Strategies of the Social Network Immunization: An Experience of an Investigation by Simulation Tools

Ilya Zimin
Student,
Perm State University, PSU
Perm, Russian Federation
ziminsrve@gmail.com

Elena Zamyatina
Associated Professor, National Research University Higher School of Economic, NRU HSE
Perm, Russian Federation
e_zamyatina@mail.ru

Abstract—**The paper considers the issues of preventing the spread of harmful information in the social networks and suggests using simulation tools to develop various strategies that reduce the risk of this information diffusion. The authors put forward requirements for a simulation system to solve such problems, provide information on the developed software and then consider its functionality using the dynamic immunization strategy as an example.**

Keywords—harmful information, diffusion, immunization, social network analysis, simulation tools.

Introduction

Currently, social networks are widespread. Social networks are both a means of communication and a means of influencing users of a social network. Social networks are successfully used by sociologists, political scientists, marketers (to promote goods, services, advertising). Unfortunately, cybercriminals and terrorists can take advantage of social networks by distributing prohibited information. For this reason, it becomes necessary to monitor prohibited information and apply various strategies to prohibit its dissemination.

Usually, to solve problems associated with managing information in a social network, one use network analysis (SNA) and the corresponding metrics (centrality, number of connections and etc.). However, using only methods of network analysis and structural characteristics of a social network cannot fully reflect the current situation, determine the cause-effect relationships of events that lead to widespread dissemination of information (including malicious). Thus, the methods of research of social networks can be divided into static methods of research of social networks [1,2,3] and dynamic, i.e those that take into account the behavior of users of a social network over time [4,5,6]. As dynamic methods, the authors of this article suggest the use of simulation methods [4,7].

One of the strategies to prevent the spread of malicious information is immunization. Immunization of a node in a social network is the process by which a node (a specific user of a social network) acquires immunity, or becomes immune to prohibited information, by blocking access to incoming messages and/or removing connections with other users of the social network and/or deleting an account from a social network.

Further, the article will be structured as follows: first, we will consider various immunization strategies, then, using the dynamic immunization strategy developed by the authors in [7] as an example, we will conduct a simulation experiment, having previously examined the requirements for a simulation software which may be used in the study of social networks.

I. Immunization

So, network immunization solves the problem of reducing the spread of malicious information on social networks. As a rule, algorithms that to some extent solve this problem can be divided into two main categories:

- Algorithms without prior knowledge of the sources of malicious information or proactive immunization algorithms; they are aimed at minimizing the spread of malicious information regardless of the presence of infected nodes; these algorithms focus on network topology and are based on the structural characteristics of social networks.

- Algorithms of the second category correspond to the immunization strategy in the presence of already infected nodes, that is, decisions on immunization are made on the basis of knowledge about the initially infected nodes

For the first category, algorithms such as targeted immunization [8] and immunization for familiarization [9] are known, they are widely used in medicine. But since these algorithms are static (immunization of nodes occurs at the beginning of the diffusion process and precedes the emergence of knowledge about malicious information on the network), such immunization is far from optimal containment of malicious information. In [7], it was proved that for the task of minimizing the influence of undesirable information, immunization of a node at a subsequent time gives better efficiency than immediate immunization. We implement the dynamic immunization algorithm model using TriadNS [10,11] simulation tools, identify the strengths and weaknesses of dynamic algorithms, and also monitor how the state of the network changes in dynamics and what it depends on.

A. The problem of dynamic immunization

In more detail, consider the task of dynamic immunization: A social network is a graph G = (V, E), where V is the set of network users, and the set of edges E represents the relationships between users. Each edge e(u, v) ∈ E has a weight pp (u, v) ⩾ 0, which indicates the probability of propagation (probability of propagation (pp)) of information between u and v (in this problem, we believe that the information can with the same probability to spread between u and v, in other words, pp (u, v) = pp (v, u)).

We assume that we know in advance how many users are the disseminators of information, and since we use virtual social networks, we will choose them randomly.

