Simulation of misinformation spreading processes in social networks: an application with NetLogo

Emilio Sulis
Computer Science Department
University of Torino
Turin, Italy
emilio.sulis@unito.it

Marcella Tambuscio
Austrian Center for Digital Humanities
Austrian Academy of Sciences
Vienna, Austria
marcella.tambuscio@oeaw.ac.at

***Abstract*—We introduce an agent-based framework (developed in NetLogo, one of most relevant simulation platforms) to simulate the diffusion of a piece of misinformation, according to a known compartmental model in which the fake news and its debunking compete in a social network. The tool allows to set different values for the spreading rate of the news, the hoax credibility, the probability of fact-checking and the forgetting rate of the agents. Moreover, it is possible to run the process over any given network. Since NetLogo is free and open source, our tool could be easily used and/or personalised by other researchers to explore different scenarios of fake news spreading.**

***Index Terms*—Information Diffusion, Fake News Spreading, Fact-Checking, Agent-based modeling, NetLogo**

I. Introduction

Misinformation spreading is currently one of the most discussed topics in our society, specially because, even if it is not a new problem, digital technology and new media can foster its diffusion [1], [2]. Research about misinformation can be categorized on three main lines: automatic detection of fake news and social bots [3]–[5], psychological aspects and effects [6]–[8] and finally analysis [9], [10] and modeling [11]–[15] of their diffusion to understand its spreading dynamics.

Here we focus on the third research line and we provide an application to simulate an hoax spreading process with NetLogo [16]. In particular we target the model presented in [15] that follows the epidemic tradition in representing rumor diffusion. The novelty of this model is that misinformation and fact-checking are competing actors in a population of agents that can decide whether believe or not to the hoax depending on the credibility of the hoax and on belief of their neighbors; they can also verify or forget about the news with certain probabilities that are parameters of the model. From a pure theoretical point of view, through simulations and mean-field analysis, the authors provided a threshold for the verifying probability that ensures the eradication of the hoax.

We investigate here the adoption of the same diffusion process in an Agent-Based Modeling (ABM) perspective [17]. ABM focuses on emergent phenomena [18] in complex adaptive systems [19]. Since the behavior of the chosen model has been extensively studied on scale-free and random networks, the goal of this work is providing a tool developed in NetLogo that can be used on a large scale to explore also slightly different and personalized versions of the model with the possibility to run the process on any topology.

We strongly believe that this implementation can provide a useful tool to realize *what-if* analysis [20] and represent various scenarios through agent-based simulations.

II. Related works

A. Modeling information diffusion

Representing information and rumor spreading through epidemic metaphors has a long tradition [21], [22]: a (fake) piece of information, indeed, can be seen as a virus that may potentially infect people. In particular, many works took inspiration from *compartmental models* in which there is a population of agents connected among them, and each agent i at each time t can be in a state (compartment) $s_i(t)$. The transitions are usually defined by simple equations ruled by some probability rates. The most famous models of this category are the SIR (Susceptible-Infected-Recovered) and the SIS (Susceptible-Infected-Susceptible). Many SIR-based models have been proposed to model rumor spreading [11], adding forgetting and remembering mechanisms [23], the presence of skeptic agents [14], and competition among rumors [13]. The SBFC model that we consider here [15] has three states (Susceptible-Believer-FactChecker) and agents have the possibility to be infected either by the hoax or by its debunking, to forget their belief or verify the news (see details in section III-A).

B. Agent-based modeling and diffusion processes

In this paper we explore an initial implementation step of a misinformation spreading model with multi-agent systems, intended as a specific class of computational models to simulate actions and interactions of autonomous agents, with the main goal of assessing their effects on the system as a whole. In particular, ABMs can "provide a more fine-grained model of the process, with many parameters that can impact the dynamics. We call such models, which explicitly model the individual agents, agent-based simulation models" [24]. In particular, we adopt NetLogo[1] which is one of the most used

[1] https://ccl.northwestern.edu/netlogo/

ABM platforms, here in use both as a modeling-simulation tool and as a general purpose tool.

The salient features of NetLogo include the capability of managing collection of instances of a class of agents, traditionally named *turtles*[2]; the openness, being a Free and Open Source Software (FOSS) written in Java and Scala; the availability of libraries or extensions to the basic code, to connect R [25] or Python [26]; in addition, an easy to use graphical representation of the agents' world. The diffusion processes in social media have been already simulated by following an ABM perspective [27]. The role of social networks to predict the diffusion process has been explored in the context of the introduction of new products in a market [28], as well as on different real-world Twitter networks structures [29].

III. METHODS

A. *Formal Description of SBFC model*

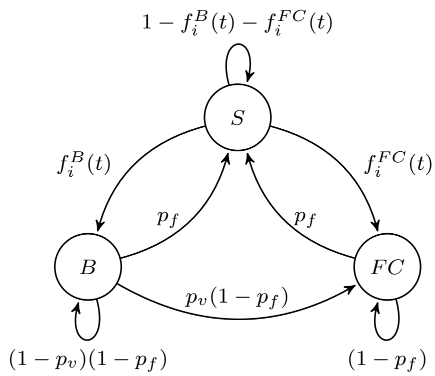

Fig. 1. A diagram of the model SBFC.

