Computer Science and Information Systems 18(3):703–728 https://doi.org/10.2298/CSIS200415050F

A Novel Information Diffusion Model Based on Psychosocial Factors with Automatic Parameter Learning

Sabina-Adriana Floria and Florin Leon

Faculty of Automatic Control and Computer Engineering,
"Gheorghe Asachi" TechnicalUniversity of Iași,
Bd. D. Mangeron 27, 700050, Iași, Romania
{sabina.floria, florin.leon}@ tuiasi.ro

Abstract. Online social networks are the main choice of people to maintain their social relationships and share information or opinions. Estimating the actions of a user is not trivial because an individual can act spontaneously or be influenced by external factors. In this paper we propose a novel model for imitating the evolution of the information diffusion in a network as well as possible. Each individual is modeled as a node with two factors (psychological and sociological) that control its probabilistic transmission of information. The psychological factor refers to the node's preference for the topic discussed, i.e. the information diffused. The sociological factor takes into account the influence of the neighbors' activity on the node, i.e. the gregarious behavior. Agenetic algorithm is used to automatically tune the parameters of the model in order to fit the evolution of information diffusion observed in two real-world datasets with three topics. The reproduced diffusions show that the proposed model imitates the real diffusions very well.

Keywords: social networks, information diffusion, psychological factors, sociological factors, genetic algorithm.

1. Introduction

The use of social networking websites is currently the most widely used form of communication. Social networks help us to keep in touch with our friends, create new relationships, develop our social life, but these can also influence our decisions especially when a lot of information is false or we may even become dependent through their excessive use. Socialization on social networks has taken on a great extent and the number of users has increased considerably. When analyzing social networks, we focus on discovering patterns of human interactions. Thus, we can observe the social structures, and the actions and friends of an individual are no longer random, but can be modeled according to well-defined rules. If we were to analyze how people interact, we might find that they do not make random connections with one another (e.g. some talk constantly, others often, and some never do). The use of social networks allows users to send and receive messages (both public and private), share photos, videos and gives them the opportunity to join certain groups. We can say that online communication means have become employed in all aspects of everyday life, from business to social life. Thus, online social networks not only connect many users, but also collect data on their daily interactions. Given these collections, we can analyze how information is transmitted through social networks, which is a topic of great interest.

In this paper we propose a model that imitates the information diffusion as well as possible, referring to the behavior of users from a real point of view, both at the psychological and social level. In modeling the individual from a psychological point of view, his/her decision in the transmission of information is a restricted one. This restriction consists in the fact that the individual has a certain interest in the information, but this information may also be useful or not for him/her. In modeling the individual from a sociological point of view, we consider the gregarious behavior of the individual regarding the transmission of information, namely that he/she is influenced by the activity of his/her neighbors. For both models, we take into account both the information credibility, and the fact that users are bored with certain information, either according to the passage of time or depending on the amount of information received. We also use a genetic algorithm to automatically learn the model parameters based on the real information diffusion. This model is applied on two real datasets [1, 2], and the results are promising.

In the literature there are many factors considered to influence the information diffusion, but it is important how these factors are modeled and combined. In our model we introduce the following influencing factors: the boredom of an individual, the activation degree of an individual that is correlated with his/her interest in information and the usefulness of the information, the login rate depending on a certain time of day (daytime/nighttime) etc. The main contribution of this paper is the combination of these influencing factors in a model in order to reproduce the decision of an individual in an agent-based simulation as well as possible. Another contribution is the automatic learning of certain parameters of the diffusion model. This is a great advantage because the evolutions of the diffusions may be different depending on a certain topic being discussed. Moreover, by combining psychosocial modeling with automatic learning of parameters, our model is able to extract the relevant characteristics from various evolutions of diffusion. To the best of our knowledge, no other models have been proposed so far in the literature that combine these influencing factors and automatically adapt their parameters for different diffusion evolutions.

The paper is organized as follows. In Section 2 we present some contributions related to information transmission models. In Section 3 we describe our model, and the experimental results are shown in Section 4. Finally, in Section 5 the conclusions and some development directions are included.

2. Related Work

An analysis of how information is spread through online social networks (OSNs) and simulation of user behavior based on their posts is presented in [3]. The method proposed in this paper uses a stochastic multi-agent approach, in which each agent is in fact a user of the social network. The analysis is made on Barack Obama’s Twitter network in the 2012 US presidential race. The authors show what happens if the central source of the network is inactive, more precisely the node that represents Barack Obama and also highlights the impact of eliminating the most active users in the process of information diffusion. By impact it is understood that the number of messages sent by users is constantly affected over time. Experimental results show that eliminating the first 100 most active users has a greater impact on the number of messages than removing the central source node.

In [4], the authors model the information diffusion using agents with well-defined states, similar to the epidemic SIR (Susceptible, Infected, Recovered) model and use two datasets from the Twitter social network to compare the efficiency of the proposed model regarding the realistic simulation of the diffusion. The model introduced in this paper is based on the fact that those users who may know that a rumor is false, will not spread messages that deny these rumors. Therefore, recovered users will not influence their neighbors, allowing them to recover as well. They use a synthetic scale-free network of 1000 nodes and the Euclidean distance to evaluate the difference between the actual and the simulated diffusion results. The authors compare the Euclidean distance of their model with a basic model and obtain a smaller distance for both datasets, so a more realistic information diffusion.

A basic model for rumor propagation is proposed in [5] and consists of node-level modeling. Nodes can have well-defined states, each state allowing specific actions such as spreading the information, ignoring it etc. As in our work, node activity is modeled in discrete time events, which is why the authors can model various time constraints, such as: some actions of the nodes are completed after a certain period of time, a node checks its information from friends at least once 24 hours and at most once an hour. The proposed model contains a large number of parameters, this being a general impediment in the complex models, hence our motivation to use a genetic algorithm for automatic parameter learning. The authors also use synthetic networks and conclude that those networks with scale-free topology are more suitable for analyzing the simulation of information diffusion.

In [6], a multi-agent model is proposed to reproduce the real transmission of information in scale-free networks. In addition, the authors also propose a mechanism to combat the spread of false information. Each agent has the opportunity to choose whether or not to transmit the information depending on his preparation level on a particular topic, which is a random threshold assigned to him. The authors propose three different ways in which an agent can spread information: spontaneous visualization, collective influence and communication persuasion. An analysis of the real information diffusion is made on a Twitter dataset with the announcement of the discovery of the Higgs Boson [1] in which the authors track the activity of the active users in the network to highlight the evolution of information diffusion. We also follow this aspect in our paper on the same dataset. Running the model on networks of different sizes, the authors observe the same form of diffusion and assume that their model does not depend on the size of the network, but only on the simulation parameters. Also, to study the spread of false information, the authors use the real dataset where fake news was spread during the Occupy Wall Street protest. In order to model the spread of fake news, the authors introduce in their model a new type of agent that is able to recognize fake news and alert its neighbors. In this experiment, based on the number of posts of users over time, they obtain a good dynamic of the event on networks of different sizes.

A spatio-temporal characterization of the information diffusion process and a model that describes the dynamics of information spreading on the Higgs dataset [1] are presented in [7]. Regarding the spatial and temporal characteristics of the observed data, the authors studied the behavior of the user both at the global (macroscopic) and individual (microscopic) levels. The users' activities are: posting message (tweet), sharing post (re-tweet) or replying to existing tweets. Starting from the observed characteristics, the proposed diffusion model takes into account the fact that a user no longer posts a certain period of time after having a recent post. Also, the authors introduce two different rates of activation or deactivation of nodes that are time-varying and can be independently modified. The probability that a node will post a message is also influenced by the number of its neighbors who repeatedly post over time. Their model has a good accuracy in reproducing the information diffusion and could be applied in other processes of diffusion of social networks.