Just like in medicine, where vaccination of all people is impossible, we will not be able to immunize absolutely all users, therefore, we will set a certain immunization budget that will determine the number of nodes to be immunized during the diffusion process. An important refinement will be the fact that we will not immunize the distribution nodes, our task is to minimize the number of infected nodes by immunizing users who have not yet been affected by the information. The diffusion process stops when the immunization budget is exhausted, due to the fact that the number of immunized users has reached the maximum possible number. The continuation of the immunization process does not make sense, since we already get the optimal set of nodes, such that after their immunization, malicious information will affect the least number of nodes at the end of diffusion. This “optimal set of nodes” will be the solution to this problem.

This problem can be interpreted as follows: how to immunize K nodes so that at the end of the diffusion process the minimum number of nodes is infected. In particular, K nodes can be immunized during the diffusion process, and not at the beginning (as in the case of static immunization, where all K nodes are immunized at the 0 moment in time). The task of dynamic immunization is a generalization of the task of static immunization.

Let Im(t) be the immunization sites established at time t. Then the goal of the dynamic immunization problem is to find the optimal immunization strategy Φ = (Im (0), Im (1), ..., Im (τ)), where τ is the diffusion end time, so that the number of end nodes that are not affected by the infection is maximum.

To solve this problem, one must learn to answer the following two questions: (1) Which node should be immunized so that the effectiveness of immunization is optimal at a certain point in time? (2) Should we immunize nodes at a specific point in time or defer immunization to a subsequent point in time?

Mathematical statement of the problem: Given: G = (V, E) social network in the form of an undirected graph, P is the set of weights (probabilities), I0 is the number of malicious nodes, K is the immunization budget is the number of nodes to be immunized (K <N, where N is the number of network users). Find: Φ = (Im (0), Im (1), ..., Im (τ)), where 0,1, ..., τ are time instants, τ is the diffusion end time.

Before considering the algorithm for solving this problem, let usl determine the types of users of the social network and understand what they can do. So, let us select the actors and precedents for the information system to be under consideration.

There are three types of users in social network:

- Active - a user of a social network that is a distributor of malicious information,
- Inactive - a user of a social network that has not yet been affected by malicious information, but is susceptible to it.
- Immunized - a user of a social network that is not able to act, i.e. not involved in the diffusion process.

All users have the ability to send messages. An active user can send both regular messages and messages with malicious information. Let's call the last action as an “user activation”. If an inactive user received a message, it’s not a fact that he is being activated, because he can consider the message as spam and ignores it, delete this message and, and so on, so an inactive user can become active only with a certain probability.

Let us slightly modify the dynamic immunization algorithm presented in [7]. Consider the graph in Fig. 1. Suppose we have one active node. We need to identify a site for immunization at a specific point in time. So, let it be an undirected graph, in which there are 6 nodes and 7 edges, each edge has weight (probability of information dissemination). Active node is marked in red, other nodes are inactive.

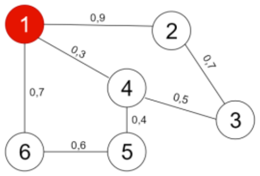


Fig. 1- An example for choosing a node for immunization

For each node, we calculate the probability of activation (probability of activation (*ap*)) – the probability with which the node can go into an active state.

We will use BFS (Breadth-first search, BFS) bypassing the graph to each node from the active one.

Further, for each inactive node u, we calculate the immunization ability (*ia*) compared to its neighboring output node v as follows:

$$ia(u,v) = ap(w) \cdot \prod_{w \neq u, w \in N_{in}(v)} \big(1 - pp(w,v)\big) - ap(w) \cdot \prod_{w \in N_{in}(v)} \big(1 - pp(w,v)\big)$$

$$ia(2,3)\ = 0.217;\ ia(3,2)\ = 0.08;\ ia(3,4)\ = 0.172;$$

$$ia(4,3)\ = 0.193;\ ia(4,5)\ = 0.195;\ ia(5,4) = 0.158;\ ia(5,6) = 0{,}222; ia(6,5) = 0{,}25$$