The SBFC model [15] describes the diffusion of misinformation as a competition among a false news and its debunking in a population of N agents. At each time t each agent i is associated with a state $s_i(t)$ that can take one of the three values:

- Susceptible (S), an agent who ignores the news;
- Believer (B), an agent who believes to the hoax;
- FactChecker (FC), an agent who has verified the news or directly knows that it is an hoax.

Formally we use a state indicator function for each time t:

$$s_i^{\{S,B,FC\}}(t) = \delta(s_i(t), \{S, B, FC\})$$

and a triplet $p_i(t) = (p_i^S(t), p_i^B(t), p_i^{FC}(t))$ that collects the probabilities that the node i is in the three states at time t. Before giving the formulas for these probabilities, whose random realization rules the dynamics of the system, we list the parameters and the phenomena represented by the model. There are four fundamental parameters: the credibility of the hoax $\alpha \in [0, 1)$ that gives some advantage over its debunking, its spreading rate $\beta \in [0, 1]$, a verifying probability $p_v \in [0, 1]$ and a forgetting rate $p_f \in [0, 1]$. The transitions through the states can be summarized in three phenomena. First of all there are two *spreading* transitions ($S \to B, S \to FC$) that determine when a susceptible agent i decides to believe or not to the rumor. These transitions are defined respectively by two spreading functions f_i^B and f_i^{FC} that depend on the belief of the neighbors of agent i, the spreading rate β and the credibility α:

$$f_i^B(t) = \beta \frac{n_i^B(t)(1+\alpha)}{n_i^B(t)(1+\alpha) + n_i^{FC}(t)(1-\alpha)} \tag{1}$$

$$f_i^{FC}(t) = \beta \frac{n_i^{FC}(t)(1-\alpha)}{n_i^B(t)(1+\alpha) + n_i^{FC}(t)(1-\alpha)} \tag{2}$$

where $n_i^{B|FC}(t)$ is the number of neighbors of i that are in the Believer/FactChecker state at time t. Second, there is a *verifying* transition $B \to FC$ that let a Believer become directly a FactChecker (because he realizes that the news is a hoax, for instance) and it is simply ruled by the verifying probability p_v. Third, agents in Believer and FactChecker states can *forget* their belief towards the news and return to the Susceptible state with a probability p_f. These transitions happen following the triplet $p_i(t+1) = (p_i^S(t+1), p_i^B(t+1), p_i^{FC}(t+1))$ that describe the probabilities that the node i, that is in a state $s_i(t)$ at time t switches to a state $s_i(t+1) \in \{S, B, FC\}$ at time $t+1$. Formally:

$$\begin{aligned} p_i^S(t+1) =& [1 - f_i^B(t) - f_i^{FC}(t)] s_i^S(t) + \\ & p_f [s_i^B(t) + s_i^{FC}(t)] \\ p_i^B(t+1) =& f_i^B(t) s_i^s(t) + (1-p_f)(1-p_v) s_i^B(t) \\ p_i^{FC}(t+1) =& f_i^{FC}(t) s_i^S(t) + \\ & p_v s_i^B(t) + (1-p_f) s_i^{FC}(t) \end{aligned} \tag{3}$$

In [15] the dynamics of the process has been explored extensively varying the credibility and verify probability that rule the victory of the hoax (high α low p_v) or the debunking (low α high p_v); moreover, it was found analytically a threshold for the verifying probability that assures the false news will be eradicated. In following papers other more complex versions of the model have been studied exploring the role of network segregation [30] and effective fact-checking strategies placing some never-forgetting debunkers in specific nodes of the network [31].

B. *NetLogo implementation*

1) Model structure: The three parts of each program are Interface, Info and Code area. The Interface presents the simulation output area, and allows users in setting parameters by using so called *buttons* for the interactions, e.g., sliders, switch, chooser buttons. Each of them relates to the corresponding procedures in the Code area. The conventional structure of a NetLogo program includes an initial part of variables declarations, followed by the procedures regarding the environment (so called *patches*) as well as the agents (*turtles*). The two main procedures in the code concern both the initialisation of the world (*setup* procedure) and the execution of the simulation (*go* procedure).

[2]The name Turtle is an inheritance from the Logo educational language https://el.media.mit.edu/logo-foundation/

```
to setup
  ca
  setup-var
  setup-turtles
  update-plot
  reset-ticks
end
```

The initial *setup* resets the world to an initial, empty state (*clear-all*), before the initialisation of the main variables in a specific *setup-var* procedure. In our case, just the shape of agents as well as the text to print in the output area. The procedure *setup-turtles* creates the agents and the network, before updating the visualisation of the line plot to count different kinds of turtles. Finally, the reset-ticks button reset the time counter to zero, sets up the initial state of the world in the plots.

