLA and irregular CLA

CLA [30], made by combining CA [44] and LA, is a powerful scientific model for some decentralized problems. During the interaction of multiple LAs, LA potentials can be thus recognized completely. When each cell in CA is equipped with an LA, a CLA is created. The action probability of the LA residing in a cell is also used to identify its state. Similar to CA, there is a rule controlling CLA performance. The reinforcement signal to LA is also identified by the rule for CLA and the actions chosen by adjacent LAs. The CLA can be accordingly divided into synchronous, in which all cells utilize their local rules simultaneously assuming an external clock to cause synchronous events for each cell; and asynchronous, in which just some cells are activated and the state of others do not change.

The generalized form of common CLA called ICLA [32] can get over the limitations of rectangular grid construction. ICLA is an undirected graph that each vertex is equipped with an LA. Regardless of its irregular limitation, ICLA performance is identical to that of CLA.

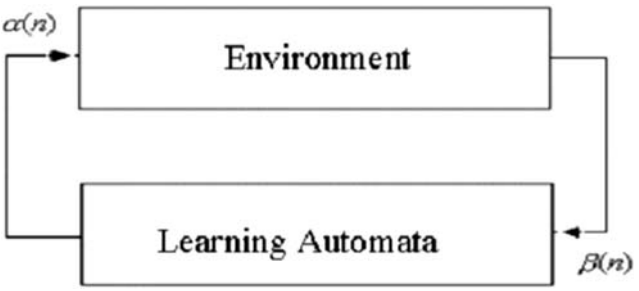

Fig. 1 The interaction between learning automata and environment

4 Proposed ICLA-EC

In this section, ICLA-EC model is proposed. It should be noted that fundamental units of every social community are individuals, making specific selections, adopting certain characters, and operating in several emotional forms. Information about individuals is also local, and they influence each other within a particular neighborhood mutually and locally. Learning and adaptation are correspondingly essential features of individuals in this social community. Each individual can accordingly learn from his own experiences and the experiences of his adjacent individuals. By voting on experiences of the best neighboring individuals, individuals can also learn others' experiences. Consequently, an individual can update his knowledge according to the results of previous self-experience as well as voting. These characteristics of individuals in the social community have motivated us to introduce a new EA called ICLA-EC, which is a hybrid algorithm based on ICLA and EC. The intuition behind this algorithm is the irregular structure and the synchronous feature of ICLA-EC, which is compatible with the requirements of social networks. ICLA-EC is also defined as a connected undirected graph in which each vertex represents a cell, and each edge induces an adjacency relation between two cells. Each cell is equipped with two components; a model genome and a genome. A model genome is composed of m LAs. Each LA also has a finite set of actions. The group of actions selected by the set of LAs of a particular cell concatenates to form the second component of the cell i.e. the genome. On the other hand, the genomes are associated with the network nodes.

Formally, an ICLA-EC with N vertices (i.e. cells) and m LAs in each vertex is a structure $\mathcal{A} = \left(G < V, E >, \Phi, A, P, \alpha, Ne, \mathcal{F}\right)$, where;

1- G is a connected undirected graph, with V as a set of vertices (i.e. cells) and E as a set of edges (that is, adjacency relations).
2- Φ stands for a finite set of actions.
3- $A = \left\{A_1, A_2, \ldots, A_N\right\}$ refers to population of model genomes assigned to ICLA-EC, wherein $A_i = \left\{LA_1^i, LA_2^i, \ldots, LA_m^i\right\}$ indicates a set of LAs allocated to cell c_i.
4- $P_i = \left\{P_1^i, P_2^i, \ldots, P_m^i\right\}$ represents action probability vector of model genome A_i.
5- $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ shows population of genomes, where $\alpha_i = \left\{\alpha_1^i, \alpha_2^i, \ldots, \alpha_m^i\right\}$ is genome in i^{th} vertex and

α_k^i represents value of k^{th} gene, $1 \le i \le N$, $1 \le k \le m$ and $\alpha_k^i \in \Phi$.

6- Neighborhood vector of any particular cell c_i is also defined as a set of all model genomes residing in adjacency of cell c_i, that is $Ne_i = \left\{ A_j \middle| \{i,j\} \in E \right\}$.

7- $\mathcal{F} : \Phi \rightarrow \beta$ indicates local rule of ICLA-EC. β is a set of values that the reinforcement signal can take. The local rule also computes the reinforcement signal based on current genome residing in cell and neighboring genomes of cell.

Algorithm 1, which is shown in Fig. 2, demonstrates the ICLA-EC steps. The operation of an ICLA-EC, in any particular vertex v_i, also takes place as follows. Each LA_j^i selects one of its actions according to its action probability vector P_i. The actions chosen by the set of LAs of the vertex are concatenated to each other to form a new genome ω_i for that vertex (line 4 of Algorithm 1). The fitness of new genome ω_i is then evaluated (line 5 of Algorithm 1). If the fitness of the ω_i is better than the current genome, ω_i replaces the current genome of that cell (line 6 to 8 of Algorithm 1). Next, a number of neighboring vertices of the vertex, according to the fitness evaluation of their corresponding genomes, are selected for mating (line 9 of Algorithm 1). The result of the mating process in the vertex is some reinforcement signals, one for each LA_j^i of the vertex (line 10 of Algorithm 1). The method of computing the reinforcement signal for each LA_j^i is described in Fig. 5. Each LA_j^i also updates its action probability vector based on the supplied reinforcement signal and its selected action (line 11 of Algorithm 1). This process continues until a termination criterion is met. The relationship between the genome and the model genome for ICLA-EC is illustrated in Fig. 3.

ICLA-EC can be also compared with other PEAs reported in the related literature in three ways: first, local interaction of LAs in ICLA-EC is based on a graph structure while it is based on a lattice in PEAs. Second, the local rule in ICLA-EC is a combination of local selection and reinforcement signal vector, but it is a combination of local selection, crossover,

Algorithm 1: procedure of ICLA-EC

1. ***Initialize***
2. ***While*** not a predetermined criterion is met ***do***
3. ***For*** each vertex v_i in the ICLA-EC ***do in parallel***
4. Generate a new genome according to probability vector
5. Evaluate the new genome based on fitness function
6. ***If*** *fitness*(new genome)> *fitness*(old genome) ***then***
7. The new genome replaces the genome of the vertex
8. ***End if***
9. Select *several* vertices based on their fitness from the neighbors of vertex
10. Generate the reinforcement signal vector
11. Update LA of vertex based on reinforcement signal and learning algorithm
12. ***End parallel for***
13. ***End while***

Fig. 2 Pseudo-code of the proposed Irregular Cellular Learning Automata-based Evolutionary Computation