Finally, we summarize the immunization ability u over all its neighboring nodes and as a result we get the immunization gain (IG). The formula is very simple:

$$IG(u) = \sum_{w \in N_{out}(v)} m(u,w)$$

where $N_{out}(u)$ set of output nodes from u.

$$IG(2) = 0.217;\ IG(3) = 0.18;\ IG(4) = 0.388; IG(5) = 0.38; IG(6) = 0.25$$

We got that at this moment time, node 4 should be immunized, since the node with the highest immunization gain has priority, which means that the assumption was correct.

Thus, we can calculate the value IG^t of each node at any time t. We mark the maximum IG^t as IG^t_{max} at time t.

Now we are wondering how many immunizations gains we can get in the next step, or in other words - do we need to immunize the node now or defer immunization to the next moment time? The answer to this question is much more difficult to give than to the previous one, because we do not know how the diffusion process will continue, because in a network with one active node (we always have one, because we go to column G' with super node I), as in fig. 2**Ошибка! Источник ссылки не найден.**, which has only 3 connections, possibly 2^n where n is the number of friend nodes, that is $2^3 = 8$ possible diffusion states at a subsequent moment in time, this is due to the fact that sending messages to all your friends each of them with a certain probability may go into an active state, or it may not happen (for example, let’s say that the node has become active – 1, remained inactive – 0, then, for 3 nodes associated with this there are 8 state options: 000, 001, 010, 011, 100, 101, 110, 111). For clarity, fig. 2 shows the diffusion state at time t = 0 and all possible states at time t = 1.

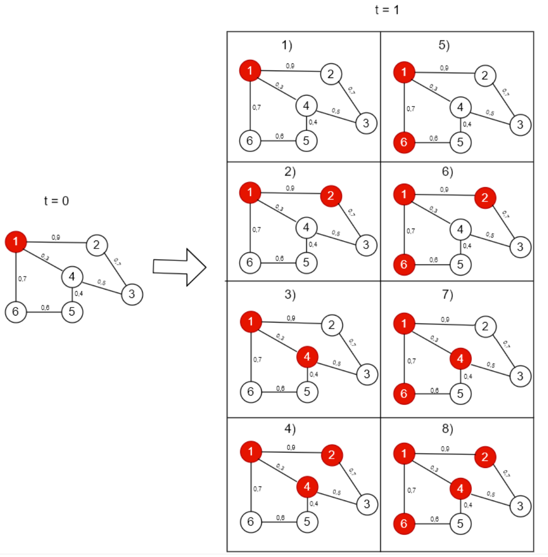
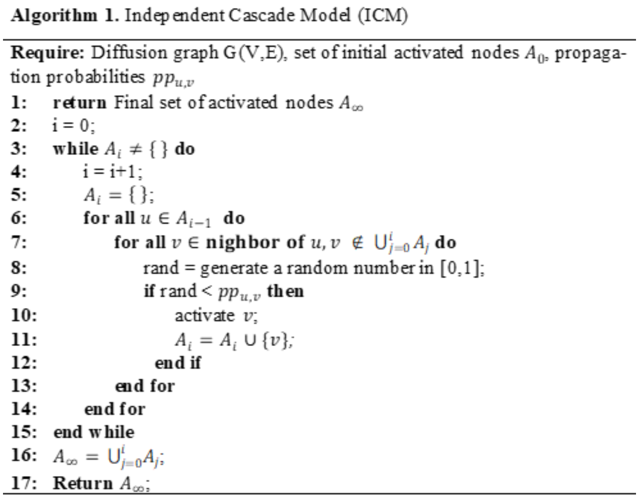


Fig. 2 – Possible diffusion states at time t + 1

to determine whether it is necessary to postpone the immunization of a node to a later moment time, consider one of the most popular models of information distribution - the Independent Cascade Model (ICM) [13]. The ICM pseudocode is shown in fig. 3.

