Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 961 – 973

doi:10.3906/elk-1711-228

MutatedSocioAgentSim (MSAS): semisupervised modelling of multiagent simulation to predict and detect the mutation in a camouflaged social network

Karthika SUBBARAJ[1,*], **Bose SUNDAN**[2]
[1]Department of Information Technology, SSN College of Engineering, Chennai, India
[2]Department of Computer Science & Engineering, College of Engineering, Anna University, Chennai, India

Received: 19.11.2017 • **Accepted/Published Online:** 19.01.2018 • **Final Version:** 30.03.2018

Abstract: A social network is a networked structure formed by a set of agents/actors. It describes their interrelationships that facilitate the exchange and flow of resources and information. A camouflaged social network is one such community that influences the underlying structure and the profile of the agents, to cause mutation. The proposed MSAM is a novel system that simulates a multiagent network whose community structure is analyzed to identify the critical agents by studying the mutations caused due to attachment and detachment of agents. The isolation of the tagged agents will demonstrate disruption of information flow, which leads to the dismantling of the camouflaged community and giving scope for a predictive study about near future reconciliation. The proposed system simulates the 9/11 covert network based on the belief matrix and uses the novel density-based link prediction and suite of fragmentation algorithms for predictive community analysis. MSAM is claimed to be an intelligent system as agents perceive the knowledge from the dynamic environment through the belief matrix and further co-evolve as a community upon which semisupervised methodologies are used to predict the critical agents causing serious mutation.

Key words: Multiagent simulation, camouflaged social network, semisupervised learning, community structure, change detection, fragmentation, 9/11 covert network

1. Introduction

The real world can be viewed as a network of communication and relationships easing the exchange and flow of resources and information like models, data, and facts among the participants and this perspective is a social network [1]. A camouflaged social network is one such community that mutates the underlying structure and the profile of the involved agents by influence. The major challenge of human society is their complex and dynamic nature with nonlinear interactions among them. Moreover, in the case of a camouflaged network, there is a serious lack of data set, which acts as barrier to understand the communication among the agents. The major reason lies in collecting the data and the nature of the data (like incompleteness, erroneous/fakeness, and fuzziness). The information flow in this social structure allows the agents to team up with any other agents just on the aspect of interestingness. This becomes favorable for the camouflaged agent/influencer to spread the factor of their influence. The dynamism further complicates the problem by making the social network structure frequently mutative and demands continuous change detection methodologies. A camouflaged social network is a typical example of a complex network in the real world. Due to its dynamic nature, the simulation model would be more suitable than any other model for approximation and decision making as it can present

*Correspondence: skarthika@ssn.edu.in

an insight of network state changes at any point in time. The proposed simulation system supports predicting the output of a system well not just based on specific inputs but by considering system dynamics.

The proposed MutatedSocioAgentSim (MSAS) system adapts the approach of placing much lower demand on the data, while concentrating on the to-be-developed model that can truly reflect the complex nature of societies and avert any notorious influencers. The objective of MSAS is to model the first of its kind system for simulating an environment to detect and predict the frequent mutation of the social network structure for exposing abnormal behavior caused by notorious influencers. The proposed system contributes in the following three ways.

Firstly, the multiagent environment is simulated to achieve self-perception through the knowledge base encompassing communication patterns, information, and resource flow. Secondly, the novel link prediction algorithm is implemented for mutation detection and prediction. This algorithm applies the distance-based link reciprocity threshold for identifying the influencer agent/s and, thirdly, the system demonstrates the dismantling of the network by isolation of predominant agents based on the optimized antiresilience parameter-degree of fragmentation. The remainder of the paper is structured to present the related work carried out by researchers in the relevant areas to support the problem described earlier. It is followed by the proposed framework of the MSAS system and elaborated algorithm explanations. The later sections present the results and discussions of the findings and the conclusions, which lead to future enhancement of the proposed system.

2. Related work

The proposed research work and findings in the area of dynamic social network analysis (DSNA) can be categorized into two groups, namely multiagent simulation and community analysis.

Multiagent systems (MASs) are models of human organizations as they are based on the idea of intelligence, communication, cooperation, negotiation, and massive parallel processing [1]. Initially a basic distinction was made between two main MAS paradigms: multiagent decision systems and multiagent simulation systems [2], which lead to the evolution of the agent-based modelling system (ABMS). Later [3] proposed complex adaptive systems (CASs), describing them as intelligent, task-oriented, bounded-rationally, and socially situated agents that are embedded with an environment that also has the potential for change. The dynamism, evolution of influence, and communities in agent-based systems was studied by considering the social theories. A multiagent perspective for diffusion in social networks was presented with respect to diffusion in agent, media, and contents by representing diffusion models that are helpful in understanding the interaction protocols and decision making mechanisms [4–6]. Communities are considered to be sets of nodes in a network that have denser connectivity to each other than to the rest of the network and are important because they can often be closely related to functional units of a system [7–10]. The authors of [11] proposed structural clustering algorithm for networks (SCAN), which is an extension of DBSCAN [12] to find clusters, hubs, and outliers in large networks. The method proposed by [13] develops CT-SNAIR, which adapts automated techniques and tools for detection and tracking of dynamically changing terrorist networks as well as recognition of capability and potential intent based on multimedia data.