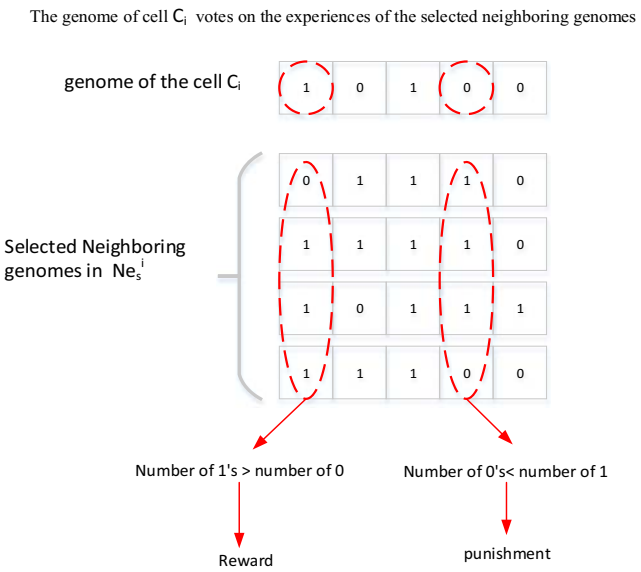

Fig. 5 The process of computing the reinforcement signal for LAs residing in a cell

and mutation in PAEs. Third, an individual in ICLA-EC is composed of a genome and a set of LAs, while it is a genome in PAEs. In the following section, this model is used to solve TSLP in a social network.

5 TSLP based on ICLA-EC (ICLA-EC-TSLP)

In this section, an algorithm based on ICLA-EC for TSLP, called ICLA-EC-TSLP, is presented.

There is an ICLA-EC that is isomorphic to the vertices of the input social network. On the other hand, each cell of ICLA-EC is assigned to a vertex of the social network, and the set of edges in the social network shows adjacency relations. Considering that each cell c_i has been assigned to vertex v_i in the graph structure of the social network, from this point on, vertex v_i and cell c_i are used interchangeably. As we have shown in Fig. 4, the main goal of given algorithm is to employ time *1* through *T* consecutively to predict future connections in time $T+1$.

To achieve this goal, after defining model genomes for vertices of social network, the ICLA-EC-TSLP algorithm tries to predict links in an iterative manner by using local information. The process of predicting links for each cell is guided by the set of LAs residing at the cell. With the aid of ICLA-EC-TSLP, the set of obtained predicted links in each cell is improved by both the previous self-experience and reinforcement signal which is generated by using genome of the cell and the best adjacent genomes.

The initial phase of this algorithm is explained below (lines 4 and 5 of Algorithm 2).

In order to use ICLA-EC for predicting links in social networks, each cell c_i of the network can be modeled by a tuple

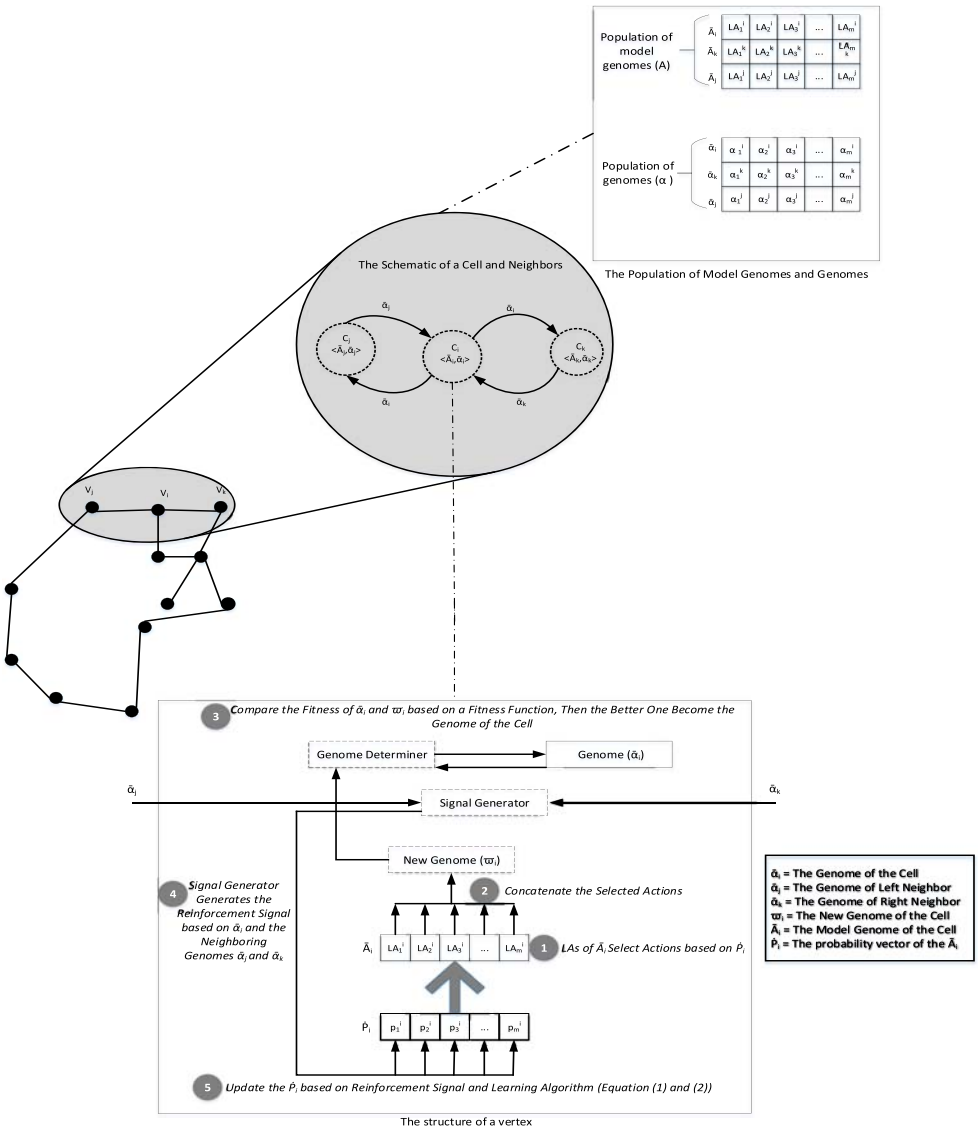

Fig. 3 Schematic of Irregular Cellular Learning Automata-based Evolutionary Computation. The numbers 1 through 5 on the figure represent the steps of the proposed algorithm

$< A_i, \alpha_i >$, where A_i depicts a model genome in which $A_i = \{LA_1^i, LA_2^i, \ldots, LA_m^i\}$ indicates a set of LAs allocated to cell c_i. Each LA_j^i is assigned to a test link that corresponds to the cell c_i in time $T+1$ which must be predicted. So, LA_j^i analyzes the probability of any future connection between cell c_i and c_j. $\alpha_i = \{\alpha_1^i, \alpha_2^i, \ldots, \alpha_m^i\}$ is a genome in which $\alpha_j^i \in \{\alpha_{j0}^i, \alpha_{j1}^i\}$ (for each $\alpha_j^i \in \alpha_i$) indicates the set of actions that can be taken by LA_j^i and controlling existence or non-existence of the corresponding connection, respectively. The genome α_i shows predicted links for the cell. Every LA_j^i also contains a probability distribution $P_j^i = \{P_{j0}^i, P_{j1}^i\}, 1 \le i \le N, \quad 1 \le j \le m$ which points out the probability of selecting *0* or *1*, respectively.