The main idea of an independent cascade model is that active nodes send messages with malicious information to all their friends, and those, in turn, are likely to join the active ones. For our example, the process of disseminating information may look, for example, as shown in fig. 4. At each step, the distribution nodes have the only chance to infect inactive nodes associated with them. The probability with which this can happen is randomly selected and is shown in bold in the figure. The process stops when at a certain step no node was infected.



Algorithm 1. Independent Cascade Model (ICM)

```
Require: Diffusion graph G(V,E), set of initial activated nodes $A_0$, propagation probabilities $pp_{u,v}$
return Final set of activated nodes $A_\infty$
i = 0;
while $A_i \neq \{\}$ do
    i = i+1;
    $A_i = \{\}$;
    for all $u \in A_{i-1}$ do
        for all $v \in$ nighbor of $u, v \notin \cup_{j=0}^{i} A_j$ do
            rand = generate a random number in [0,1];
            if rand < $pp_{u,v}$ then
                activate $v$;
                $A_i = A_i \cup \{v\}$;
            end if
        end for
    end for
end while
$A_\infty = \cup_{j=0}^{i} A_j$;
Return $A_\infty$;
```

Fig. 3 – Pseudocode IC dissemination model

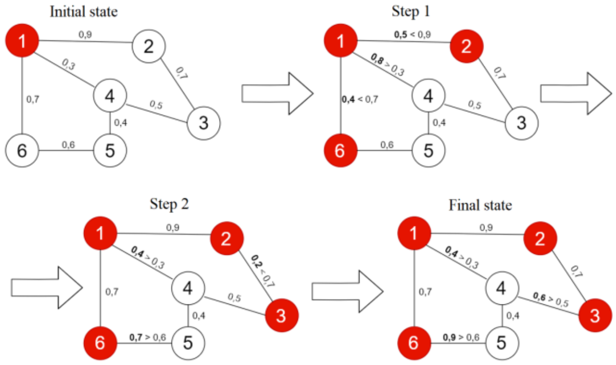


Fig. 4 – IC model of information dissemination by example

The presented model allows us to transfer the social network to a new diffusion state (at time t + 1), in which immunization gain is again calculated for each node, determine the maximum IG^{t+1}_{max} and compare it with immunization gain at time t.

We will use the Monte Carlo simulation (the process is modeled using a random variable generator, this is repeated many times, and then, based on the received random data, the probability characteristics of the problem are solved), In particular, we first run the IC model R times (randomly selected) at time t and get R states of the diffusion process at time t + 1. Then we get the average value $\overline{IG^{t+1}_{max}}$, calculating IG^{t+1}_{max} for each state and only then draw conclusions about whether or not to delay immunization.

The criterion for the need to immunize the node at the current time t: If $IG^t_{max} \geq \overline{IG^{t+1}_{max}}$, then node, which provides the maximum boost to immunization, is immediately immunized (all connections with other nodes are deleted, the top of the graph is isolated).

If immunization was performed at time t, then all the steps described above are repeated at the same time, while the criterion is met, otherwise, the current time increases., At the next time, by starting the IC model once.

In fig. 5 for illustrative purposes, a block diagram of the dynamic immunization algorithm described above is shown.

In fig. 6 shows the pseudocode of this algorithm, which greatly helped in its implementation.