A protocol in which the network becomes more immune to the spread of false information based on the evidence theory (Dempster-Shafer theory and Yager's rule) is presented in [8]. Their model is based on the choice of two source nodes, one that transmits true information and one that transmits false information. The effects of the collision of the two pieces of information through the network are shown, but also the effect of using the evidence after establishing the ground truth. This approach based on the evidence theory plays an important role in the individual's decision to transmit or not the received information. The authors also consider the confidence degree of the neighbors regarding the character of the information spread by a certain source. Once the ground truth is established, the authors show how the spread of false information is blocked. Also, the work [9] is an extended version of the work [8] in which the following case studies are considered: different positioning of the source nodes, a source node might not always transmit the same information during a simulation, use of a larger network, adding new connections to the original networks and analyzing the number of messages during information diffusion.

There are many other approaches that analyze the process of information diffusion through the network. For example, a dynamic model is proposed in [10] to investigate the influence of node activity on the information spread process. Through an active node, the authors refer to the fact that it can contact all its neighbors, while an inactive node can only communicate with its active neighbors. The behavior of the model is studied on both homogeneous and heterogeneous networks. In [11], the authors study the dynamics of the information diffusion on homogeneous social networks in which they consider a mechanism to combat false information. A stochastic model for information diffusion is proposed in [12] and the authors mention the limited attention property of the users, in which they may lose some of the received messages if they have many connections. In [13], an extension of the Susceptible-Infected diffusion model is proposed, in which the authors include elements of human dynamics, such as bursty and limited attention, with a significant impact on the diffusion process. In [14] a competitive model of information diffusion is presented, which consists in the simultaneous spread of two different pieces of information. A diffusion model called GT is presented in [15], in which the nodes are considered intelligent and rational agents and have two types of payoff: a social and an individual one. Also, the proposed model can be used to predict what behavior the users will have in a certain time frame.

Other models of information diffusion are also presented in surveys on this topic: [16-18].

3. Model Description

In this paper we propose a protocol of information diffusion in social networks in which we take into account as many realistic factors as possible in order to model the individual's decision to transmit information or not. We consider both the personality of the individual from a psychological point of view, as well as his degree of sociability. We model user behavior using two categories of influences: internal and external. These categories are presented by [19] in a detailed analysis of consumer behavior. Some examples of internal influences of a consumer's behavior presented by the authors are perception, motivation, learning, memory, attitude, and the main external influences are those of groups or different factors of a society (e.g. demographic or cultural factors). We refer to these internal factors as psychological factors of an individual, while external factors are correlated with sociological factors. These terms can be jointly referred to as "psychosocial" factors. In psychological modeling we chose the perception and motivation of an individual as internal factors: perception is modeled as the usefulness of information, while motivation is modeled as a combination of the individual's interest and the usefulness of information. In sociological modeling, we chose the external factors related to the influence of the neighbors on an individual.

In addition, we propose that an individual may be influenced by the information credibility when making a decision in its transmission. The credibility of the information or the credibility of the source of information is difficult to assess. In an online social environment, a user usually assesses credibility based on certain indicators provided by the social platform. For example, in [20] the authors analyze the relevance of certain indicators on Twitter based on which users try to assess the credibility of posts. The most important indicators are those that refer to an official source, or posts that contain links, facts, informative or professional messages. To analyze these indicators, the authors chose different evaluators to judge the credibility of the posts and used the majority vote for the final evaluations. The aforementioned credibility indicators cannot be used in our model because we do not make an analysis based on the content of the messages. However, in [20] the authors show that the number of posts is also an indicator of credibility. Therefore, in our model we propose to use the number of the neighbors' messages as an indicator of information credibility. We propose that the information held by a node has an initial credibility, which is a value in the range (0, 1]. In our model there are two ways in which a node can have information: it is assigned to it by simulation (source node or informant node) or it can receive it from neighbors (special node). The mechanism for determining credibility applies only to special nodes. Majority voting is an intuitive strategy to model an individual's decision when multiple options are available. For example, [20] and [21] use majority voting to assess the final credibility of posts, and [22] uses majority voting as a mechanism for modeling the gregarious behavior of a node. The majority vote cannot be applied in our model because the credibility of the information is not a categorical variable, but a real one. Therefore, as an alternative, we propose that a node weigh the credibility of information received from neighbors. Moreover, in this model we assume that the credibility of the information increases as it is discussed more. To achieve this growth, we increase the weight of that credibility received from the neighbor with the highest number of messages.

Users cannot always send information, but only during certain periods. We start from the premise that they log on to a social platform with a certain average login rate.

Most of the parameters of the proposed model are learned using a genetic algorithm. Therefore, having a diffusion model and the automatic learning of the parameters, our objective is to obtain an accurate evolution of the information diffusion, comparing it with the real diffusion.

The first dataset we use contains both the structure of a social network on Twitter and the activity of users during the announcement for the discovery of the Higgs Boson. The second dataset does not contain the structure of the network, but only the activity of users in the Twitter social network on certain topics, of which we have chosen two, namely: “lipstick on a pig” and “fundamentals of our economy are strong”. We chose datasets that contain the timestamps of communication between users. Based on these timestamps, we managed to extract the evolution of information diffusion. Regarding the second dataset, the subjects were chosen at random.

3.1. User Login Rate and User Handling

When a person initiates an activity on a social network, we say that by this action he or she logs in. For this purpose, we choose that users have an average login rate (λ_{login}) following a negative exponential distribution law according to equation (1), where u represents a randomly generated number in the range [0, 1) and t_{login} is the time period between two successive logins of a user expressed in minutes:

$$t_{login} = -\frac{\ln(1-u)}{\lambda_{login}} \qquad \textbf{(1)}$$

For example, if an individual logs in every two hours, then the average login rate is $\lambda_{login} = \frac{1}{2[h]} = \frac{1}{2 \cdot 60[m]} = 0.0083$ logs per minute. For this login rate, one can see in Fig. 1 on the Y axis that t_{login} is generated in an interval of approximately [0,600] minutes. We can also see that we have a higher chance of generating short duration times and we mark with the dotted line the duration of 200 minutes, i.e. in about 80% of cases we will have small values.

We consider that the nodes are handled in the order of the login duration. Thus, after the login duration of a node has been generated, it is added to a sorted list. We chose to increment the simulation clock in discrete steps, where each step represents a period of one minute. After each increment of the simulation clock, the list is checked to identify which nodes are able to log in, i.e. a minute has passed and some nodes may be able to log in. A node that is able to log in is extracted from the list and then it is checked whether it can transmit its information to its neighbors according to the two models (from a psychological and sociological point of view). Subsequently, a new time period is generated for this node and it is added back to the list, such way that the list remains sorted (i.e. the node that has the smallest login period is placed first in the list).

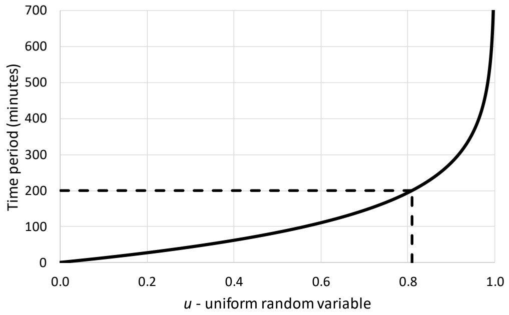


Fig. 1. Generation of time between two successive logins according to the uniformly distributed random variable *u*

Before describing how we handle nodes, we specify that in our model we have three types of nodes, namely: source node, special node and informant node. A source node holds the information and its transmission is based only on the basic probability attached to the node. Also, the transmission of a source node is not influenced by the information credibility or by the psychosocial modeling. A source node will change its type into a special node when at least one piece of information is received from one of its neighbors. If the source node does not receive any information from its neighbors, it automatically becomes a special node after a period of time to avoid the continuous transmission of messages. The second type of node, the special node, has a transmission probability that is determined according to the two models and it is also influenced by the information credibility. The last type of node, the informant node, has 100% probability of transmission and 100% credibility for information. We propose this type of node for the moment when we want to suddenly encourage socialization between the nodes, i.e. several nodes adopt the information when it is transmitted very often. Informant type nodes spread information for a certain period of time ($T_{max\_informant}$), after which they become special nodes.