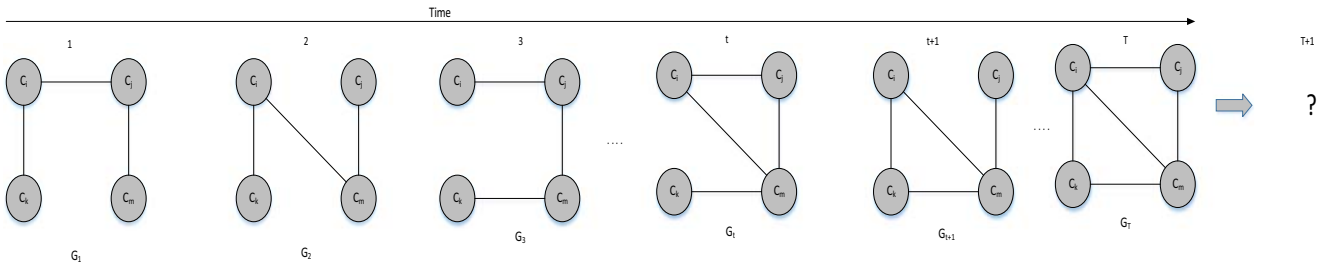

Fig. 4 Structure of social network for time 1 through T

Since no previous information exists about the presence of the connections, the initial probability of selecting action *1* (i.e. connection existence) is set to 0.5 i.e. $P^i_{j1} = 0.5$ and $P^i_{j0} = 1 - P^i_{j1}$ for each LA^i_j.

ICLA-EC-TSLP algorithm contains several principal steps i.e. $\{s_1, s_2, \ldots, s_T\}$. The social network structure of time t is called one step. In each step s_t, each LA i.e. LA^i_j in each cell c_i tries to learn about existence or non-existence of the corresponding link for time $t+1$ with the reinforcement signal produced from the local environment in time *t*. This prediction also repeats until some termination criteria are met. When prediction operation of step s_t is completed, the ICLA-EC-TSLP goes to the next time. Then, in the next time, every cell attempts to enhance its estimation via the new social network structure in time $t+1$ and predict the connections incidence of the corresponding links for time $t+2$ and the rest. In each step the structure of social network will be changed. So, ICLA-EC-TSLP in each cell can use different experiences of adjacent cells while it moves along different steps for predicting links.

In addition to the initial phase, there are two other principal phases: (1) local search phase as a repetitive one to learn ICLA-EC, and (2) prediction phase as one to generate the output of the proposed method.

In step t, for each cell c_i, when LAs in $A_i = \{LA^i_1, LA^i_2, \ldots, LA^i_m\}$ choose their actions and generate genome $\alpha_i = \{\alpha^i_1, \alpha^i_2, \ldots, \alpha^i_m\}$, each gene in α_i i.e. α^i_j determines whether the corresponding link appears in time $t+1$ or not. After the local search is applied to genome α_i, the probability vector of the model genome i.e. P_i is updated according to the learning algorithm of Eqs. (1) and (2) based on the reinforcement signal received from the local search on the genome of the cell and adjacent genomes of the corresponding cell. In the next iteration, LAs of model genome select new actions again based on the new probability vector of the model genome and this procedure repeats until a termination criterion is met. Then the CLA-EC-TSLP goes to the step $t+1$ and start to learn the links existence by the current action probability vector using the new environment $t+1$. Table 1 demonstrates the definition of the main symbols.

The pseudo-code in Algorithm 2 correspondingly accounts for the principal phases of the suggested algorithm in more detail, as discussed in the following sub-sections.

5.1 Local search phase

This sub-section presents the local search phase of the ICLA-EC-TSLP. Figure 6 demonstrates the pseudo-code for ICLA-EC-TSLP. To execute LP, ICLA-EC commences its learning procedure in step s_1. In this step, each LA i.e. LA^i_j in each cell c_i tries to learn about existence or non-existence of the corresponding link for time *2* with the reinforcement signal produced from the time one. This prediction also repeats until some termination criteria are met. Then, the ICLA-EC-TSLP goes to step *2*. In step *2*, the group of LAs, in each cell tries to update and improve its actions using the new time *2* in order to predict the occurrence or non-occurrence of the corresponding links for time *3*. These steps are further repeated until the ICLA-EC-TSLP goes to step S_T and tries to learn the predictions for time $T+1$. Considering that the group of cells (vertices) runs in parallel, the lifecycle of the given cell i.e. c_i in each step is explained as follows:

1- Each LA of model genome $A_i = \{LA^i_1, LA^i_2, \ldots, LA^i_m\}$ chooses its action using action probability vector $P_i = \{P^i_1, P^i_2, \ldots, P^i_m\}$ and employs it as an indicator of existence or non-existence of the corresponding link (line 10 of Algorithm 2).
2- The chosen actions of LAs are concatenated and a new genome $\omega_i(n) = \{\alpha^i_1, \alpha^i_2, \ldots, \alpha^i_m\}$ is generated.

Note 1. Steps 1 and 2 are related to the state in which an individual in a social community presents certain selections and operates in specific emotional forms.

3- The fitness of the genome is computed via Eq. (3) (line 11 of Algorithm 2).

$$fitness\ of\ string\ genome = \frac{\#correct\ links}{\#predicted\ links} \tag{3}$$

4- If the computed fitness is better than that of the current genome, the new genome $\omega_i(n)$ becomes the genome of that cell (lines 12 to 14 of Algorithm 2). That is,

$$\alpha_i(n) = \begin{cases} \alpha_i(n-1) & fitness(\alpha_i(n-1)) \geq fitness(\omega_i(n)) \\ \omega_i(n) & fitness(\alpha_i(n-1)) < fitness(\omega_i(n)) \end{cases} \tag{4}$$

Note 2. Step 4 is associated with the state in which an individual in a social community can learn from his earlier self-experience.