3. Methodology

The proposed MSAS system majorly focuses on detecting and predicting the mutation, and analyzing the impact of isolating the influencer in the social network structure. The system discusses about building a multiagent environment in order to collect the large-scale communication patterns, detect the influencers, and predict the impact of the social influence upon the mutation caused in the agent network. The overall architecture of the proposed system is illustrated in the following Figure 1.

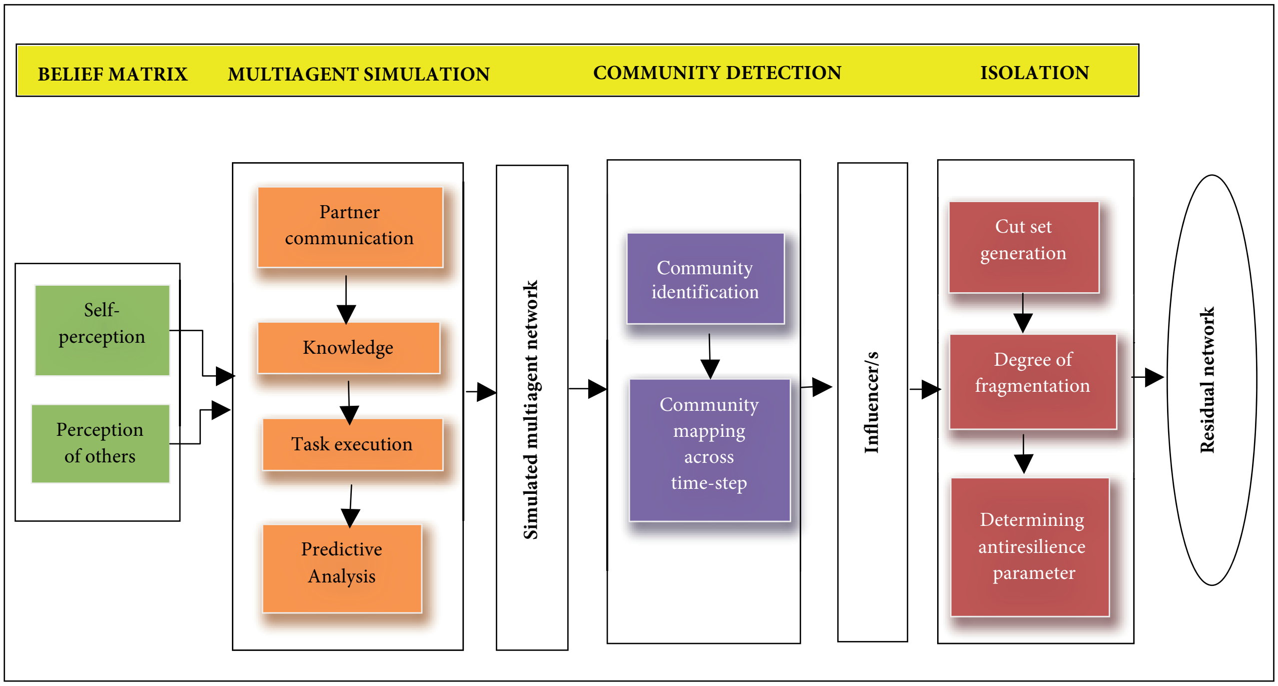

Figure 1. MSAS system proposed architecture.

3.1. Multiagent simulations

The proposed framework initially models a dynamic network to simulate the communication patterns, information, and resource flow on task-based principles. The proposed network paradigm follows the conventional postulates about the agents. They are independent; autonomous entities with limited intelligence; do not have accurate information about their neighbors and learn the state of the world through interaction; do not use predefined locations or neighbors and the networks in which they are a part usually co-evolve. These networks will evolve as communities when agents interact, learn, and perform tasks. The agents' perception about each other will decrease as their social distance decreases, which is similar to the empirical reality. The network is represented as a directed graph G_{Agents} and their social proximity is represented as D_{ij}, which correspond to the distance between the agents a_i and a_j:

Network = $\mathrm{G}_{Agents},\ \mathrm{D}_{ij}$

$$G_{\text{Agents}} \supset a_i : \text{ set of agents} \tag{1}$$

Every agent is accessed based on the communication probability vector (CP_k), which is the probability of an agent to communicate with all other agents to derive information about their neighbors. The CP_k helps to build the belief matrix and is represented as

$$CP_k = \frac{\text{Interaction of agent ak to } ai}{\text{Interaction of agent } ak \text{with agent in GAgents}} \tag{2}$$

The matrix stores the derived information from neighboring agents. The agent communicates with the other agent based on their social proximity. The proximity will be determined based on the relative similarity (Rel_{SIM})

and relative expertise (Rel_{EXP}), derived from the metamatrix of perceptions. These parameters are defined as follows:

$$Rel_{SIM} = \frac{\text{Similar knowledge between } ai, aj}{\text{Similar knowledge of } ai \text{ in GAgents}} \tag{3}$$

$$Rel_{EXP} = \frac{\text{Exclusive knowledge that } aj \text{ prosses from the perspective of } ai}{\text{Exclusive knowledge of Gagents from the perspective of } ai} \tag{4}$$

The multiagent simulation adds intelligence to the proposed system because in the initial stages the agents will have minimal knowledge and during knowledge exchange the agents will acquire more knowledge. The knowledge transfer will take place in three steps as depicted in Figure 2. The belief matrix will generate the CP_K value, which identifies the agent to be communicated with ($\mathbf{a_j}$). The agent ($\mathbf{a_i}$) in the information seeking mode will determine what to communicate and accordingly places a query. If $\mathbf{a_j}$ has a reply, it properly responds to $\mathbf{a_i}$; otherwise, based on its belief matrix, the system recognizes another agent in the same network who could respond to $\mathbf{a_i}$. This will be presented as referential data to $\mathbf{a_i}$. The knowledge base of $\mathbf{a_i}$ will be updated. When a task is assigned to an agent, it will be executed according on the knowledge base it possesses. A task could also be subtasked to multiple agents.