3.2. Transmission of Information

In the initial phase only the source nodes hold the information along with the credibility attached to it. After a node has logged in, its decision to transmit or not the information is a probability determined according to the type of node:

$$Prob_{send} = \begin{cases} P_b \text{ , } source\ node \\ P_f \text{ , } special\ node \\ 1 \text{ , } informant\ node \end{cases} \qquad (2)$$

where P_b is the basic probability (the same for all nodes), and P_f is the final probability determined according to the psychosocial modeling. Depending on this probability, if the node has the chance to transmit the information, it will spread the information to all its neighbors along with the credibility attached. A node can transmit information multiple times due to repeated logins and also a node counts the information received from each of its neighbors separately. The special node has a particular behavior when it spreads information. For this type of node, we determine a final probabilityP_f, which is based on both the psychological and sociological modeling and also on the information credibility:

$$P_f = (P_P \cdot W_p + P_S \cdot W_S) \cdot InfoCred(n_t) \tag{3}$$

where $P_p$is the probability from the psychological modeling, $P_s$is the probability from the sociological modeling, $W_p$and $W_s$are weights that control the impact of P_p and P_s, and *InfoCred*(n_t) is the credibility that the node n_t has on the information. *InfoCred*(n_t) is computed based on all information received from the neighbors of the n_t node.

3.3. Determining the Credibility that a Node Has on the Information

In our model, a node stores statistics for each neighbor. Thus, when a node receives information from a neighbor, it stores the information credibility and also the number of messages received. The node has these two fields separately for each neighbor. We choose that the information from certain neighbors should be more important or less important depending on the number of messages received. To determine the information credibility of a transmitter node (n_t), we weight each credibility received from neighbors according to equation (4), where v represents the number of neighbors of node n, and *InfoCred*(i) is the credibility received from neighbor i:

$$InfoCred(n_t) = \sum_{i=1}^{v} W_i \cdot InfoCred(i) \tag{4}$$

The weight W_i associated with the neighbor i is computed as the ratio between the number of information received from the neighbor$i$ and the total number of the information pieces received from all the neighbors:

$$W_i = \frac{Info(v_i)}{Info_{total}} \tag{5}$$

In order to implement a mechanism for increasing the credibility of information, we choose that the maximum weight should be encouraged by a percentage increase. In other words, we take into account to a greater extent the credibility of that neighbor who transmitted the highest number of messages:

$$W^*_{\max} = W_{\max} + W_{\max} \cdot W_{cred} \tag{6}$$

$$W_{\max} = \max(W_i), i = 1, ..., v \tag{7}$$

where W^*_{max} is the maximum adjusted weight, and W_{cred} represents a parameter that controls the increase in credibility (the same for all nodes). The value of the control parameter W_{cred} is learned using the genetic algorithm.

This process can indeed be manipulated by artificially developing a high number of neighbors or another approach by spamming messages [20]. We do not use a mechanism for detecting and correcting such manipulations, but we propose a simple mechanism for combining the information that a user has from friends.

In Fig. 2 we show an example on a small network in which we determine the credibility that node 3 has for the information.

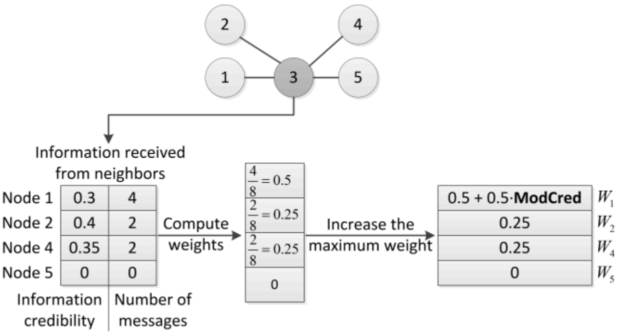


Fig. 2. Example for computing the information credibility of a node

3.4. Modeling the Individual from the Psychological Point of View

Regarding the modeling of the individual from the psychological point of view, we propose that his/her decision to spread or not the information (probability P_p) should be influenced by the individual's activation degree and an attenuation factor. We also consider the basic probability of the node (P_b) and a weight for the activation degree (W_{act}):

$$P_p = Activation \cdot W_{act} \cdot P_b \cdot \text{Attenuation}_{\mathrm{P_p}} \qquad \mathbf{(8)}$$

We propose that the node's activation degree should be correlated both with the node's interest for information and with the level of usefulness of the information. Therefore, we define the following measures: $N_{interest}$ is the level of interest that the node has for the information and $N_{usefulness}$is the level of usefulness of the information. $N_{interest}$ and N_{util} have the same definition domain: integers in the range [1, 10]. Before starting the simulation, we initialize $N_{interest}$ from each node with a random value in the range [1, 10], thus each node has its own interest for the information. $N_{usefulness}$ is attached to the information that is spreading and this measure does not differ from one node to another. $N_{usefulness}$ is also initialized with a random value in the range [1,10]. Our model is capable of spreading a single type of information during the simulation, thus the value of $N_{usefulness}$ remains constant. $N_{interest}$ and $N_{usefulness}$ are used to determine the node's activation degree and this step is done before the node transmits the information:

$$Activation = \frac{N_{interest} \cdot N_{usefulness}}{Max_{interest} \cdot Max_{usefulness}} = \frac{N_{interest} \cdot N_{usefulness}}{10 \cdot 10} \qquad \textbf{(9)}$$

where $Max_{interest}$ and $Max_{usefulness}$ are the maximum limits for $N_{interest}$ and $N_{usefulness}$. The node's activation degree is directly proportional to both $N_{interest}$and $N_{usefulness}$. Basically, the activation degree increases in greater proportion as both levels ($N_{interest}$ and $N_{usefulness}$) increase (Fig. 3).

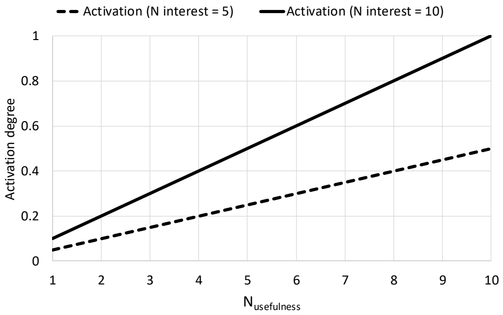


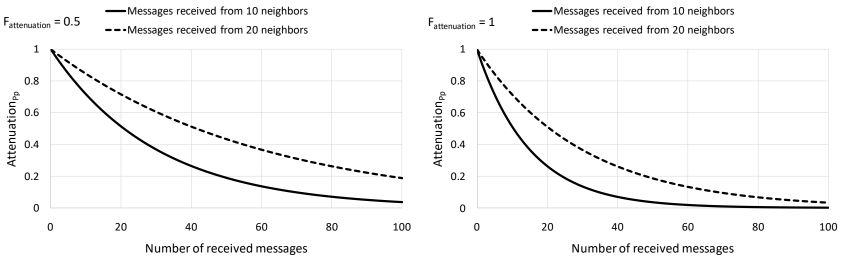
Fig. 3. Example of activation degree for $N_{interest}$=5 and $N_{interest}$ =10

Using this approach, we want to model various boundary cases. For example, when the interest of a node is very high, but the usefulness of the information is low, we can say that the activation degree has a small value. This behavior is due to the fact that although the node is very interested in information, the reduced utility of the information does not satisfy the node. The behavior is the same if $N_{interest}$ is small and the $N_{usefulness}$ is very large: the information is satisfactory, but the individual has no interest in it. We also attach a weight (W_{act}) for the activation degree in order to control the impact it has on the sending probability of the psychological model (P_p). Depending on the diffusion evolution, the value of the W_{act} weight is learned by the genetic algorithm in the range [0.2, 1].