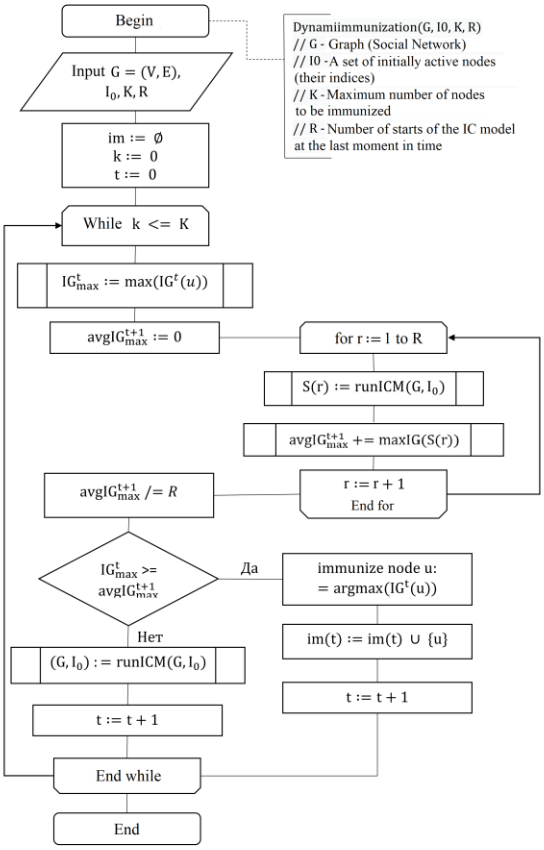

Begin
Input G = (V, E), I0, K, R
im := ∅
k := 0
t := 0
While k <= K
IGtmax := max(IGt(u))
avgIGt+1max := 0
for r := 1 to R
S(r) := runICM(G, I0)
avgIGt+1max += maxIG(S(r))
r := r + 1
End for
avgIGt+1max /= R
IGtmax >= avgIGt+1max
Да
Нет
immunize node u: = argmax(IGt(u))
im(t) := im(t) ∪ {u}
t := t + 1
(G, I0) : = runICM(G, I0)
t := t + 1
End while
End
Dynamiimmunization(G, I0, K, R)
// G - Graph (Social Network)
// I0 -A set of initially active nodes (their indices)
// K - Maximum number of nodes to be immunized
// R - Number of starts of the IC model at the last moment in time

Fig. 5 – Flow chart of a dynamic immunization algorithm

```
Algorithm 2. DynamicImmunization($G$, $I_0$, $K$)
Input: $G = (V,E), I_0$ and $K$;
Output: $\Phi = (Im(0), Im(1), \dots, Im(\tau))$
Initialize $Im(i) = \emptyset$; $k = 0$; $t = 0$;
while $k \leq K$ do
    $IG^{t}_{max} = \max(IG^{t}(u))$;
    for all $r = 1$ to $R$ do
        run IC_Model to get state $S(r)$;
        $IG^{t+1,r}_{max} \leftarrow$ maximum $IG$ in state $S(r)$;
    end for
    $\overline{IG^{t+1}_{max}} = \frac{1}{R} \cdot IG^{t+1,r}_{max}$;
    if $IG^{t}_{max} \geq \overline{IG^{t+1}_{max}}$ then
        Immunize node u = arg max($IG^{t}(u)$);
        $Im(t) = Im(t) \cup \{u\}$;;
        k = k+1;
    else
        run IC_Model;
        t = t+1;
    end if
end while
```

Fig. 6 – Pseudocode of the dynamic immunization algorithm

II. MODELING THE DYNAMIC IMMUNIZATION PROCESS

So, to study the process of disseminating information on social networks, both static research methods (SNA) are used (focusing on the structural characteristics [1,2] of social networks), and dynamic [4,5] (using simulation software). Let us consider what characteristics these software tools should have in order for the constructed simulation model to best meet the objectives of the study.

A. *Requirements for simulation systems in the study of social networks*

Currently, there are a large number of specialized software systems that are designed to study social networks. Consider the possibility of using the simulation system TriadNS [11] for the study of social networks. The following requirements can be presented to the software tools of social network simulators:

• The simulator must be able to simulate the behavior of users of social networks. Users are in the nodes of a computer network and exchange information with each other over communication lines, sending and receiving messages. In this case, it is most appropriate to use the agent modeling paradigm.

• The simulator must have software tools for building Internet graphs. When building Internet graphs, it is necessary to comply with the properties of the generated graph to the properties of real social networks [12].

• Another criterion is the flexibility of software tools that allow you to quickly change the parameters of the models of Internet graphs.