5- Of the neighboring vertices of cell c_i, a number of vertices whose genomes have a higher fitness than threshold γ are selected for mating (line 15 of Algorithm 2). We called this set of cells Ne^i_s. Note that the mating in this context is

Table 1 Definition of symbols

Symbol	Definition
c_i or v_i	Cell or vertex indexed by i in the ICLA-EC
A_i	Model genome of cell $c_i$that consists of a set of LAs
LA^i_j	j[th] LA in model genome A_i
α_i	Genome of cell c_i
ω_i	New generated genome of cell c_i
α^i_k	The value of k^{th} gene in i^{th} genome
P_i	Actions probabilities vector of A_i
Ne_i	Neighborhood vector of cell c_i
Ne^i_s	The set of best neighboring cells of cell c_i
Ne^i_{jk}	The number of cells in N^i_s whose j^{th} *LAs* have selected their k^{th} actions
β	The set of values that the reinforcement signal can take
a	Reward parameter
b	Punishment parameter
Φ	Finite set of actions
$F : \Phi \rightarrow \beta$	The local rule of the *ICLA-EC*
s_i	The step which is defined for i^{th} time

not reciprocal i.e. a cell selects another one for mating but not necessarily vice versa.

6- Defining Ne^i_{jk}, the number of vertices in Ne^i_s whose j^{th} LAs have selected their k^{th} actions is:

$$Ne^i_{jk} = \sum_{l \epsilon N^i_s} \delta\left(\alpha^i_j(n) = k\right) \tag{5}$$

where,

$$\delta(exp) = \begin{cases} 1 & if \ \ \text{exp}is\ true \\ 0 & otherwise \end{cases} \tag{6}$$

Then, the reinforcement signal for j^{th} LA i.e. LA^i_j is computed as [31]:

$$\beta^i_j(n) = \begin{cases} U\left[Ne^i_{j1} - Ne^i_{j0}\right], & if \ \ \alpha^i_j(n) = 0 \\ U\left[Ne^i_{j0} - Ne^i_{j1}\right], & if \ \ \alpha^i_j(n) = 1 \end{cases} \tag{7}$$

Where, $U(.)$ shows step function. According to Eq. (7), the genome of the vertex votes on the experiences of the selected neighboring genomes (line 16 of Algorithm 2). On the other hand, for j^{th} position of the genome, a reward signal is given to j^{th} learning automaton of the vertex if most of the vertices in Ne^i_s have selected the same action as the vertex has. This process is described in Fig. 5.

7- Each LA in $A_i = \left\{LA^i_1, LA^i_2, \ldots, LA^i_m\right\}$ updates its internal structure using the learning algorithm of Eqs. (1) and (2) and the given reinforcement signal (line 17 of Algorithm 2).

Notes 3. Steps 5, 6, and 7 are related to the state in which an individual in a social community selects some experts, and then updates his knowledge according to the results of voting amongst the experiences of the chosen individuals and earlier self-experience.

These procedures continue until a termination criterion is met. After the termination criterion is met, then the ICLA-EC-TSLP goes to the next step and these procedures repeated for the next steps. After stopping in step T, to predict the links for time $T+1$ we use the following prediction phase.

5.2 Prediction phase

For every LA^i_j in cell c_i, the action with a higher probability is selected as the ultimate prediction outcome for the related connection (lines 24 to 25 of Algorithm 2). In other words, in time $T+1$ for every test connection j corresponding to cell c_i, the eventual prediction is explained as ϑ^i_j and computed according to the following rule:

$$\vartheta^i_j = \begin{cases} \alpha^i_{j1} & P^i_{j1} > P^i_{j0} \\ \alpha^i_{j0} & else \end{cases} \tag{8}$$

At last, the ICLA-EC-TSLP algorithm considers set ϑ values as the result of the suggested LP method.

Algorithm 2: Procedure of the ICLA-EC-TSLP

1. **Let** $M_1, M_2, \ldots, M_T$ be the adjacent matrix for time *1* through *T*.
2. **Let** *I* be the maximum number of iterations for one step.
3. **Let** *i, s* be the iteration counter and step counter and initially set to *0* and let *S* be the total number of steps.
   ****initialization phase****
4. **Let** $\overline{A}$ in each cell, be the set of LAs indexes with *1* through *m*, in each cell one learning automaton for each link that must be predicted for that cell.
5. **Set** the initial probability distribution of choosing action *1* for each LA_j^i in a cell to be $P_{j1}^i(j) = 0.5$.
6. ****local search phase****
7. **While** $s < S$ do
8. **While** $i < I$ do
9. **For** each cell c_i in ICLA-EC do in parallel
10. Select actions $\overline{\alpha}_i = \{\alpha_1^i, \alpha_2^i, \ldots, \alpha_m^i\}$ of LAs based on the action probability $\overline{p}_i = \{p_1^i, p_2^i, \ldots, p_m^i\}$.
11. Evaluate the new genome based Equation (3).
12. **If** fitness (new genome) > fitness (old genome) then
13. Accept the new string genome
14. **End if**
15. Select a number of genomes from neighbors of the cell based on its fitness
16. Generate the reinforcement signal based on genome of the cell and genomes of the best adjacent cell using Equation (7)
17. Using Equations (1) and (2)Update the probability distribution of LAs of cell.
18. **End parallel for**
19. *i=i+1*
20. **End while**
21. *s=s+1*
22. **End while**
23. ****test phase****
24. **For** each test link *j* corresponds to the cell c_i
25. ϑ_j^i be the output of prediction result based on Equation (8).

Fig. 6 Pseudo-code of the proposed time-series link prediction based on ICLA-EC

6 Experiments and results

In this section, the experimental evaluation of the suggested method together with results is presented. In section 6.1, the required social network data along with the analysis process are explained. In section 6.2, the required assessment metrics for the present investigations, together with two proposed experiments, are described. The suggested method is also compared with some LP algorithms in section 6.3. Finally, the precision rate of ICLA-EC-TSLP is analyzed in section 6.3.1.