As a topic is discussed more often, the number of messages received by the node increases. We propose that the node should get bored of the topic discussed as the number of messages received becomes larger. In other words, we introduce an attenuation (10) that depends on the total number of messages received by a node (Msg_{total}) from all its neighbors and by an attenuation factor ($F_{attenuation}$) learned by the genetic algorithm.

$$Attenuation_{P_p} = e^{-\frac{F_{attenuation} \cdot Msg_{total}}{v}} \qquad \textbf{(10)}$$

We divide the exponent into the total number of neighbors (v) because we would obtain different results for networks of different sizes. The total number of messages of a node in a small network is smaller than in a large network (where the nodes have more neighbors).



a) b)

Fig. 4. Example of attenuation for a) $F_{attenuation}$ = 0.5; b) $F_{attenuation}$ = 1

In Fig. 4 we show the attenuation evolution according to the number of messages received from a certain number of neighbors for two cases: $F_{attenuation}$ = 0.5 and $F_{attenuation}$ = 1. In the case of a node with 10 neighbors and 40 received messages, one can see that the attenuation $Attenuation_{Pp}$ changes as follows: when $F_{attenuation}$ is 0.5, $Attenuation_{P_p} \cong 0.25$ (Fig. 4.a), and when $F_{attenuation}$ is 1, $Attenuation_{P_p} \cong 0.07$ (Fig. 4.b).

3.5. Modeling the Individual from the Social Point of View

In this type of modeling we focus on the percentage of active neighbors of the node because we want the probability provided by this modeling (P_s) to depend only on the activity of its neighbors and not on the amount of information received by the node. To model this probability, we start from a sigmoid function (P_{social_infl}) to which we include an attenuation ($Attenuation_{Ps}$):

$$P_s = P_{social\_\inf l} \cdot Attenuation_{P_s} \qquad \textbf{(11)}$$

$$P_{social\_\inf l} = \frac{1}{1+e^{-(\theta + Pct_{neighbors} \cdot F_{social\_\inf l})}} \qquad \textbf{(12)}$$

We choose the sigmoid function (12) because, considering its shape, we want $P_{social_infl}$to have a lower value in the beginning, when the number of active neighbors is relatively small. Then, as users discuss more, we want $P_{social_infl}$to have a sudden increase at one point, thus modeling the social behavior of a node. In order for the evolution of the sigmoid function to start from the origin on the X axis and not from the negative domain, we introduce a parameter θ with the value –5. Also, we introduce in P_{social_infl}a social influence factor (F_{social_infl}) in order to control the shape of the curve. We show in Fig. 5 the impact of $F_{social_infl}$for two different values, 5 and 10. These values are actually the limits of this parameter.

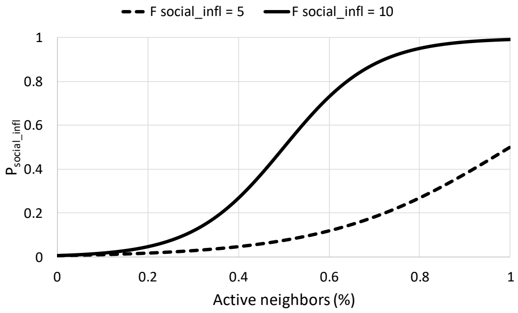


Fig. 5. Evolution of $P_{social_infl}$for two values of F_{social_infl}: 5 and 10

We choose the value 5 for the lower limit of F_{social_infl} because in this case the sigmoid function shape is incomplete and one can see that P_{social_infl}reaches the maximum value of 0.5. This is the situation in which the individual is weakly influenced by the percentage of his active neighbors. We choose the value 10 for the upper limit of F_{social_infl}because we want a complete sigmoid shape. Therefore, P_{social_infl}tends to value 1, modeling the fact that an individual is more influenced by the activity of his neighbors. The F_{social_infl}parameter, which represents the social influence of the node, is not learned by the genetic algorithm, but is randomly generated in the range [5, 10] for each node at the beginning of the simulation. As in the case of psychological modeling, we introduce an attenuation ($Attenuation_{Ps}$) in the probability P_s (11) to simulate the fact that a node gets bored with the activities of its neighbors. In P_s probability, the node does not take into account the amount of information from its neighbors, but the number of active neighbors expressed as a percentage. Therefore, $Attenuation_{Ps}$ (13) does not dependent on the number of messages received by the node, but we choose to be time dependent. However, a node does not have many neighbors active at the beginning of the diffusion, so it is important to choose a start time (T) from which we can consider that the neighbors of the node are quite active. We choose to define the moment T when W_s>W_p, i.e. when probability P_s is more important than probability P_p. We consider that this criterion is suitable because the weight of W_s is dependent on the percentage of active neighbors of the node:

$$Attenuation_{P_s} = \begin{cases} e^{-T_{elapsed} \cdot F_{social\_tolerance}} & ,W_s > W_p \\ 1 & ,W_s < W_p \end{cases} \qquad \textbf{(13)}$$

$$T_{elapsed} = \frac{Simulation\_clock - T}{T_{max\_socialization}} \qquad \textbf{(14)}$$

In Fig. 6 we show an example where the moment T is defined. The P_s probability is affected by the attenuation $Attenuation_{Ps}$ from the beginning of time T. After defining the moment T, we can also determine the elapsed time ($T_{elapsed}$) to evaluate the degree of attenuation.

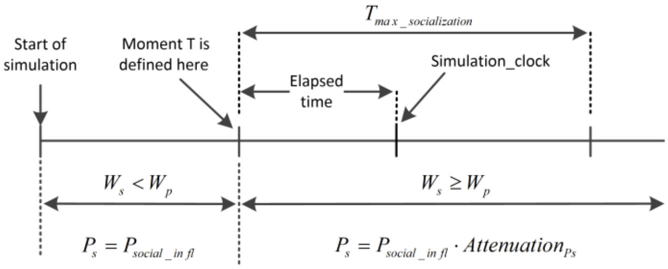


Fig. 6. Example in which the moment *T* is defined

The elapsed time (14) is the difference between the current simulation time (*Simulation_clock*) and the start time *T*. We normalize $T_{elapsed}$ according to a maximum socialization time ($T_{max\_socialization}$) to obtain a period expressed in percentages and which is specific to each node. Thus, $Attenuation_{Ps}$ tends to 0 (maximum attenuation for P_s) as $T_{elapsed}$ tends to 100%. In our model, $T_{max\_socialization}$ is the time required for a node to become completely bored with the activity of its neighbors. Each node has its own value for $T_{max\_socialization}$ and is generated randomly at the beginning of the simulation with a period between 1 and 7 days. In this way, each node has a different period length in which it gets bored with the activity of its neighbors. The genetic algorithm controls $Attenuation_{Ps}$ by learning the parameter $F_{social\_tolerance}$ defined on the range [1, 15].

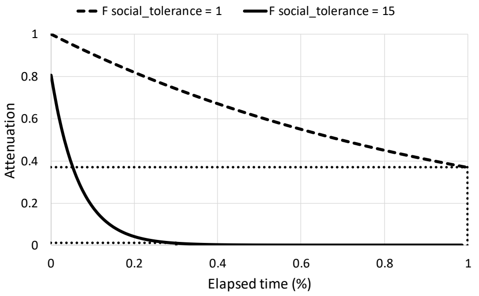


Fig.7. Example of attenuation for a) $F_{social\_tolerance}$ = 1 b) $F_{social\_tolerance}$ = 15

In Fig. 7 we show two examples for the evolution of $Attenuation_{Ps}$. One can see a severe attenuation when $F_{social\_tolerance}$ =15, i.e. probability P_s is completely suppressed when the elapsed time is 30% of the $T_{max\_socialization}$. In the case of $F_{social\_tolerance}$=1, the attenuation is very low and $Attenuation_{P_S} \cong 0.4$ at 100% elapsed time.

3.6. Determining the Weights

The final sending probability (P_f) for a special node (3) depends on the P_p and P_s probabilities. P_p and P_s are weighted by W_p and W_s, which are complementary weights (i.e.$W_p = 1 - W_s$). So, we will only discuss about W_s, which is defined as follows:

$$W_s = 1 - e^{-Pct_{neighbors} \cdot F_{act\_social\_model}} \quad \textbf{(15)}$$

where $Pct_{neighbors}$ is the percentage of active neighbors of the node, and $F_{act_social_model}$is a control parameter. We say that a node is active if it has transmitted at least one information, so the weights $W_p$and W_s are not influenced by the number of messages from the neighbors of a node. Instead, W_p and W_s are influenced by the state of the neighbors (i.e. active or inactive node).