• The simulator must have software tools for researching Internet graphs.

• The simulator must be able to work with large amounts of data, ie must have software that allows you to use the computing power of several computing nodes or processors.

We will consider what properties of the above have the TriadNS computer network simulator, namely, we point out the features of the representation of the simulation model in this software.

B. *Simulation model at TRIADNS*

The computer simulator TriadNS was developed based on the Triad computer-aided design and simulation system [10]. Initially, the Triad simulation system was intended for the design and simulation of computing systems.

TriadNS adopted a three-level representation of the simulation model: M = (STR, ROUT, MES), where STR is the structure layer, ROUT is the routine layer, MES is the message layer. A layer of structures is a collection of objects interacting with each other by sending messages. Each object has poles (input and output), which respectively serve to receive and transmit messages. The basis of the representation of the layer of structures is graphs. As the vertices of the graph we will consider users of the social network. In the future, we will call them agents (agent-based modeling paradigm).

The arcs of the graph define the relationships between agents. The simulation model has a hierarchical view. Individual vertices of the graph can be decoded by a subgraph of a lower level, etc. This property of the TriadNS simulator can be useful in exploring a user group (exploring relationships within a user group).

To build the structure of the simulation model, a special procedure is used, the syntax of which is given below:

structure <*structure name*> ***def*** (<*list of settings*>) (<*list of input and output parameters*>) <*variable description*> <*operators*>) ***endstr***

The procedure is parameterized. As a parameter, you can use the number of vertices of the graph and configure the graph of the social network, changing the number of its participants during the simulation experiment. In addition, to construct a network graph, you can use graph constants that correspond to the main topologies of computer networks (chain, ring, lattice, etc.) and operations on graphs (adding, removing vertices, adding, removing arcs, edges, etc.).

To study social networks and represent their structure, it is customary to use random graphs (Erdési-Renyi graph (Fig. 7), Barabashi-Albert, etc.). Parameterized procedures for constructing these graphs are also implemented in TRIADNS.

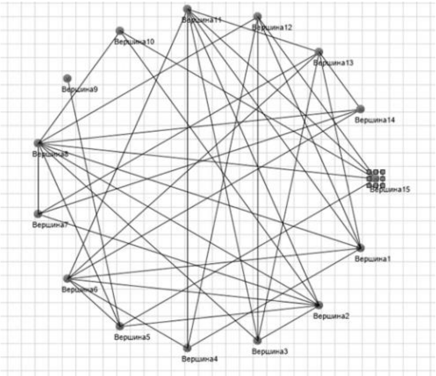


Fig. 7 – Erdési-Renyi graph

To study graphs, TRIADNS developed procedures with which you can determine the degree of a vertex, the diameter of a graph, the shortest path between two vertices, etc. To study a random graph, specially developed procedures are used that determine: (1) mutual orientation — the property indicates whether the relationship between the vertices is binary (whether the connection is bidirectional); (2) Homogeneity - indicates the degree of appearance of bonds between similar agents (by gender, age, interests) [13]; (3) transitivity of bonds — an increase in the likelihood of bonds between agents that have bonds with the same peaks [14]; (4) the difference in distribution - indicates a large number of bonds for some agents and minimal for others; the important phenomenon in this case is the “rich becomes richer” phenomenon, which leads to a high dispersion of vertices; (5) centrality - a metric that allows you to determine the significance or influence of a particular node or group in a network [2].

Agents act according to a certain scenario, which they describe with the help of a routine. A routine is a sequence of ei events planning each other (E is a set of events; a set of routine events is partially ordered in model time). The execution of the event is accompanied by a change in the state of the object. The state of the object is determined by the values of the variables of the routine. Thus, the simulation system is event-driven.

A routine, like an object, has input and output poles. The input poles are used for receiving messages, and the output poles are for transmitting them. In many routine events, the input event ein is highlighted. All messages that arrive at the input poles of the routine are processed by the input event.