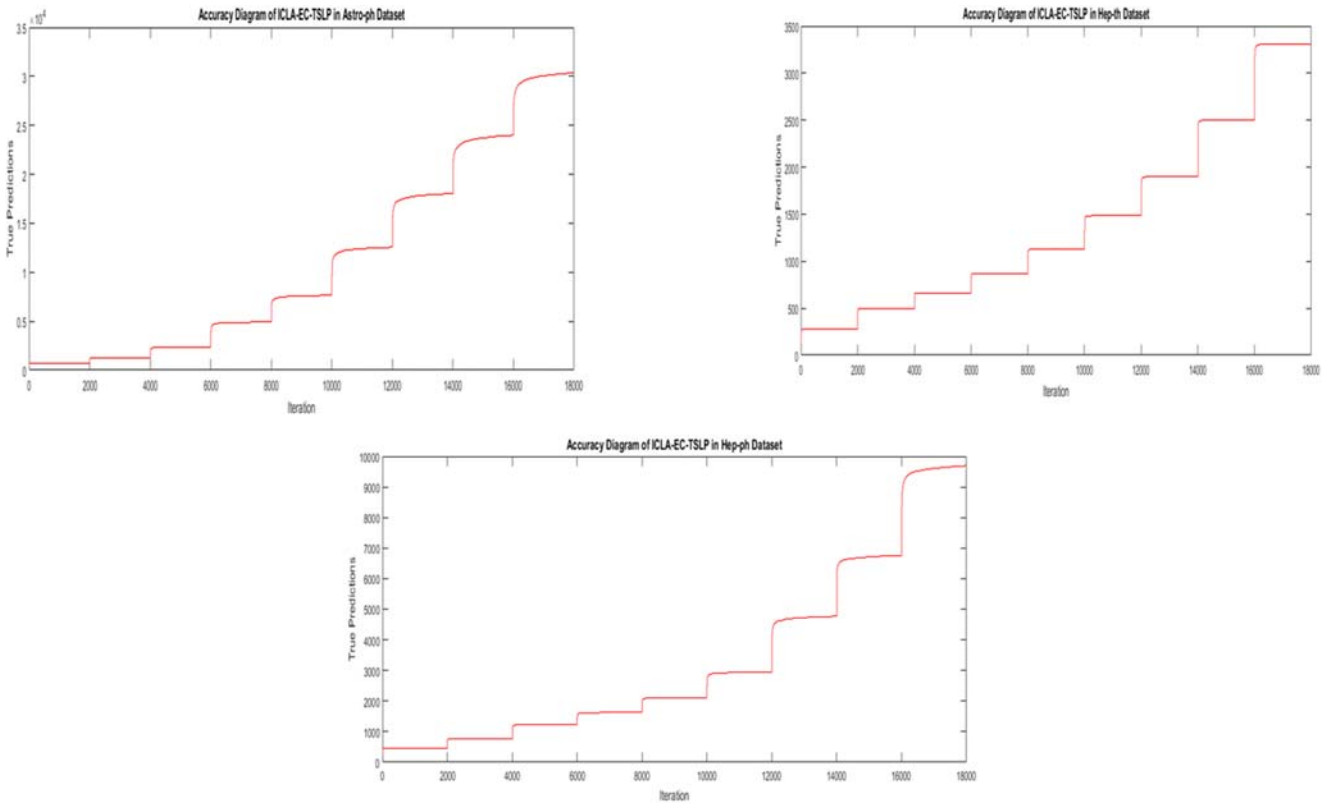

Fig. 7 The progressive accuracy diagram of the ICLA-EC-TSLP for co-authorship datasets

Table 2 The six similarity-based methods. Let Γ(x) denotes neighbors of the node x. The degree of node x is represented by $k(x)$

Topological similarity indices	Formula	Description
Common Neighborhood[3]	$CN(x,y) = \lvert\Gamma(x) \cap \Gamma(y)\rvert$	Two nodes x and y, are more likely to have a link if they have many common neighbors.
Jaccard Index[33]	$Jaccard(x,y) = \frac{\lvert\Gamma(x) \cap \Gamma(y)\rvert}{\lvert\Gamma(x) \cup \Gamma(y)\rvert}$	Measures the probability that a neighbor of x or y is a neighbor of both x and y.
Preferential Attachment[35]	$PA(x,y) = k(x) \times k(y)$	Gives higher scores to pairs of nodes for which one or both have high degree.
Adamic-Adar Index[34]	$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(\lvert\Gamma(z)\rvert)}$	This index refines the simple counting of common neighbors by assigning the less-connected neighbors more weight.
Local Path Index[36]	$Local\ Path\ Index(x,y) = A^2 + \varepsilon A^3$	A restricted version of the Katz metric such that only paths of length 1 and 2 are considered.
Katz Index[37]	$Katz(x,y) = \sum_{l=1}^{\infty} \beta^l . \lvert path(x,y)^{<l>} \rvert$	Sum of the number of paths with different lengths, such that shorter paths have more weights.

The introduced ICLA-EC-TSLP is compared with two categories:

1- Topology-based similarity metrics: Of this set of algorithms, local similarity metrics is chosen such as CN [3], JC [33], PA [35], and AA [34]; global similarity metrics such as Katz index [37] and quasi-local metrics such as local path [36]. The definition of this set of metrics is also shown in Table 2.

2- Learning-based metrics: In this group, feature-based classification algorithms such as IP [18], CMA-ES [15], MI-LP [17], ARIMA-LP [16], VAR [39], ACO-LP [38]; and LA-based algorithm like LA-TSLP [22], CALA [23], FLP-DLA [24] are chosen.

6.1 Data and setting

With a specific end goal to assess the execution of the proposed method, data from three co-authorship networks, one college messaging network, and one email network are used in these experiments. Table 3 describes the features of these datasets.

A vertex in a co-authorship network corresponds to an author, while an edge demonstrates that two particular nodes have co-authored a paper [45]. In the co-authorship networks, the publishing year of the co-authored paper is saved by every edge. The required co-authorship network data to analyze the proposed algorithm is also gathered using arXiv[1] e-print, which preserves an immense database of various scientific articles. To analyze the proposed approach, three co-authorship networks are considered, and the required data are gathered for the interval between 1993 and 2003 for all these datasets. The first network includes papers of authors who went along theoretical high energy physics (Hep-th) while the second one contains the papers of those who collaborated in high energy physics (Hep-ph). Finally, the third network involves papers of authors who studied Astro Physics area (Astro-ph). Co-authorship networks are highly sparse; therefore, it is necessary to use a lower number of candidate pairs to improve the calculation process. Therefore, just those with not less than two contributions between 1993 and 2003 are selected.

Enron email communication network as a public dataset is also used as the second dataset to analyze the proposed approach. The required data are thus gathered for the months between May 1999 and May 2002 (36 months). In the Enron email network, a node corresponds to an email address, while an edge demonstrates that address *j* has received at least one email from address *i* [46].

A college messaging network is further considered as the third dataset whose network includes students of California University, Irvine. This network dataset encompasses an interval between April and October 2004 [47].

6.2 Evaluation metrics

Two common evaluation metrics are used to assess the proposed method with other LP methods as follows [3].