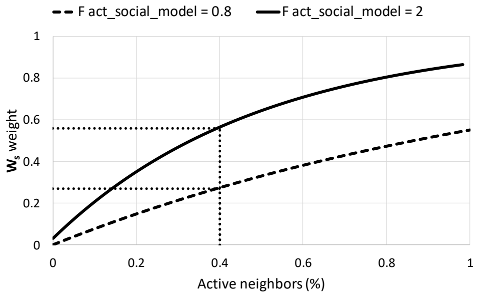


Fig. 8. Evolution of W_s for $F_{act\_social\_model}$= 0.8 and $F_{act\_social\_model}$= 2

We control the evolution of the weight W_s by using the parameter $F_{act\_social\_model}$. In this way we actually control the degree of an individual to be influenced by others, or how quickly the individual adopts a gregarious behavior. In Fig. 8 we show the evolution of W_s for two values of the parameter $F_{act\_social\_model}$: 0.8 and 2. These two values are in fact the limits in which $F_{act_social_model}$is learned by the genetic algorithm.

If a node has, for example, 40% active neighbors, one can see that $W_s \cong 25\%$ when $F_{act\_social\_model}$ has the value 0.8 and $W_s \cong 55\%$ when $F_{act\_social\_model}$ has the value 2. Therefore, when the value of $F_{act\_social\_model}$ is higher, the node is more encouraged to follow its neighbors during information diffusion. Also, the login rate of a node is influenced by W_s during the simulation:

$$\lambda_{login} = \begin{cases} \lambda_{normal}, W_s < W_p \\ \lambda_{social}, W_s \geq W_p \end{cases} \quad \textbf{(16)}$$

We choose a lower login rate (λ_{normal}) for a node when it is not strongly influenced by the activity of its neighbors and a higher login rate (λ_{social}) when the node has a gregarious behavior:

$$\lambda_{final\_login} = \begin{cases} \lambda_{login}, 11AM \leq simulation\_clock < 11PM \\ \dfrac{\lambda_{login}}{2}, 11PM \leq simulation\_clock < 11AM \end{cases} \quad \textbf{(17)}$$

Moreover, we consider a change in the login rate of the nodes depending on the time of day. Thus, we define two time periods (17): the time interval 11AM – 11PM is considered the daytime period (λ_{login} remains unchanged for users), and the time interval

11PM – 11AM is considered the nighttime (λ_{login} is halved for each user). In this way, we model the fact that users are less active at night.

3.7. Learning the Model Parameters

The proposed information diffusion protocol contains a large number of parameters that influence the evolution of the diffusion. It is difficult to adjust so many parameters in order to obtain an evolution of the diffusion as close as possible to the real one. The automatic learning of the parameters is the solution that helps us in this problem and we have chosen to use a genetic algorithm. In Table 1 we show the parameters of our diffusion model that are learned by the genetic algorithm, and in Table 2 we show the parameters that are not learned.

Table 1. The parameters of the diffusion model that are learned by the genetic algorithm

Parameter	Definition range	Comment
λ_{normal}	[120, 240]	The login rate used by the nodes with a small number of active neighbors (e.g. the minimum value means 1 login every 120 minutes)
λ_{social}	[30, 60]	The login rate used by the nodes with a high number of active neighbors
$T_{max\_informant}$	[1, 90]	The maximum time period (in minutes) in which an informant node spreads the information
$F_{attenuation}$	[0.01, 1]	Attenuation factor for probability P_p
W_{cred}	[0.01, 0.2]	The control weight over increasing the information credibility over time
$F_{act\_social\_model}$	[0.8, 2]	Control factor to adjust the evolution of the W_s weight
$F_{social\_tolerance}$	[1, 15]	Control factor to adjust the attenuation of the P_s probability
W_{act}	[0.2, 1]	The control weight of the activation level from the $P_p$probability

Table 2. The parameters of the diffusion model that are not learned by the genetic algorithm

Parameter	Definition range	Comment
F_{social_infl}	[5,10]	Control factor to adjust the evolution of P_{inf_social}
Source nodes	5%	Percentage of source nodes
Informant nodes	30%	Percentage of informant nodes
P_b	0.8	The basic probability of the nodes
$T_{max\_socialization}$	[1, 7]	The maximum time period (in days) in which a node becomes bored with the activity of its neighbors
$Cred_{Info}$	0.3	Initial credibility of the information (used by the source nodes)
$T_{start\_informant}$	Depending on the real diffusion	The time when the informant type nodes activate and suddenly encourage the spread of information

3.8. The Genetic Algorithm

Background. Genetic algorithms are based on the principles of natural selection [23], which states that the survival of an organism consists in the survival of the most adapted species. For a species to survive in time, the following stages are required: selection, reproduction and mutation. The *selection* process consists in the fact that certain organisms of the species better tolerate the environment in which they live and, consequently, have a greater chance of survival. Thus, these organisms are more adapted (fitted) to the environment, due to specific genes (the set of all genes is called a chromosome). More adapted organisms have a higher chance of reproducing. In the *crossover* process, parents transmit certain genes to the offspring. The new generation that results from the crossover is a new epoch and this process represents a way to simulate the evolution of the species over time. However, selection and reproduction are not sufficient to ensure long-term improvement of an organism's adaptation. Also, there is the possibility that an organism will suffer changes of the genes that did not result from the crossing of the parents and these changes may lead to a better adaptation of the new individual. The process in which these new genes suffer unexpected changes is called *mutation*.

By analogy with the search for solutions to a problem, the genetic algorithm is based on the concept of biological evolution to simulate a finite number of epochs in order to find the most fitted individuals that ultimately represent the desired solutions. [24]. The set of all individuals of an epoch is called population. In our case, the genetic algorithm learns certain parameters (Table 1) to obtain an evolution of the information diffusion as close as possible to the real one. The real evolution of diffusion and the result provided by our diffusion model from a particular individual are both represented as a one-dimensional array. We choose that the stop condition of the genetic algorithm should be given by the iteration of a certain number of epochs. The following operations are performed at each epoch: selection, crossover and mutation.

Selection. There are different methods of selecting parents to create the new generation, and we choose the *tournament selection*. The basic idea for this type of selection is as follows:

- We sample *k* random individuals from the current population and choose the one with the best fitness as a parent (*k*=2 in our case);
- The procedure is repeated to select more parents.

We also use the *elitism* operation in which we get the individual with the best fitness from the previous population and add it to the new population. In this way we will never lose the best solution found throughout the epochs.

Crossover. After the selection process is completed, the selected parents (i.e. mating pool) are used for *crossover*. We choose pairs of two parents to create children with new genes for the new population. In Fig. 9 we show how the parents are paired. If the number of created children is not sufficient to create the new population at the expected size, then the pairs of parents are crossed again to obtain other children.

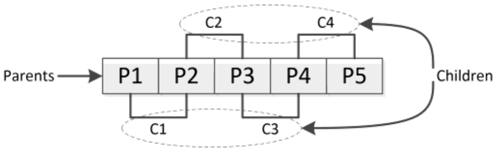


Fig. 9. The way the parents are paired for mating

In our case, the genes from each individual have a real numerical representation, and we use the *arithmetic crossover*:

$$z_i = \alpha \cdot x_i + (1 - \alpha) \cdot y_i \tag{18}$$

where x_i and y_i are the *i*-th gene of the two parents, z_i is the *i*-th gene of the child, and α is a uniformly distributed random number in the interval [0, 1].

Mutation. Each gene of a new child has a small chance of undergoing unexpected changes. This unexpected event represents the *mutation* operation. We choose the *random resetting mutation*, in which the value of a gene is replaced by a random value from its given range (Table 1).

General Pseudocode. Below one can see the general pseudocode of a genetic algorithm (Algorithm 1), as used for the experiments in the present paper.