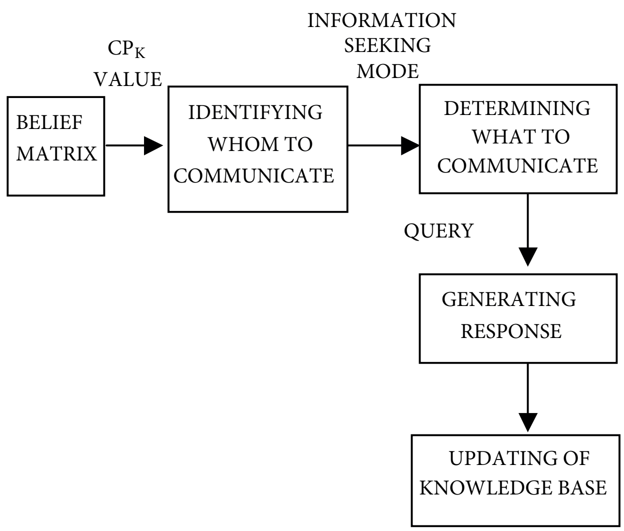

Figure 2. Knowledge transfer in multiagent network.

3.2. Community analysis

The major objective of this module is to track initial community structure and then to map it for different mutations at different time-steps. In the simulated network, SN_t at time t the changes in the community are identified using the active agents of the new time stamp t+1. Later, the resolution parameter is formulated to identify the overlapping structure. The nodes are initially defined as not labelled and not visited. The process selects a not visited node on a random basis. It identifies the first or primary community by assigning it as core agent (P). The community is formed by grouping the neighbors using the feature ε_P, local neighborhood threshold of P, which is computed as shown in Eq. (5).

$$\varepsilon_P = \frac{\frac{|\text{Answered links for } P|}{\text{set of nodes to which } P \text{ has outlinks}}}{\text{Total outlinks of } P} \tag{5}$$

If this is greater than zero, then a distance function is computed to find the neighbors of p as shown below.

$$dist(p,q) = \begin{cases} \Delta(p,q) & |V_{pq}| > (\eta * \min(|V_p|, V_q|)) \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

where $|V_{pq}|$ is the commonly interacting node and Δ(p, q) measures the maximum average link reciprocity among the nodes p, q and their common neighbors. It is determinable if the neighboring overlap among communities is significant. Smaller distance values represent higher response between the nodes and translate to more closeness. The link reciprocity threshold is optimized using the constant η. In simple terms, the distance function is computed if $|V_{pq}|$ is a predominant; otherwise the distance is taken as 1. If node P has nonzero reciprocated interactions then P is called a core agent. The detected communities are subjective to changes with time. These changes will result in many intermediate stages that show the evolved overlaps among communities. The following **Communalyzer** algorithm describes the steps of community evolution and the **Mutated_Community** algorithm presents the change detection methodology.

Algorithm 1 Communalyzer (SN_t, C_t, P, N_p)

```
/* SN_t is the simulated network state at time t */
/* SN_{t+1} is the present or network state at time t + 1 considered for community analysis */
/* C_t is the community id assigned to agent at time t
/* P is randomly chosen Primary Core agent */
/* N_p is the Neighbors of P */
/* NA_{VL} is the Neighboring agents in the Visited_List */
/*|V_P| is the commonly interacting agents among P */
Begin
    Mark all nodes of SN_{t+1} as unvisited
    for each unvisited node P do
        Add to Visited_list (P)
    while Visited_list is nonempty
        Determine the neighbors N_p at t using the local neighborhood threshold N_t = N_p^t (ε_P)
        Determine the neighbors N_p at t + 1 using the local neighborhood threshold N_{t+1} = N_p^{t+1} (ε_P)
        Identify the mutation in SN_{t+1}
            if P evolves as the node with Max(|V_p|) in the Visited_list and has common C_{t+1} then
                    Birth_Community (P)
                        Determine the N_p^{t+1} (ε_P)
                        Determine the C_{t+1} in Visited_list
                        Agent P forms a new community and is the core node of C_{t+1}
            End
            else if Visited_list is nonempty
                Mutated_Community (P, NA_{VL})
            End
    End
End
```

Algorithm 2 Mutated_Community (P, NA_{VL})