AUC: Considering the score given to all non-observed links, the AUC value can be interpreted as the probability that a randomly chosen missing link is assigned with a higher score than a randomly chosen non-existent link. Accordingly, a missing link as well as a non-existent one is randomly selected for a comparison of their scores. If out of n independent comparisons, n' times missing links have higher

[1] Arxiv.org e-print archive

Table 3 Network size in terms of nodes and edges

Data Set	Nodes	Edges	Description
Hep-th	9877	51,971	Collaboration network of ArXiv's High Energy Physics Theory
Hep-ph	12,008	237,010	Collaboration network of ArXiv's High Energy Physics
Astro-ph	18,772	396,160	Collaboration network of ArXiv's Astro Physics
Email-Enron	87,273	1,148,072	Email communication network from Enron
CollegeMsg	1899	59,835	College message network of students at the University of California

score and n" times have the same score, the AUC value is:

$$AUC = \frac{n' + 0.5n''}{n} \tag{9}$$

Precision: Given the rankings of the non-observed links, precision is specified as the ratio of chosen relevant items to the number of those which are chosen. In other words, it is supposed that the top-L links are taken to be the predicted ones, out of which L_r links are right; as a result, precision equals Eq. (10). Obviously, higher precision denotes greater accuracy of prediction.

$$Precision = \frac{L_r}{L} \tag{10}$$

6.3 LP comparison

In this section, results for ICLA-EC-TSLP are provided and compared with similarity-based methods. Each result reported in the following sections is based on averages taken over 20 runs. According to experimental outcomes, ICLA-EC-TSLP is used in the conducted experiments with a reward parameter $a = 0.04$ and penalty parameter $b = 0.01$. The convergence of all LAs is also considered as a termination condition at a specific time. The algorithm may also terminate after 2000 iterations at a specific time. Note that these tables give the best outcomes in bold.

6.3.1 Co-authorship networks

Table 4 shows the AUC value of the proposed algorithm and similarity-based methods for Hep-ph, Hep-th, and Astro-ph datasets. In this experiment, each year is considered as a time. To do this, 10 years from 1993 to 2002 is regarded as the training data and the test links for the years 1996 through 2003 are predicted. To predict a test link in year α, ICLA-EC is also used which consists of steps 1993 through $\alpha - 1$. The LP results of the ICLA-EC-TSLP for years 1996 through 2003 are also compared with some similarity-based algorithms using the AUC metrics. The results confirm that the proposed method can achieve a better AUC measure than CN, PA, AA, and JC.

The growing precision diagrams of ICLA-EC-TSLP for co-authorship datasets are demonstrated in Fig. 7. According to these results, ICLA-EC-TSLP acts stochastically during the first steps, while it learns new patterns after several steps which are the combination of older ones as well as novel environmental patterns.

6.3.2 Enron E-mail dataset

In this dataset, each three-month is considered as one time. To do this, 6 times from January 2000 to December 2001 are considered, and the test links for times of May 2000 through April 2002 are predicted and compared with some similarity-based LP using the AUC metric.

The results reported in Table 5 also show that the proposed TSLP method can achieve better results compared with CN, JC, PA, and AA on average.

6.3.3 College messaging network

In this experiment, ICLA-EC-TSLP is compared with some similarity-based methods for a college messaging network using the AUC metrics. To do this, each month is considered as one time. The analysis also includes all users sending or receiving at least one message during that period. The obtained results of the proposed algorithm and its comparison with other methods for college messaging network are reported in Table 6.

The results presented in Tables 4, 5, and 6 confirm that the ICLA-EC-TSLP algorithm can achieve the highest AUC criterion value for various datasets compared with local similarity-based metrics such as CN, AA, PA, and JC. The reason for this superiority is that temporal information is an important aspect to consider for LP.

6.3.4 Comparison of ICLA-EC-TSLP and recent LP methods

In this section, the suggested ICLA-EC-TSLP algorithm is compared with the latest LP approaches. The IP, ARIMA-LP, VAR, LA-TSLP, FLP-DLA, CALA, and ICLA-EC-TSLP use temporal information in different time periods.

According to Table 7, the suggested ICLA-EC-TSLP can reach AUC (0.8010) measure, which far surpasses local path

Table 4 Comparison of AUC of the ICLA-EC_TLSP with similarity-based methods on Astro-ph, Hep-ph, Hep-th datasets

Method	CN			JC			AA			PA			ICLA-EC-TSLP		
Year	Astro-ph	Hep-ph	Hep-th	Astro-ph	Hep-ph	Hep-th	Astro-ph	Hep-ph	Hep-th	Astro-ph	Hep-ph	Hep-th	Astro-ph	Hep-ph	Hep-th
1996	0.5240	0.5230	0.5350	0.5243	0.5255	0.5347	0.5247	0.5250	0.5378	0.5273	0.5125	0.5150	**0.8590**	**0.8247**	**0.7863**
1997	0.5233	0.5705	0.5480	0.5287	0.5725	0.5540	0.5190	0.5735	0.5494	0.5110	0.5675	0.4910	**0.8920**	**0.8069**	**0.7924**
1998	0.5507	0.5250	0.5390	0.5577	0.5265	0.5413	0.5647	0.5205	0.5443	0.5090	0.5125	0.5005	**0.8802**	**0.8926**	**0.7863**
1999	0.5737	0.5580	0.5677	0.5850	0.5530	0.5630	0.5950	0.5545	0.5610	0.5263	0.5278	0.5114	**0.8821**	**0.8190**	**0.7582**
2000	0.5503	0.5505	0.5443	0.5600	0.5465	0.5507	0.5525	0.5470	0.5486	0.5112	0.5315	0.5263	**0.8577**	**0.8265**	**0.7916**
2001	0.5583	0.5365	0.5507	0.5365	0.5415	0.5510	0.5750	0.5460	0.5590	0.5167	0.5015	0.5312	**0.8467**	**0.8729**	**0.7835**
2002	0.5250	0.5840	0.5600	0.5437	0.5805	0.5593	0.5937	0.5850	0.5587	0.5214	0.5412	0.5175	**0.8550**	**0.8544**	**0.74**
2003	0.5350	0.5275	0.5577	0.5550	0.5385	0.5587	0.5788	0.5420	0.5537	0.5188	0.5510	0.5212	**0.8650**	**0.8515**	**0.7849**

and Katz algorithms. To predict future connections, local path and Katz algorithms utilize the path length between two nodes, while ICLA-EC-TSLP considers temporary neighborhood changes. The given method also surpasses the CMA-ES algorithm since it employs time-series information, while CMA-ES attempts to merge similarity metrics via defining a weight for every similarity. Given that the proposed algorithm is much superior to similarity-based methods, it is not surprising that it outperforms CMA-ES as well.

The results reported here show that the ACO-LP algorithm has the ability to achieve the better AUC criterion value for various datasets compared to local similarity criteria such as CN, AA, PA, and JC. The reason for this superiority is that the ACO-LP algorithm calculates the connection probability of two nodes in future based on local communities.

Moreover, it can be seen that ICLA-EC-TSLP achieves better AUC than the IP algorithm. Since both algorithms use time-series graph information, the superior results produced by the proposed algorithm are due to the fact that ICLA-EC-TSLP estimates the weight of test links using neighborhood changes that exist in time-series graphs. In contrast, the IP algorithm estimates the future value of any feature in the network using a prediction model. It becomes obvious that ICLA-EC-TSLP algorithm achieves better results than MI-LP. The reason for that is selecting different neighborhoods while moving along different time-series, whereas MI-LP only uses the mutual node information to predict links.