Algorithm 1. The proposed algorithm

```
For epoch = 1 to MAX_EPOCH do
//Compute the fitness of all individuals in the
  //population. The dissemination of information is
  //simulated for each individual and is compared to the
  //real one
Compute_fitness(Population);
//Parent selection for the mating pool
Mating_pool = Selection(Population);
//Elitism: the best individual is always chosen for the
  //next population
Best_individual = Best(Population);
//Empty the population and keep only the best individual
Clear_population(Population);
Population.Add(Best_individual);
//Create (MAX_INDIVIDUALS - 1) children
Children = Crossover(Mating_pool, MAX_INDIVIDUALS - 1);
//Modify newly obtained children
Final_children = Mutation(Children);
//Add children to the population
Population.Add(Final_children);
End
//The best solution obtained
Get_parameters(Best_individual);
```

The Fitness Function. The fitness of an individual, which is used in the selection process, is computed using a fitness function that is problem specific. In our case, the fitness function (19) computes the difference between the real and the simulated diffusion using the Euclidean distance, where O_i is a sample from the real diffusion, S_i is a sample from the simulated diffusion, and N is the size of the arrays.

$$f = \sqrt{\sum_{i=1}^{N} (O_i - S_i)^2} \quad \textbf{(19)}$$

In our case, a better individual will have a lower value for *f*: the smaller the *f*, the closer the individual is to the optimal solution.

In Fig. 10 we show an example in which the parameters of each individual are used in the information diffusion model to provide an evolution of the diffusion. Then, the obtained evolution can be compared with the real one using the fitness function to obtain the fitness of an individual.

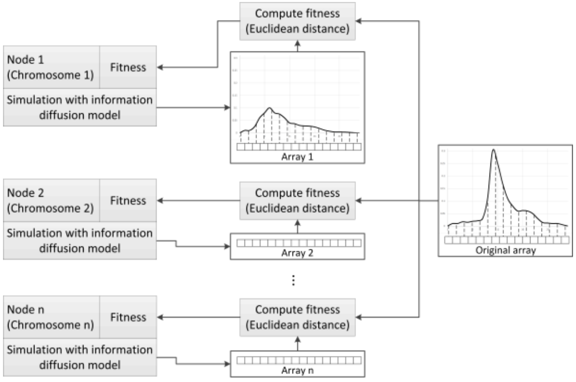

Compute fitness (Euclidean distance)
Node 1 (Chromosome 1)
Fitness
Simulation with information diffusion model
Array 1
Node 2 (Chromosome 2)
Fitness
Simulation with information diffusion model
Compute fitness (Euclidean distance)
Array 2
Original array
Node n (Chromosome n)
Fitness
Simulation with information diffusion model
Compute fitness (Euclidean distance)
Array n

Fig. 10. Example of fitness computation for each individual

In our case, the genetic algorithm has the following configuration: epochs – 100, population size – 100 individuals, and mutation rate – 10%. This algorithm aims to learn 8 parameters from our diffusion model, shown in Table 1.

4. Experimental Results

4.1. Description of the Two Real-World Datasets

In order to evaluate the efficiency of our information diffusion model, we use two real datasets that contain the activity of users over time. We use the genetic algorithm to learn the parameters of our information diffusion model for each real diffusion of a dataset. The first dataset is a collection of information on the activity of Twitter users during the announcement for the discovery of the Higgs Boson [1], which also contains the network structure. The activity of a user on Twitter represents his/her action to share a piece of information he/she saw on a neighbor's page (retweet). The shared information is visible to all of the user's neighbors. The dataset also contains the timestamps when users share their information. Therefore, we can extract the activity of users over time and correlate it with the evolution of information diffusion, i.e. the target solution used by the genetic algorithm. The second dataset, *memetracker9* [2], is a large collection of data in which the exchange of information between users is described by text messages or links to other web pages. This dataset contains the diffusions of several topics that are collected over several months. The diffusion of each topic discussed by users is easily identified in paper [25], and we choose two of them: "lipstick on a pig" and "fundamentals of our economy are strong". The diffusion evolutions from each dataset are provided to the genetic algorithm to learn the parameters of our diffusion model and to obtain evolutions as close as possible to the real ones.

4.2. Results for the *Higgs* Dataset

For the Higgs dataset, we apply the genetic algorithm on a synthetic network of 1000 nodes with scale-free topology because the original network contains 456,626 nodes and 14,855,842 connections, which results in a very high simulation time and cannot be used in the genetic algorithm. In [6], the authors state that the evolution of information diffusion in scale-free networks is similar even if the networks have different sizes. Thus, we can apply the genetic algorithm without having to simulate a very large network.

The parameters learned by the genetic algorithm are initialized with random values in their specific range (Table 1), and it is expected to obtain weaker solutions in the first epochs. In Fig. 11 we show an evolution example of the best solutions from each epoch on the diffusion of the Higgs dataset and one can see that the solutions are drastically improving in the first 20 epochs. We consider that 100 epochs is an acceptable stop condition for the genetic algorithm because no significant improvements can be observed for a higher number of epochs.

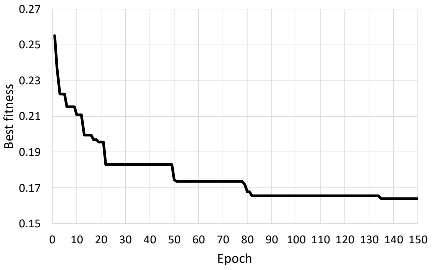
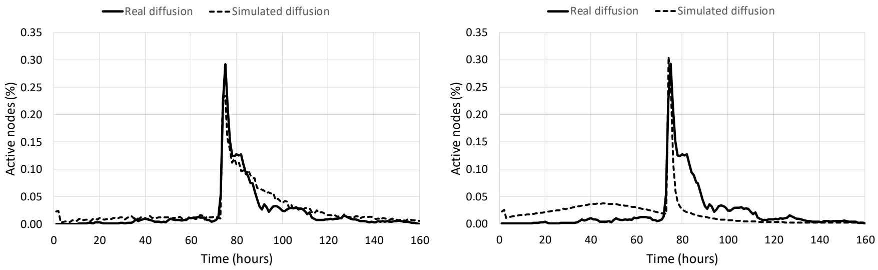


Fig. 11. The best fitness obtained at each epoch

In the case of the Higgs dataset, we show in Fig. 12.a the evolution of the information diffusion obtained by our diffusion model on the 1000-node synthetic network. The continuous line represents the real evolution of the diffusion, while the dotted line represents the diffusion obtained by our model. The simulated diffusion is obtained by counting the active nodes at intervals of one hour. Then, we use the learned values of the diffusion model parameters to run a simulation on the real network provided by this dataset.



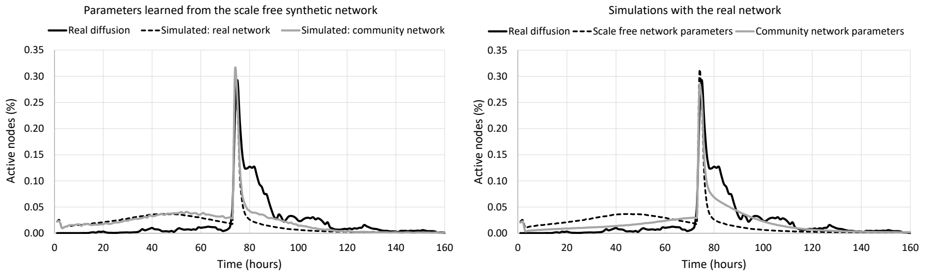
a) b)

Fig. 12. Simulated diffusion on: a) synthetic 1000-node network, b) real network

We show the result in Fig. 12.b and one can see that the obtained diffusion is close to the real one, except that the simulated diffusion has a greater attenuation. We could not make any further adjustments of the parameter values on this large network because it takes a day to run 2-3 simulation rounds.

We mention that for the Higgs dataset we do not use the modeling for the daytime and nighttime periods because users had different geographical positions on several distant continents, according to [7], and our algorithm does not take into account different times of the day between users.