Begin
Growth_Community (P, $\mathbf{NA_{VL}}$)
NA_i € NA_{VL}
if NA_i has Max($|V_{na}|$) than P with common C_{t+1} then
  Compute dist (P, $|V_p|$) and dist (na, $|V_{na}|$)
  if NA_i has the max dist
   then NA_i is also core agent and P loses the primary core agent C_{t+1} includes new neighbors of NA_i
  End
End
Merge_Community(P, $\mathbf{NA_{VL}}$)
if NA_{VL} has $|\ C_{t+1}| > 1$
  then distinct C_{t+1} are merged and assigned a new community id C' with P as the core node
   (C_{t+1}, NA_{VL}) € C'
End
Shrink_Community(P, $\mathbf{N_P}$)
N_i is a set of neighbors of P in C_{t+1}
if dist(P, N_i) > link_reciprocity_threshold
  then C_{t+1} is split into C^N_{t+1} where N = 2
   Existing community C_{t+1} has split into at most 2 new communities losing few N_P
End
Split_Community(P, $\mathbf{N_P}$)
N_i is a set of neighbors of P in C_{t+1}
if dist(P, N_i) > link_reciprocity_threshold
  then C_{t+1} is split into C^N_{t+1} where N = 1,2,....i communities which have split from C_{t+1}
End
Death_Community(P, $\mathbf{N_P}$)
if N_P is empty remove the community id C_{t+1} from P
End
End

3.3. Isolation of influencer/s

The major objective in community analysis is to determine the ways in which information can be efficiently diffused. This objective demands identifying a/set of critical node(s) devoted for effective targeted communication and more receptive to behavior changes. This is called critical node/agent or cut point. In this research work, the agents in the region of overlap between the communities are treated as critical nodes and the edges that link them to the network are named bridges because such agents share common behavior features, which is determined using the high threshold value. The cut-vertex or bridge is thought to dismantle the existing cohesive network. A direct method for network graph fragmentation is to count the number of components after the removal of cut-points [14]. The count features the strength of the network after fragmentation. If the count is 1, then no fragmentation has occurred. Ideal separation is achieved if the number of components and the number of agents in the network are the same, i.e. all the agents are isolated from one another. The antiresilience factor counts the number of pairs of agents disconnected from each other, resulting in numerous fragments. The fragmentation component F is adapted from the research work of [14] as shown in Eq. (7).

$$F = 1 - \frac{2\sum_i \sum_{j<i} r_{ij}}{n(n-1)} \tag{7}$$

where r_{ij} is an instance of the adjacency matrix R. r_{ij} is 1 if i could reach j; otherwise it is 0; and n is the number of agents in the network. The metric F can also defined by the reciprocal of distance [14]. The parameter r_{ij} is replaced by $1/d_{ij}$, which provides a degree of reachability. This is termed the degree of fragmentation, D_F, which is mathematically shown in the equation given below.

$${}^{D}F = 1 - \frac{2\sum_{i<j} \frac{1}{d_{ij}}}{n(n-1)} \tag{8}$$

When distances within components are greater than 1, the measure captures the relative cohesion among them. The **AgentIsolator** algorithm uses the suite of metrics like Newman, Clique, and Johnson similarity [14]. This algorithm identifies the agents with unique metrics like the high degree centrality or group of agents with strong tie strength called a clique. This group fragmenter algorithm identifies the agents holding the residual components. The fragmenter algorithm determines the different combination of cut vertex. It iteratively computes the D_F after the removal of each combination. The cutset with maximum D_F is eliminated and number of components is determined. The fragmentation analysis determines the degree of damage caused to the network due to removal of the core agent. This measure will help the analyst to predict the chances of changes due to removal of an elite agent in the network. The resulting fragmented network is evaluated, based on the number of isolated components and the size of each fragment. The following section presents the results and discussion concerning the findings of the MSAS system.

4. Results and discussion

The proposed MSAS system uses the covert network involved in the 9/11 terrorist attack for the simulation module. Complete information about this attack has been collected from the 9/11 commission report (https://govinfo.library.unt.edu/911/report/911Report_Notes.htm) presented by the National Commission on Terrorist Attacks Upon the United States created by Congress and the President of the United States (Public Law 107-306, November 27, 2002). The dataset is structured as a timeline that has the year, location, hijacker names, other associate names, and their purpose in meeting or visiting that specific location. The hijacker is a person who was involved directly in the attack and other associate is a person who supported the hijackers to complete their mission successfully by giving financial aid, provided residence for the hijackers in the US, and gave weapons and flight training.

4.1. Simulated network

The actors involved in the attack are simulated as agents and the various relations existing among them are used in the belief matrix to build self-perception. The simulated network consists of 75 agents and involves 181 relations. The evolving communities are incrementally numbered in each time line. The community analysis presented in this section is from January 2001 to December 2001. Table 1 presents a sample of relations existing among the agents. The agents understanding/interpretation are derived from the matrix, which involves beliefs and interestingness/preferences of the agents. It can be listed as Roles & relations, Skill set & training, Category (Hijacker/ Other Associate), Time line, Location, and Position in the hierarchy. Figure 3 illustrates the simulated camouflaged network of the 9/11 attack based on the belief matrix. The above figure illustrates the presence of communication among agents using the links. The belief matrix aids to build the perception for various agents based on their associated features. Agent Ziad is taken as P to begin the analysis and for the computed local neighborhood threshold value, $\varepsilon_P = 0.1$. The agents are indexed using numbers.

Table 1. Sample relationships among agents.