The ICLA-EC-TSLP is better than LA-TSLP, since every vertex utilizes a combination of local information of equivalent vertex and neighboring vertices; while LA-TSLP employs local similarity-based methods in time *1* through *T* to predict future connections. LA-TSLP has not used the experiments of other neighbors, so it has not ability to track the changes of the social network. The ICLA-EC-TSLP is thus better than FLP-DLA, since every vertex uses a combination of local information in time-series graphs. So, appropriate results spread over the whole population, because of neighborhood overlap; while FLP-DLA determines a fuzzy strength for each relation which is defined based on the date of link occurrence and the number of collaborations in the corresponding link. Then, the distributed learning automata use fuzzy links to find the strength of test links to predict future connections.

Considering that ARIMA-LP, VAR and ICLA-EC-TSLP use time series information, a better result can be obtained for two main reasons: 1) in contrast to ARIMA-LP and VAR algorithms, ICLA-EC-TSLP uses previous self-experience as well as voting and 2) ICLA-EC-TSLP considers dynamicity of adjacent nodes in graph navigation and because of neighborhood overlap appropriate results spread over the whole population, so it can adopt with graph changes and receive better results. In order to predict the future node similarity

Table 5 Comparison of AUC of the ICLA-EC_TLSP with similarity-based methods on Enron E-mail dataset

Method Year/month	CN	JC	PA	AA	ICLA-EC_TSLP
2000-5to8	0.6600	0.6627	0.6561	0.6593	**0.7014**
2000-9to12	0.6584	0.6565	0.5789	0.6578	**0.6112**
2001-1to4	0.6531	0.6548	0.5083	0.6496	**0.6694**
2001-5to8	0.5899	0.5835	0.5928	0.5919	**0.6247**
2001-9to12	0.6194	0.6209	0.5854	0.6155	**0.6459**
2002-1to4	0.5911	0.5916	0.5811	0.5841	**0.7212**

Table 6 Comparison of AUC of the ICLA-EC_TLSP with similarity-based methods on College Messaging dataset

Method Year/month	CN	JC	PA	AA	ICLA-EC_TSLP
2004–6	0.4999	0.5330	0.4883	0.3969	**0.6963**
2004–7	0.5330	0.5180	0.5129	0.4704	**0.7849**
2004–8	0.5024	0.4989	0.4871	0.4276	**0.8131**
2004–9	0.4904	0.4871	0.4734	0.3796	**0.7788**
2004–10	0.4451	0.4414	0.4291	0.3802	**0.7825**

Table 7 Comparison of AUC of the ICLA-EC_TLSP with recent methods on different datasets

Method/Dataset	Astro-ph	Hep-ph	Hep-th	Enron Email	College-Msg	Average
Local Path [36]	0.6322	0.6112	0.6370	0.6554	0.5334	0.6138
Katz [37]	0.6602	0.6527	0.6478	0.6827	0.5529	0.6392
ACO-LP [38]	0.5872	0.6253	0.5692	0.6122	0.5197	0.5827
IP [18]	0.6634	0.6478	0.6894	0.6733	0.5665	0.6481
MI-LP [17]	0.7985	0.8034	0.7634	0.7078	0.6994	0.7545
CMA-ES [15]	0.6476	0.6311	0.6581	0.6702	0.5597	0.6333
ARIMA-LP [16]	0.8226	0.8135	0.7716	0.7110	0.7389	0.7715
VAR [39]	0.8269	0.8195	0.7749	0.7145	0.7335	0.7738
LA-TSLP [22]	0.7325	0.7947	0.7553	0.7022	0.6945	0.7358
FLP-DLA [24]	0.8297	0.8201	0.7795	0.7212	0.7395	0.7780
CALA [23]	**0.8745**	**0.8527**	**0.7930**	**0.7395**	**0.7795**	**0.8078**
ICLA-EC-TSLP	**0.8650**	**0.8515**	**0.7849**	**0.7212**	**0.7825**	**0.8010**

scores in ARIMA-LP algorithm, a time series forecasting model, ARIMA, based on past node similarity scores is used.

The VAR model is a little better than ARIMA model because VAR algorithm uses different similarity metrics and link occurrences information.

Furthermore, CALA is a little better than the proposed approach because it considers different similarity metrics and times with a variety of coefficients.

7 Conclusion

In this paper, a novel TSLP approach employing a combination of ICLA and EC is proposed to predict existence or non-existence of connections in time $T+1$ via a network construction from time *1* to *T*. The traditional static algorithms consider the network within time *T* to predict new relationships at the future time $T+1$. However, the proposed algorithm considers time-series information via observation of network evolution over time. In other words, the suggested procedure is an EA in which every genome corresponds to a part of the solution, and the entire population is equivalent to the complete solution. In the suggested method, a genome and a group of LAs is assigned to every vertex of input social networks. The genome also corresponds to the predicted connections of the corresponding cell and saves the histories of exploration. ICLA-EC-TSLP is thus a distributed algorithm in which every genome, regardless of any previous information, is locally developed via local information. Since the vertices in social networks are dynamic and also social networks are branching, changes in social networks take place at different times. Neighboring vertices may also appear, disappear, or even go unchanged at different times. A realization of the evolutionary patterns in the network can thus help in understanding the evolution of the network. In this paper, ICLA-EC is used to analyze network evolution through neighborhood dynamicity. Several experiments are further devised to analyze the performance of ICLA-EC-TSLP compared with other LP algorithms. According to the results of the experiments, ICLA-EC-TSLP surpasses earlier approaches. So, the mixture of interaction among vertices and the dynamicity of adjacent nodes lead to an improvement in the prediction process.

References

1. Lü L, Zhou T (2011) Link prediction in complex networks: a survey. Physica A: Stat Mech Appl 390(6):1150–1170
2. Amaral LAN, Scala A, Barthelemy M, Stanley HE (2000) Classes of small-world networks. Proc Natl Acad Sci 97(21):11149–11152
3. Al Hasan M, Zaki MJ (2011) A survey of link prediction in social networks. Social network data analytics. Springer, In, pp 243–275
4. Martínez V, Berzal F, Cubero J-C (2017) A survey of link prediction in complex networks. ACM computing surveys (CSUR) 49(4):69
5. Dunlavy DM, Kolda TG, Acar E (2011) Temporal link prediction using matrix and tensor factorizations. ACM Trans Knowl Discov Data (TKDD) 5(2):10
6. Bhattacharyya P, Garg A, Wu SF (2011) Analysis of user keyword similarity in online social networks. Soc Netw Anal Min 1(3):143–158
7. Anderson A, Huttenlocher D, Kleinberg J, Leskovec J (2012) Effects of user similarity in social media. In: ACM, pp 703–712
8. Akcora CG, Carminati B, Ferrari E (2013) User similarities on social networks. Soc Netw Anal Min 3(3):475–495
9. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. J Am Soc Inf Sci Technol 58(7):1019-1031%@ 1532-2882
10. Valverde-Rebaza J, de Andrade LA (2013) Exploiting behaviors of communities of twitter users for link prediction. Soc Netw Anal Min 3(4):1063–1074
11. Liu H, Hu Z, Haddadi H, Tian H (2013) Hidden link prediction based on node centrality and weak ties. EPL (Europhysics letters) 101(1):18004