Using a Community Network. Given that real networks contain many communities, we want to observe the impact of using a synthetic network with communities in our information diffusion model. The motivation of this study is due to the fact that the simulated diffusion has a greater attenuation (Fig. 12b). Regarding the high peak obtained with our model on the real network (Fig. 12b), in our investigations we observe a very similar behavior when we use a synthetic network with communities. This community network has 1000 nodes and was generated using the Gaussian random partition graph [26]. In Fig. 13a, one can see the evolution of simulated diffusion both on the real network and on the network with communities. These evolutions are provided by our model using the parameters learned by the genetic algorithm on the synthetic scale free network. Due to the similarity between these two evolutions, we use the genetic algorithm to learn the parameters of the diffusion model on this network with communities. We want to simulate the diffusion of information on the real network with two different sets of parameters: the first set is learned by the genetic algorithm on the synthetic scale free network and the second set is learned on the synthetic network with communities. We can observe in Fig. 13b that the simulation of the real network with the second set of parameters now has an evolution closer to the real diffusion. Based on these observations we can say that a limitation of our model is the choice of a synthetic network with a certain topology.



a) b)

Fig. 13. Comparative simulations: a) real network and community network, b) real network with different model parameters

4.3. Results for the *memetracker9* Dataset

The second dataset (*memetracker9*) is very large and contains conversations between users for several months, between 2008 and 2009. This dataset does not provide the real network, so we show experimental results using only the 1000-node synthetic network. In paper [25], the authors provide a picture that contains various diffusions from this dataset and highlight the main phrases (i.e. the topic of discussions) for each diffusion. In our work, we use specific keywords to extract the diffusion of the following two topics from September 2008: “lipstick on a pig” and “fundamentals of our economy are strong”. The keywords used to identify the phrases of the first topic are “lipstick” and “pig”, while for the second topic we use the keywords “strong”, “economy” and “fundamentals”. Our diffusion model is capable of providing a single evolution of the diffusion for a single topic discussed by users. In Fig. 14 we stack the diffusion of the two topics on the same plot to easily observe the different time periods in which these diffusions are active and also it is easier to compare the evolution of the diffusions with those of [25]. The continuous lines represent the real diffusions and the dotted lines are the simulated diffusions. We also distinguish the two topics by line width: high width for the “lipstick on a pig” topic, and low width for the second topic. The simulated diffusion is obtained by counting the active nodes at six-hour intervals.

The evolution of diffusion from each topic is obtained by a separate simulation using the genetic algorithm, therefore the parameters learned for the diffusion model have different values for the two topics. Regarding the parameters that are not learned with the genetic algorithm, the main difference between the simulation of the two topics is that we change $T_{start\_informant}$, which is the moment when the information is suddenly spread (i.e. users discuss a lot about the given topic). Although the diffusion model is applied on the synthetic network, one can see in Fig. 14 that the obtained diffusion is very close to the real one.

In our model we propose that the user login rate depends on the time of day (i.e., day or night). In [27] a detailed analysis of user activity is presented on two datasets and a certain sinusoidal periodicity is observed in this activity. The authors notice that these activities are periodic at 24-hour intervals on both datasets. They also illustrate the activity of users during a week, and an interesting aspect is that they identify a consistent drop in activity during weekends on both datasets. Our model does not take into account an attenuation of user activity during weekends, therefore it cannot reproduce these low amplitudes of periodicity as the real data (e.g. Fig. 14, time frame 230-270 or 460-480).

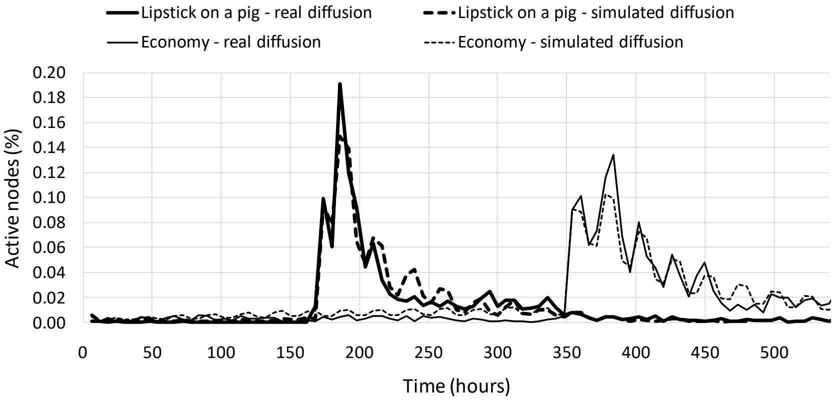
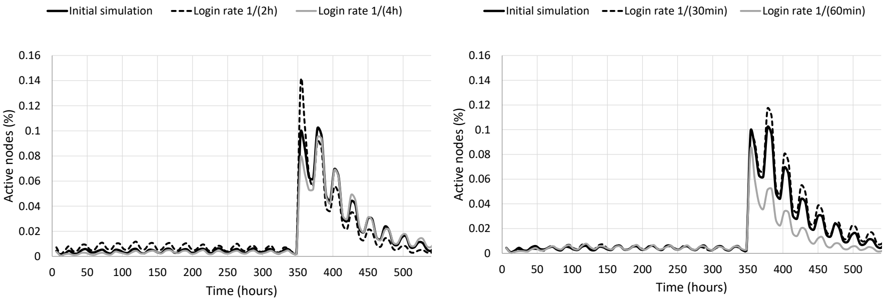


Fig. 14. Information diffusion simulation for two topics in the *memetracker9* dataset

4.4. Influence of Parameters

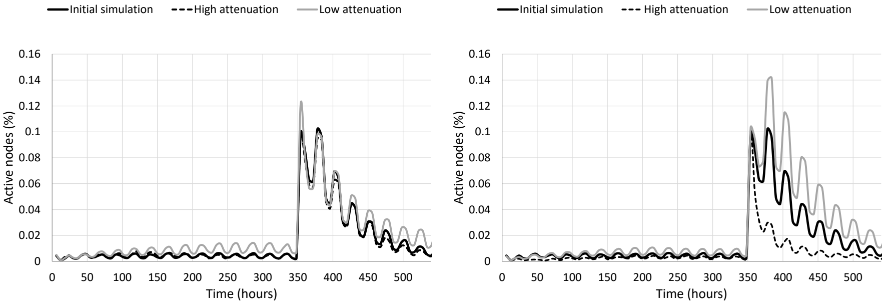
In this section we make an analysis for some of the individual parameters of our diffusion model. The impact of a parameter can be analyzed separately by simulating the diffusion if we keep all the parameters constant and adjust only the parameter of interest. If we refer to the parameters learned by the genetic algorithm, we can show, for example, the impact of the login rate (Fig.15) and that of the boredom of the nodes (Fig. 16). Depending on the two models we have:

- psychological modeling: normal login rate (λ_{normal}), boredom over the amount of information ($F_{attenuation}$)
- sociological modeling: social login rate (λ_{social}), boredom over the activity of neighbors ($F_{social\_tolerance}$)



a) b)

Fig. 15. Login rate modeling: a) normal login rate; b) social login rate



a) b)

Fig. 16. Boredom modeling: a) over the amount of information; b) over the activity of neighbors

We can see that the parameters of sociological modeling have a great impact on diffusion. The continuous black line represents the simulated diffusion using the values of the parameters obtained with the genetic algorithm. The other two evolutions are obtained by varying a single parameter (the one of interest) to observe its impact on the diffusion.

Unlike other works (e.g. [5, 6]) in which the parameters are manually adjusted through repeated simulations, we present a model in which its parameters are automatically learned, and the obtained diffusions are very promising compared to the real ones.

5. Conclusions

Developing models capable of imitating the information diffusion on a social network is a challenging task at the moment. In this paper we propose such a model that imitates the diffusion of information as well as possible. The model is based on stochastic node-level decisions. Each node has its own set of rules by which its actions are defined. The decision of a node, whether or not to transmit information, is modeled both from a psychological point of view and from a sociological point of view. In modeling from the psychological point of view, we propose that the decision of a node should be influenced by its preferences on the content of the information. On the other hand, when modeling from the sociological point of view, we propose that the decision of the node should be influenced by the activity of its “friends”, i.e. we model the gregarious behavior of the node. Also, most of the parameters of our diffusion model are learned by means of a genetic algorithm to eliminate the effort of adjusting their values. Then, we can use the learned values in the proposed diffusion model to obtain an evolution of information diffusion as close as possible to the real one. We use two datasets that contain real diffusions, and the results show that our model reproduces them very well.