Fights	Callee of Al-Qaeda Summit	Pilot Program	Member of Cell_Brooklyn	Boards_AA 11
Meeting	Member of USS Cole Attack	Advanced flight training	Member of Madrid Bombing	Work_Together
Trainer	Wife of	Brother Of	Trainer	Father-In-Law
Gets Training	Flight Class	Attendee of Summit	Financial Aid	Ready For Attack

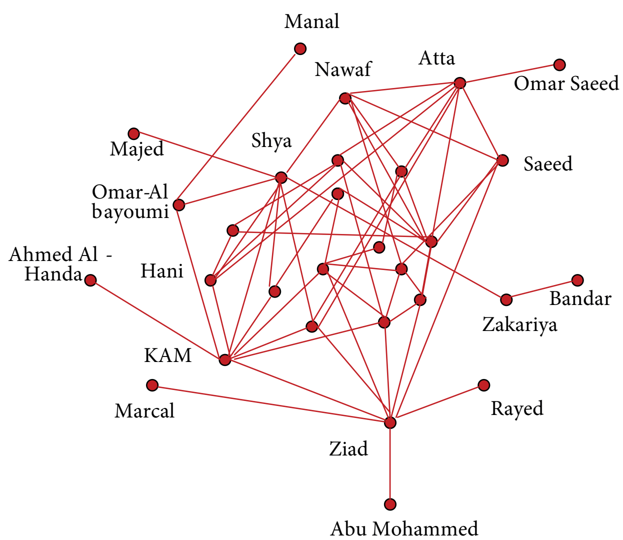

Figure 3. Sample simulation of 9/11 covert network.

Table 2 shows a sample of core nodes and corresponding overlapping nodes. The overlapping node represents that the node is a member in multiple communities. For example, overlapping node 28 is a member of core nodes 55 and 75. The evaluation of the Mutator algorithm is also presented in Table 2, which shows the evolution of new and disappeared communities for the given timeline. It highlights 25 new communities were formed and 13 communities disappeared in the year 2002.

The table shows the growth and shrink of the communities by joiners in community 50 and some lost in community 178 between 2001 and 2002. For instance, in the table, the events such as growth and shrink that occurred in communities 50 and 178 are presented by increases and decreases in neighboring nodes. The table shows sample split of the communities and merge of communities 1, 12, 13, 14, 2, 11, and 15 as community 41.

The impact of influencer analysis is exhibited using the AgentIsolator algorithm that tags the cutset of

Table 2. Sample of community evolution and related operations.

<table>
<tr><td>Date</td><td colspan="9">From:2001-01-01 To:2001-12-31</td></tr>
<tr><td>ε_P</td><td colspan="9">0.1</td></tr>
<tr><td>Community</td><td>Core</td><td>Neighbor</td><td>Overlapping</td><td colspan="6">Community analysis operations</td></tr>
<tr><td>Number</td><td>Nodes</td><td>Nodes</td><td>Node</td><td>Split</td><td>New</td><td>Growth</td><td>Shrink</td><td>Merge</td><td>Death</td></tr>
<tr><td>1</td><td>23</td><td>30, 38, 50, 55,</td><td>38, 55</td><td>50, 178</td><td rowspan="5">8, 10, 50, 95, 108, 171</td><td rowspan="5">50</td><td rowspan="5">178</td><td rowspan="5">Communities: 1, 12, 13, 14, 2, 3, 8, 11, 15 MERGED as 41 Communities: 10, 14, 1, 2, 7, 8, 15 MERGED as 42 Communities: 5, 2, 6, 9 MERGED as 16</td><td rowspan="5">23, 38, 55, 56, 151, 192, 236</td></tr>
<tr><td>2</td><td>38</td><td>54, 75, 55, 56</td><td>54, 56, 75, 55</td><td>50, 178, 8</td></tr>
<tr><td>3</td><td>50</td><td>54, 61, 23,</td><td>54</td><td></td></tr>
<tr><td>4</td><td>55</td><td>38, 28</td><td>28, 38</td><td></td></tr>
<tr><td>5</td><td>56</td><td>75</td><td></td><td>5, 8</td></tr>
</table>

agents. The cutset size is determined based on the frequent path length upon which an agent can influence or control the network. Hence, 6 is assumed as the cutset size. Another reason for choosing this cutset size is to overcome the issue of smaller cutset size. The accuracy of the chosen predominant cutset is derived through the data collected from the 9/11 commission report presented by the National Commission on Terrorist Attacks Upon the United States created by Congress and the President of the United States (Public Law 107-306, November 27, 2002). According to the report the agents {Nawaf, Atta, KAM & Hani} are identified as the key players among the 19 hijackers based on the features like {Training in military and flight aviation & Order of Hierarchy}.

The predominant cutset is chosen based on the D_F factor, which shows the cohesiveness and reachability among the agents. It gives the highest D_F value and presents the least cohesion among the network. The predominant cutset is found among the following agents, namely Nawaf (12), Atta (18), KAM (23), Ramzi (38), KBA (36), Hani (50), Wail (51), Marwan (50), and Ziad (55) for the timeline between 2000 and 2001. Table 3 shows the sample of the predominant cutsets and their corresponding D_F measure.

The cutset of size 3 with the agent set {Nawaf, Atta, and KAM} retains the network in a highly cohesive way, with a D_F measure of 0. The cutset of size 6 provides the maximum disruption to the cohesiveness of the network. The agent cutset {Nawaf, Atta, KAM, Ramzi, Ziad, Hani} has the maximum D_F value of 0.9. This cutset has a combination of agents like Ramzi who communicates with the high level leaders; Ziad and Hani, who are the intermediate agents, could communicate with the lower levels also. The covert network is fragmented by removing the various cutsets generated, and the status of the disrupted network is shown in Table 4.