12. Qiu B, Ivanova K, Yen J, Liu P Behavior evolution and event-driven growth dynamics in social networks. In: 2010. IEEE, pp 217–224
13. Yang S-H, Long B, Smola A, Sadagopan N, Zheng Z, Zha H Like like alike: joint friendship and interest propagation in social networks. In: 2011. ACM, pp 537–546
14. Dong Y, Tang J, Wu S, Tian J, Chawla NV, Rao J, Cao H Link prediction and recommendation across heterogeneous social networks. In: 2012. IEEE, pp 181–190
15. Bliss CA, Frank MR, Danforth CM, Dodds PS (2014) An evolutionary algorithm approach to link prediction in dynamic social networks. J Comput Sci 5(5):750–764
16. Huang Z, Lin DKJ (2009) The time-series link prediction problem with applications in communication surveillance. INFORMS J Comput 21(2):286–303
17. Tan F, Xia Y, Zhu B (2014) Link prediction in complex networks: a mutual information perspective. PloS one 9(9):e107056
18. Rossetti G, Guidotti R, Pennacchioli D, Pedreschi D, Giannotti F Interaction prediction in dynamic networks exploiting community discovery. In: 2015. IEEE, pp 553–558
19. Clauset A, Moore C, Newman MEJ (2008) Hierarchical structure and the prediction of missing links in networks. Nature 453(7191): 98
20. Guimerà R, Sales-Pardo M (2009) Missing and spurious interactions and the reconstruction of complex networks. Proceedings of the National Academy of Sciences 106(52):22073–22078
21. Menon AK, Elkan C (2011) Link prediction via matrix factorization. In, 2011. Springer, pp 437–452
22. Moradabadi B, Meybodi MR (2017) A novel time series link prediction method: learning automata approach. Physica a: Stat Mech Appl 482:422–432
23. Moradabadi B, Meybodi MR (2016) Link prediction based on temporal similarity metrics using continuous action set learning automata. Physica a: statistical mechanics and its applications 460:361–373
24. Moradabadi B, Meybodi MR (2017) Link prediction in fuzzy social networks using distributed learning automata. Appl Intell 47(3): 837–849
25. Whitley D (1995) A review of models for simple genetic algorithms and cellular genetic algorithms. Applications of Modern Heuristic Methods:55–67
26. Baluja S (1992) A massively distributed parallel genetic algorithm. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE
27. Baluja S (1993) Structure and Performance of Fine-Grain Parallelism in Genetic Search. In: ICGA, pp 155–162
28. Narendra KS, Thathachar MA (2012) Learning automata: an introduction. Courier Corporation
29. Thathachar MA, Sastry PS (2011) Networks of learning automata: techniques for online stochastic optimization. Springer Science & Business Media
30. Beigy H, Meybodi MR (2004) A mathematical framework for cellular learning automata. Adv Complex Syst 7(03n04):295–319
31. Rastegar R, Meybodi MR, Hariri A (2006) A new fine-grained evolutionary algorithm based on cellular learning automata. Int J Hybrid Intell Syst 3(2):83–98
32. Esnaashari M, Meybodi MR (2015) Irregular cellular learning automata. IEEE Trans Cybern 45(8):1622–1632
33. Jaccard P (1901) Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat 37: 547–579
34. Adamic LA, Adar E (2003) Friends and neighbors on the web. Soc Networks 25(3):211–230
35. Newman ME (2001) Clustering and preferential attachment in growing networks. Phys Rev E 64(2):025102
36. Zhou T, Lü L, Zhang Y-C (2009) Predicting missing links via local information. The. Eur Phys J B 71(4):623–630
37. Katz L (1953) A new status index derived from sociometric analysis. Psychometrika 18(1):39–43
38. Sherkat E, Rahgozar M, Asadpour M (2015) Structural link prediction based on ant colony approach in social networks. Physica a: statistical mechanics and its applications 419:80–94
39. Özcan A, Öğüdücü ŞG (2015) Multivariate temporal link prediction in evolving social networks. In: Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on. IEEE, pp 185–190
40. Soares PR, Prudêncio RB (2013) Proximity measures for link prediction based on temporal events. Expert Syst Appl 40(16):6652–6660
41. Potgieter A, April KA, Cooke RJ, Osunmakinde IO (2009) Temporality in link prediction: understanding social complexity. Emergence: Complexity Organ (E: CO) 11(1):69–83
42. Janacek G (2010) Time series analysis forecasting and control. J Time Ser Anal 31(4):303–303
43. Thathachar MA, Sastry PS (2002) Varieties of learning automata: an overview. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 32(6):711–722
44. Wolfram S (1994) Cellular automata and complexity: collected papers, vol 1. Addison-Wesley Reading, MA
45. Barabâsi A-L, Jeong H, Néda Z, Ravasz E, Schubert A, Vicsek T (2002) Evolution of the social network of scientific collaborations. Physica A: Statistical mechanics and its applications 311(3–4):590–614
46. Shetty J, Adibi J (2004) The Enron email dataset database schema and brief statistical report. Information sciences institute technical report, University of Southern California 4(1):120–128
47. Ahn Y-Y, Han S, Kwak H, Moon S, Jeong H (2007) Analysis of topological characteristics of huge online social networking services. In: Proceedings of the 16th international conference on World Wide Web. ACM, pp 835–844

Mozhdeh Khaksar Manshad received the B.S. and M.S. degrees in computer engineering from Islamic Azad University of Qazvin, Qazvin, Iran, in 2003 and 2007, and she is now Ph.D. student in computer engineering in the Islamic Azad University, South Tehran Branch, Tehran, Iran (from 2014). Her current research interests include evolutionary computing, social networks, learning systems, and soft computing.

Mohammad Reza Meybodi received the B.S. and M.S. degrees in economics from Shahid Beheshti University, Tehran, Iran, in 1973 and1977, respectively, and the M.S. and Ph.D. degrees in computer science from Oklahoma University, Norman, OK, USA, in 1980 and 1983, respectively. He is currently a Full Professor with Computer Engineering Department, Amirkabir University of Technology, Tehran. Prior to his current position, he was an Assistant Professor with Western Michigan University, Kalamazoo, MI, USA, from 1983 to 1985, and an Associate Professor with Ohio University, Athens, OH, USA, from 1985 to 1991. His current research interests include channel management in cellular networks, learning systems, parallel algorithms, soft computing, and software development.