However, one must take into account the fact that there is no unique model for all situations. One goal of our work was to show that the proposed model with psychosocial factors is capable of approximating real data. But every case will likely need different values of the parameters, which can be found through automatic search using genetic algorithms or other optimization methods, e.g. based on gradients.

As a future direction of investigation, one can investigate the inclusion of additional parameters in the model (such as those accounting for weekend activity or the different geographical distribution of users on different continents), in order to further increase the prediction accuracy. One must also consider the trade-off between increase flexibility and the growth of the search space, while applying the automatic determination of parameter values.

References

1. Stanford Network Analysis Platform (SNAP): Higgs Twitter Dataset. [Online]. Available: https://snap.stanford.edu/data/higgs-twitter.html
2. Stanford Network Analysis Platform (SNAP): 96 million memes from Memetracker. [Online]. Available: https://snap.stanford.edu/data/memetracker9.html
3. de C Gatti, M. A., Appel, A. P., dos Santos, C. N., Pinhanez, C. S., Cavalin, P. R., Neto, S. B.: A Simulation-based Approach to Analyze the Information Diffusion in Microblogging Online Social Network. In Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World. IEEE Press, Washington, DC, USA, 1685-1696. (2013)
4. Serrano, E., Iglesias, C. A., Garijo, M.: A Novel Agent-Based Rumor Spreading Model in Twitter. In Proceedings of the 24th International Conference on World Wide Web. Association for Computing Machinery, Florence, Italy, 811-814. (2015)
5. Chen, J., Song, Q., Zhou, Z.: Agent-Based Simulation of Rumor Propagation on Social Network Based on Active Immune Mechanism. Journal of Systems Science and Information, Vol. 5, No. 6, 571-584. (2017)
6. Mazzoli, M., Re, T., Bertilone, R., Maggiora, M., Pellegrino, J.: Agent Based Rumor Sspreading in a Scale-free Network. (2018). [Online]. Available: https://arxiv.org/abs/1805.05999
7. De Domenico, M., Lima, A., Mougel, P., Musolesi, M.: The Anatomy of a Scientific Rumor. Scientific Reports, Vol. 3, No. 2980. (2013)
8. Floria, S.-A., Leon, F., Logofătu, D.: A Credibility-based Analysis of Information Diffusion in Social Networks. In Proceedings of the 27th International Conference on Artificial Neural Networks (ICANN). Springer International Publishing, Rhodes, Greece, 828–838. (2018)

9. Floria, S.-A., Leon, F., Logofătu, D.: A Model of Information Diffusion in Dynamic Social Networks Based on Evidence Theory. Journal of Intelligent & Fuzzy Systems, Vol. 37, No. 6, 7369-7381. (2019)
10. Huo, L., Ding, F., Liu, C., Cheng, Y.: Dynamical Analysis of Rumor Spreading Model Considering Node Activity in Complex Networks. Complexity, Vol. 2018, 1-10. (2018)
11. Zhao, L., Wang, X., Wang, J., Qiu, X., Xie, W.: Rumor-Propagation Model with Consideration of Refutation Mechanism in Homogeneous Social Networks. Discrete Dynamics in Nature and Society, Vol. 2014. (2014)
12. Liu, L., Qu, B., Chen, B., Hanjalic, A., Wang, H.: Modeling of Information Diffusion on Social Networks with Applications to WeChat. Physica A: Statistical Mechanics and its Applications, Vol. 496, 318-329. (2018)
13. Yan, Q., Wu, L., Liu, C., Li, X.: Information Propagation in Online Social Network Based on Human Dynamics. Abstract and Applied Analysis, Vol. 2013, 1-6. (2013)
14. Sun, Q., Li, Y., Hu, H., Cheng, S.: A Model for Competing Information Diffusion in Social Networks. IEEE Access, Vol. 7, 67916-67922. (2019)
15. Li, D., Zhang, S., Sun, X., Zhou, H., Li, S., Li, X.: Modeling Information Diffusion over Social Networks for Temporal Dynamic Prediction. IEEE Transactions on Knowledge and Data Engineering, Vol. 29, No. 9, 1985-1997. (2017)
16. Guille, A., Hacid, H., Favre, C., Zighed, D. A.: Information Diffusion in Online Social Networks: A Survey. ACM SIGMOD Record, Vol. 42, No. 2, 17-28. (2013)
17. Li, M., Wang, X., Gao, K., Zhang, S.: A Survey on Information Diffusion in Online Social Networks: Models and Methods. Information, Vol. 8, No. 4, 118. (2017)
18. Dey, K., Kaushik, S., Subramaniam, L. V.: Literature Survey on Interplay of Topics, Information Diffusion and Connections on Social Networks, (2017). [Online]. Available: https://arxiv.org/abs/1706.00921
19. Hawkins, D. I., Mothersbaugh, D. L.: Consumer Behavior: Building Marketing Strategy (11th ed.). Boston, MA: McGraw-Hill. (2010)
20. Shariff, S. M., Zhang, X., Sanderson, M.: User Perception of Information Credibility of News on Twitter. In Proceedings of the 36th European Conference on IR Research, ECIR 2014. Springer Cham, Amsterdam, The Netherlands, 513-518. (2014)
21. Castillo C., Mendoza, M., Poblete, B.: Information credibility on twitter. In WWW'11: Proceedings of the 20th international conference on World wide web. Association for Computing Machinery, New York, NY, United States, 675-684. (2011)
22. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, chapter 16. (2010)
23. Darwin, C.: Origin of the Species. (1859).[Online]. Available: http://darwin-online.org.uk/converted/pdf/1861_OriginNY_F382.pdf
24. Baeck, T., Fogel, D. B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Institute of Physics Publishing Publishing and Oxford University Press. (1997)
25. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the Dynamics of the News Cycle. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Association for Computing Machinery, Paris, France, 497-506. (2009)
26. NetworkX developers: NetworkX Python package. [Online]. Available: https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.generators.community.gaussian_random_partition_graph.html#networkx.generators.community.gaussian_random_partition_graph (current September 2020)
27. Flamino, J., Dai, W., Szymanski, B. K.: Modeling Human Temporal Dynamics in Agent-Based Simulations. In Proceedings of the 2019 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Association for Computing Machinery, New York, NY, United States, 99-102. (2019)

Sabina-Adriana Floria received the B.Sc. degree in Computer Science (2014), the M.Sc. degree in Embedded Computers (2016) and the Ph.D. degree in Computer Science and Information Technology (2020) from the “Gheorghe Asachi” Technical University of Iași, Romania, Faculty of Automatic Control and Computer Engineering. She became a Teaching Assistant at the Department of Computer Science and Engineering, “Gheorghe Asachi” Technical University of Iași in 2017, where she teaches the following subjects: Discrete Mathematics, Computer Programming, Computational Logic, Logical Design, Modelling and Simulation, Computer System Testing. Her research interests include the following aspects: information diffusion in social networks and performance evaluation in complex networks.

Florin Leon received a Ph.D. degree in computer science from the “Gheorghe Asachi” University of Iași, Romania in 2005, followed by a postdoctoral fellowship completed in 2007. In 2015, he defended his habilitation thesis. He has been a faculty member at the Department of Computers and Information Technology of the same university since 2005. In 2015, he became a Full Professor at the same department. He authored and co-authored more than 160journal articles, book chapters and conference papers, and 14 books. He was a member in the guest editorial boards for three journal special issues, and he participated in 29 national and international research projects, three of which as principal investigator. His research interests include: artificial intelligence, machine learning, multiagent systems and software design. Prof. Leon was a member of the organizing committees or program committees chair of five conferences. He is currently a member of IEEE Systems, Man and Cybernetics Society: Computational Collective Intelligence Technical Community and the Romanian Association for Artificial Intelligence.

Received: April 15, 2020; Accepted: December 10, 2020