The table varies the cutset sizes among 4 to 6 agents and ranges the degree of cohesion ($D_{cohesion}$) as tight, moderate, and loosely coupled. The experiments conducted results in different number of components (C) for each cutset size. The average size of component (N_{Avg}) presents the number of agents in each fragment. The D_F is computed based on C and N_{Avg} metrics. In the case of the cutset size as 4 with a tight cohesion, the fragmentation resulted in 2 components where one fragment has minimum number of agents as 20 and the

Table 3. Sample of predominant cutsets and corresponding DF.

Cutset size	Agents	D_F
4	{Nawaf, Atta, KAM}Ramzi	0.3
	{Nawaf, Atta, KAM}KBA	0.1
	{Nawaf, Atta, KAM}Marwan	0.2
5	{Nawaf, Atta, KAM}Ramzi, Ziad	0.5
	{Nawaf, Atta, KAM}Hani, Ziad	0.47
	{Nawaf, Atta, KAM}Wail, Marwan	0.4
6	{Nawaf, Atta, KAM}Ramzi, Wail, Marwan	0.8
	{Nawaf, Atta, KAM}KBA, Ziad, Marwan	0.7
	{Nawaf, Atta, KAM}Ramzi, Ziad, Hani	0.9

Table 4. Degree of breaking covert network by varying sizes of agents in cutset.

Predominant cutset size	$D_{cohesion}$	C	N_{Avg}		D_F (%)
			Min	Max	
4	Tight	2	20	31	27.4
5	Moderate	3–4	5	35	42.1
6	Moderate	6–8	1	7	59.7

other has 31 agents in it. The experimentation does not lead to an ideal fragmentation where all the agents are isolated from the other agent. Hence, $D_{cohesion}$ is never loosely coupled. Figure 4 shows the overall performance of the AgentIsolator algorithm. Figure 5 presents the degree of fragmentation and the strength of the residual network based on the group fragmenter algorithm using the similarity metrics. Figure 6 indicates that maximum fragments (31) are generated by the 5-Clique algorithm with comparatively maximum size of only 4 agents. The 3-Clique algorithm results in the most strongly couple fragment with the maximum size of 24 agents in it. The Johnson similarity algorithm leads to 22 fragments with a maximum size of 4.

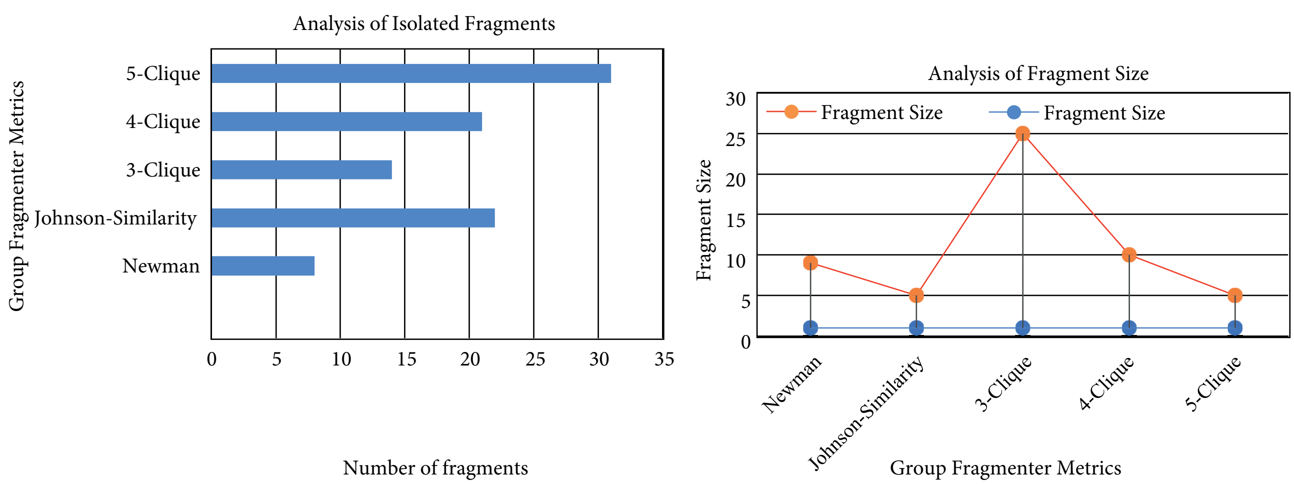

Figure 4. Performance of AgentIsolator algorithm.