Routine <Name> {<*A section of parameters*> | <*A definition of poles*>} [<*A section of initialization of routine*>] {<*A description of events*>} ***EndRout***

The experience of using TriadNS software tools for modeling messages that are generated at the output poles of a routine is carried out by ordinary routine events. To send a message, use the special out operator (***out*** <*message*> ***through*** <*pole name*>). A set of routines defines a ROUT routine layer.

The message layer (MES) is intended to describe messages of complex structure.

Next, we consider an example of the implementation of the dynamic immunization algorithm and present the results of a simulation experiment.

C. Implementation of the dynamic immunization algorithm of TRIADNS simulation system tools

In fig. 8 social network is presented, which is built in the graphical editor of the simulation system TRIADNS. This is a network of 6 nodes where the server is presented (needed to send data to all users); 5 nodes - users of a social network.

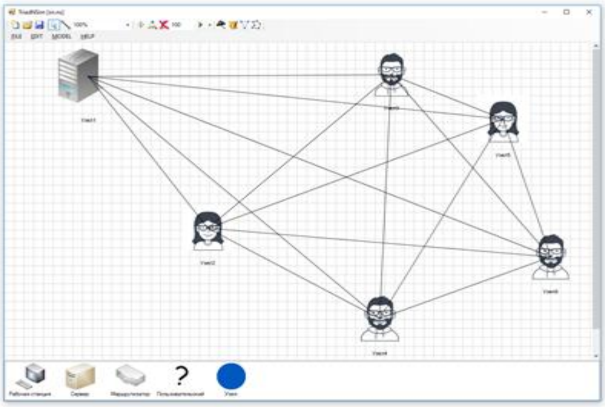


Fig. 8 – Social network in TriadNS

The immunization algorithm is implemented as follows:

- The server generates an array of distribution probabilities and sends it to all network users.
- Next, all active nodes “declare” themselves. They send messages to all their friends so that they remember their identifiers (IDs).
- After that, all inactive nodes consider their IG-criterion for the need for immunization and send it to the server, it selects the maximum.
- Next, the server calculates IG at a subsequent point in time and compares it with the current one. If the current

maximum is greater than at the next moment in time, then the server sends the owner with the highest immunization gain a message with the message "immunization", otherwise the server sends letters to all active nodes, and they begin to spread unwanted information.

- The process is then repeated.

The results of the simulation experiment showed the following: let I_0 – the number of nodes with malicious agents; K – the immunization budget. Run the model on each of the data sets 10 times and present the average value on the graphs as the result. All experiments were carried out on a 20-node network with a random type of node connections.

In fig. 9 is a graph showing how the final number of active nodes depends on the immunization budget.

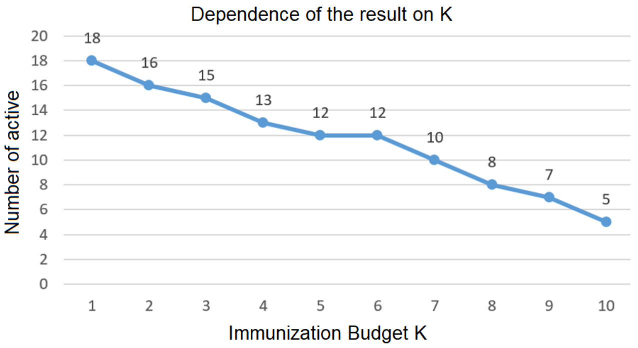


Fig. 9 – analysis at various K

In fig. 10 is a graph showing how the number of nodes for immunization in the first step depends on I_0.