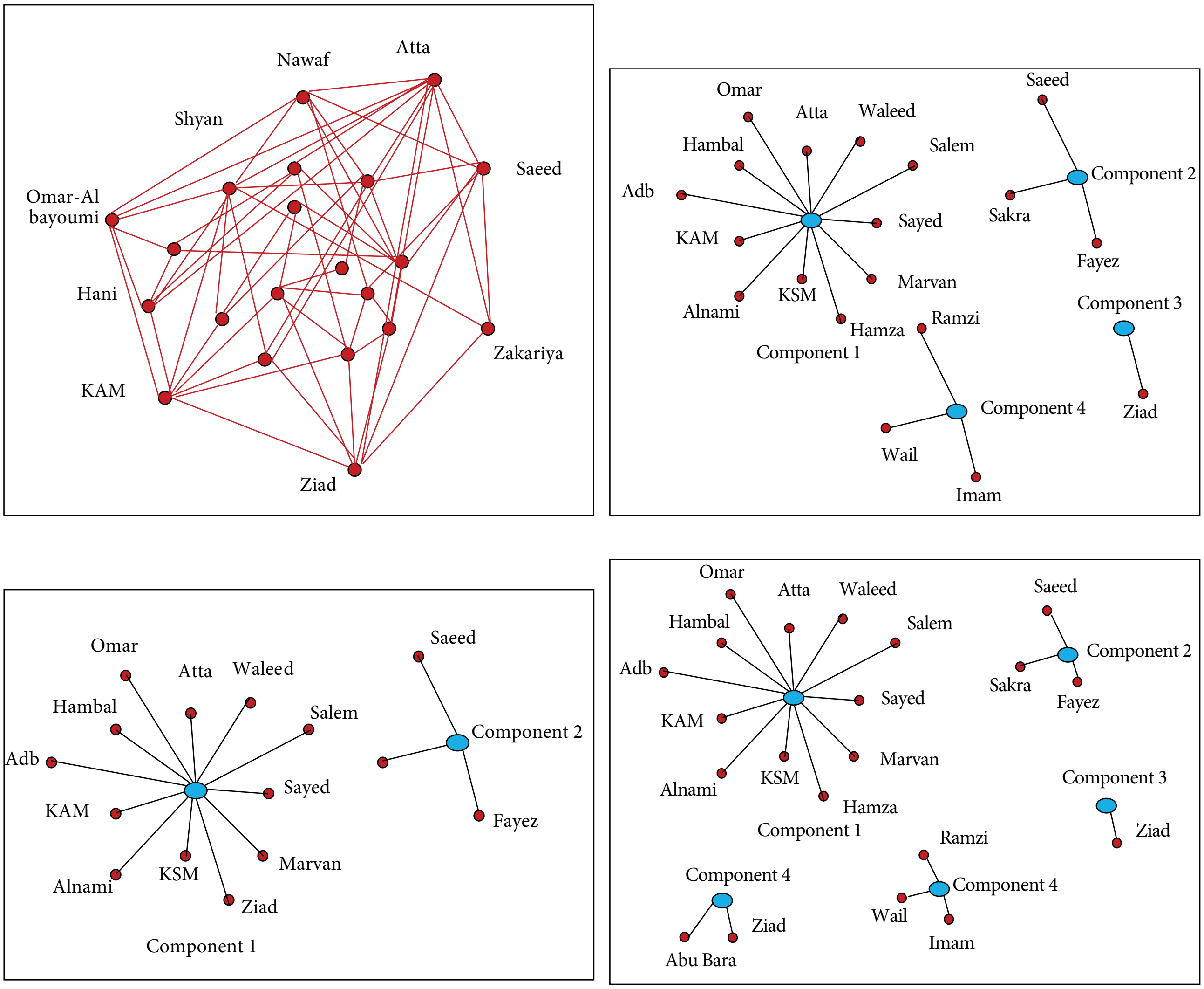

Figure 5. Sample residual network based on AgentIsolator algorithm – removal of pivot agents.

5. Conclusion

The proposed MSAS system has been able to simulate and intelligently build the 9/11 covert network through multiagents adapting the communication pattern using the knowledge base derived from the belief matrix. The novel link reciprocity metrics recognize the evolution of 25 communities and the various mutations in the structure. The identified bridging agents have been used in the predictive isolation analysis and the predominant cutset {Nawaf, Atta, KAM, Ramzi, Ziad, Hani} is tagged for giving the maximum D_F. The comparative study shows the best fragmentation is achieved using the 5-Clique metric, resulting in 31 residual fragments with only a maximum of 4 agents in the components. Conclusively, MSAS is a novel intelligent system modelled for simulating the real-time scenario of any camouflaged activities. It has been used to customize the agents to build the required scenario in order to predict the impact. This system can offer potential assistance in understanding the structure of a covert network (like cellular, hierarchical, hub & spoke) by community analysis. It can assist the disaster management team to track the influencer network and predict the changes to avert any adverse consequences. The proposed MSAS system can be transferred to microblogging services in

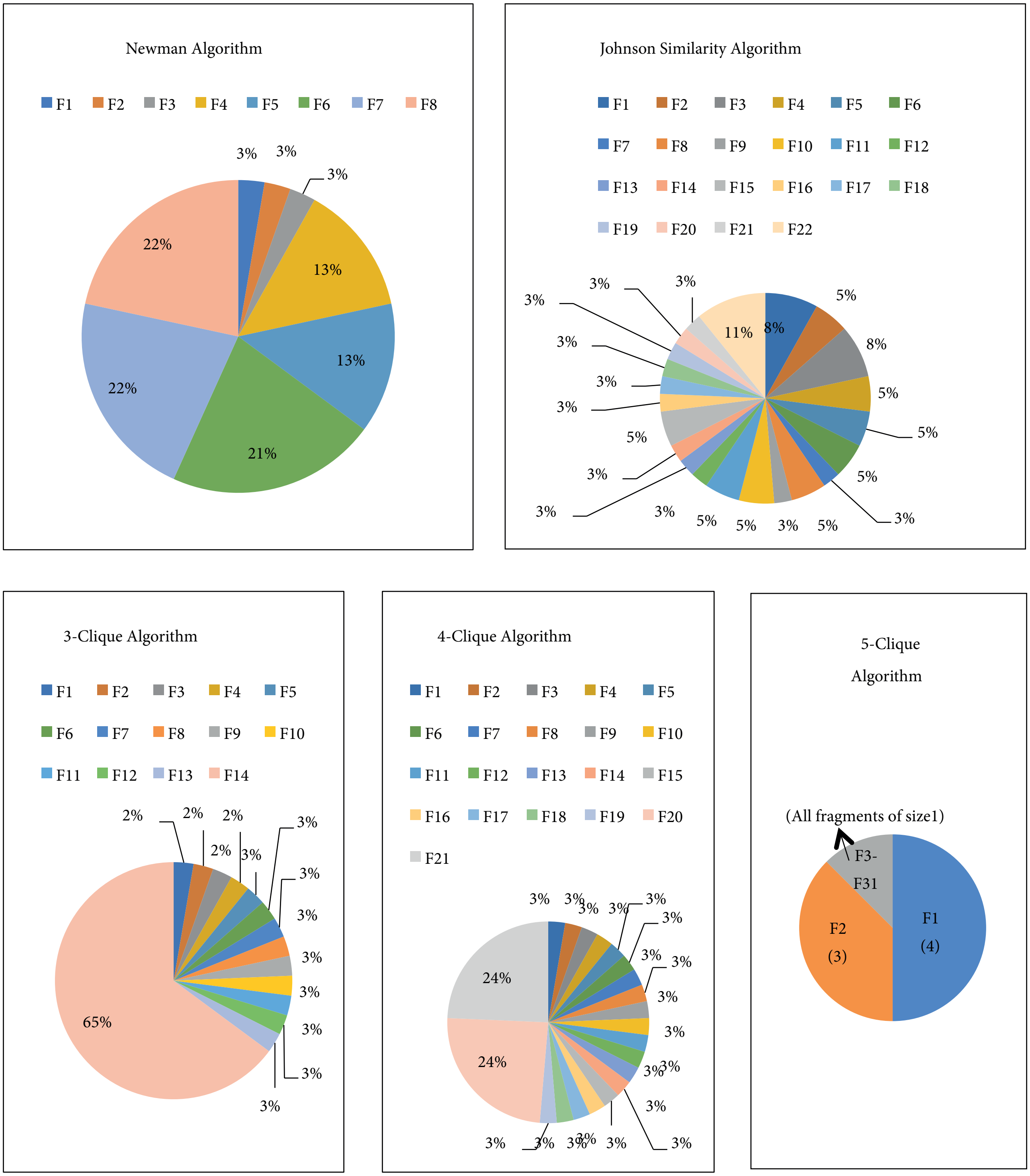

Figure 6. Residual network based on AgentIsolator algorithm.

OSN. Human behavior like posting, forwarding, or replying of messages can be analyzed and predicted based on topics and sentiments. The egocentric network can be used to present the agent interaction and simulation results. It can be further enhanced to build an autonomous trust model for the studied ego network.

References

[1] Gazendam HW, Jorna RJ. Theories about Architecture and Performance of Multi-agent Systems. University of Groningen. SOM Research Report 98 A02; 1998.

[2] Ghanem AG, Vedanarayanan S, Minai AA. Agents of influence in social networks. In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems Volume 1 Jun 4 2012; 551-558.

[3] Luck M, McBurney P, Shehory O, Willmott S. Agent technology: computing as interaction (a roadmap for agent based computing). University of Southampton, 2005.

[4] Jiang Y, Jiang JC. Diffusion in social networks: A multiagent perspective. IEEE T Syst Man Cyb 2015; 45: 198-213.

[5] Li W, Bai Q, Zhang M. Agent-based influence propagation in social networks. InAgents (ICA), IEEE International Conference 2016; 51-56.

[6] Such JM, Espinosa A, García-Fornes A. A survey of privacy in multi-agent systems. Knowl Eng Rev 2014; 29: 314-44.

[7] Clauset A, Moore C, Newman ME. Hierarchical structure and the prediction of missing links in networks. Nature 2008; 453: 98-101.

[8] Fortunato S, Castellano C. Community structure in graphs. Comput Complex 2012; 490-512.

[9] Newman ME, Girvan M. Finding and evaluating community structure in networks. Phys Rev E 2004; 69: 026113.

[10] Xu X, Yuruk N, Feng Z, Schweiger TA. Scan: a structural clustering algorithm for networks. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Aug 12 2007; pp. 824-833.

[11] Falkowski T, Barth A, Spiliopoulou M. Dengraph: A density-based community detection algorithm. In Web Intelligence, IEEE/WIC/ACM International Conference Nov 2 2007; pp. 112-115.

[12] Huang J, Sun H, Han J, Deng H, Sun Y, Liu Y. SHRINK: a structural clustering algorithm for detecting hierarchical communities in networks. In Proceedings of the 19th ACM international conference on Information and Knowledge Management Oct 26 2010; pp. 219-228.

[13] Weinstein C, Campbell W, Delaney B, O'Leary G. Modeling and detection techniques for counter-terror social network analysis and intent recognition. In Aerospace conference; Mar 7 2009; pp. 1-16.

[14] Borgatti SP. Identifying sets of key players in a social network. Comput Mqth Organ Th 2006; 12: 21-34.