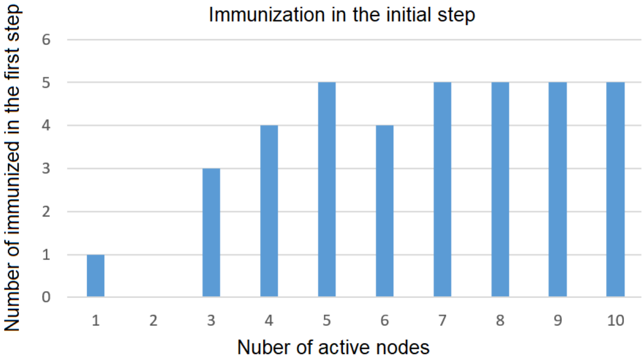


Fig. 10 – analysis at various I

Conclusion

In this paper, the diffusion process in social networks was examined in detail. The dynamic immunization algorithm has been disassembled. A simulation model of this algorithm was constructed in the TriadNS simulation environment. A detailed description of the development of this model was given.

The algorithm was tested on various sets of input parameters and well-grounded regularities were determined: with an increase in the number of initially active nodes, immunization is performed at earlier stages because the speed of information dissemination also increases, so you can "intercept" its distribution as early as possible. The larger the immunization budget, the more nodes remain uninfected at the end of diffusion, because each immunized node can “protect” from one to n nodes, where n is the number of friends of this node.

Acknowledgment

The reported study was funded by RFBR and the Krasnodar Region Administration, project number 19-47-230003.

References

[1] D. Gubanov, A.Chkhartishvili: A conceptual approach to the analysis of online social networks. Large-Scale Systems Control, pp.222−236, 2013

[2] V.A.Davydenko, G.F.Romashkina, S.N.Chukanov,: Modelirovanie sotsial'nkh setei. Vesntik TSU, 68–79, 2005

[3] N.Zhao, X.Cheng, X.Guo: Impact of information spread and investment behavior on the diffusion of internet investment products. Physica A: Statistical Mechanics and its Applications 512, 427–436 (December 2018)

[4] M. Gatti, A.P.Appel,C.Pinhanez, C.Santos, D.Gribel, P.Cavalin, S.B.Neto: Large-Scale Multi-agent-Based Modeling and Simulation of Microblogging-Based Online Social Network. Multi-Agent-Based Simulation XIV. MABS, 17−33 (2014)

[5] Y.Zhang, J.Zhu: Stability analysis of I2S2R rumor spreading m.odel in complex networks. Physica A: Statistical Mechanics and its Applications 503, 862–881 (August 2018)

[6] Y.Zan, J.Wu, P. Li, Q.Yu: SICR rumor spreading model in complex networks: Counteratta.k and self-resistance. Physica A: Statistical Mechanics and its Applications 405, 159–170 (July 2014)

[7] D.Yang, X. Liao, H. Shen, X.Cheng, G.Chen: Dynamic node immunization for restraint of harmful information diffusion in social networks. Physica A: Statistical Mechanics and its Applications 503, 640–649 (August 2018)

[8] Y.Zhang., B.A.Prakash. Dava: Distributing vaccines over networks under prior information, in: Proceedings of the 2014 SIAM International Conference

[9] R.Pastor-Satorras, A.Vespignani: Immunization of complex networks, Phys. Rev. E 65 (3) (April, 2002) 036104.

[10] E.B.Zamyatina, A.I. Mikov, R.A.Mikheev: TRIADNS Computer Networks Simulator Linguistic and Intelligent Tools. International Journal “Information theories & Applications" (IJ ITA), 355–368 (2012)

[11] E. B. Zamyatina, A. I. Mikov: Programmnye sredstva sistemy imitatsii Triad.Net dlya obespecheniya ee adaptiruemosti i otkrytosti. Informatizatsiya i svyaz', 130–133 (2012)

[12] A. M. Raigorodskii.: Proceedings of Moscow Institute of Physics and Technology (State University). In : Random graph models and their application, pp.130–140 (2010)

[13] M.E.J. Newman: A measure of betweenness centrality based on random walks. URL: http://aps.arxiv.org/pdf/cond-mat/0309045.pdf (дата обращения: 10.06.2020).

[14] M.E.J.Newman: The mathematics of networks. URL: http://wwwpersonal.umich.edu/~mejn/papers/pal grave.pdf (дата обращения: 13.06.2020).