The main cycle *go* consists of the continuous increment of time steps until the stop condition (300 time steps) is reached (*ticks*).

```
to go
  tick
  if ticks > 300 [stop]  ; the stop condition
  spreading
  forgetting
  verifying
  (...)
end
```

The three procedures in the main cycle are *spreading*, *forgetting* and *verifying* corresponding to the functions for agents' changing state, as previously detailed in section III-A.

We describe here the code of the procedures *forgetting* and *verifying*, while *spreading* just corresponds to the computation of the above mentioned equations. In particular, the *forgetting* procedure describes how each agent, regardless of belief state, forgets the fake-news with a fixed probability *pforget*. Similarly, each agent can fact-check the hoax with a fixed probability *pverify* in the *verifying* procedure.

```
to forgetting  ; B -> S; F -> S
  ask turtles with [state="B" or state="F"][
    if random-float 1 < pForget [
      set state "S"
    ]
  ]
end

to veryfing  ; B-> F ;
  ask turtles with [state = "B"][
    if random-float 1 < pVerify [
      set state "F"
    ]
  ]
end
```

2) State of agents: Each agent includes a variable *state* to trace the personal condition, identified by a character: Susceptible ("S"), Believer ("B"), or FactChecker ("F").

```
turtles-own [
  state   ; "B", "F", "S"
]
```

3) Display: The diffusion process is represented with three different colors for each state changing over time. Susceptible nodes are gray, Believer nodes are blue and FactChecker nodes are red. The output area in the NetLogo Interface visualizes the network of agents connected by curved gray edges. Several buttons control the simulation setting, while some monitors display the results of some computations, i.e. graph metrics or count of agents (well described also by a line-plot graph).

4) Network extensions: The network of agents is generated according to a specific library included in the very first row of the code with the command *extensions [nw]*. The creation of the network is defined by the *type-of-network* button in the Interface ("Barabási–Albert algorithm" by default).

IV. ABM OUTPUT

A. The Network representation

At the beginning of the simulation, the initial setup of the model creates a network whereas agents are mostly Susceptible (90%), instead of Believer (10%). As described in Figure 2 edges are gray. In addition, some procedures improve the visualisation of the graph by adopting a spring layout, as well as expanding the distances between nodes.

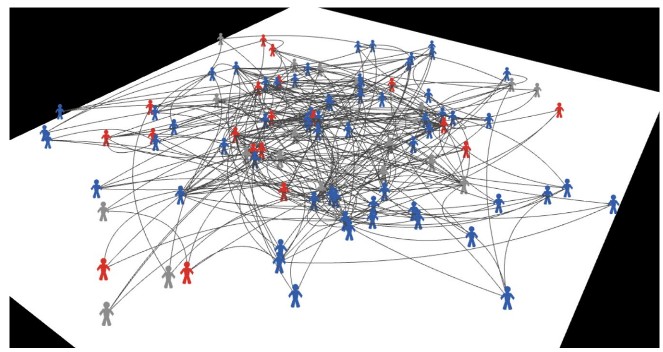

Fig. 2. Output representation of the network in NetLogo. Agents' color indicates the corresponding State. Arcs are gray and a Force-directed Spring layout improves the visualisation.

B. Simulation results

The execution of the simulation can be firstly explored in the graph of the Interface. The count of the three different state types of agents describes the evolution of the diffusion process.

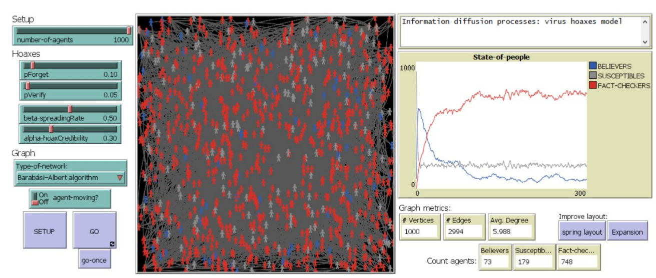

Fig. 3. The Interface tab in NetLogo platform with buttons on the left, the network of agents in the center and simulation results described with a line plot and monitors.

1) Verifying simulation trends: The model behaviour obtained by varying initial parameters with NetLogo are consistent with the ones obtained in the original paper by simulations with R programming language and mean-field analysis [15]. In first experimental settings, Barabasi-Albert (BA) and Erdos-Renyi (ER) networks are considered with the same number of nodes (1,000) and mean degree (6). These simulations maintain fixed the values of both the spreading rate (0.5) and the forgetting probability *pforget* (0.1), with the aim to understand the influence of fact-checking activity by varying only *pverify* and the credibility parameter. Similarly, we tested other configurations which clearly reveal the validity of resulting trends, as represented by the line plot in the Interface (see Figure 3).

2) Verifying raw values: At the same time, we checked values obtained by simulations. For instance, the above mentioned scenario converges to an amount of about 70% Fact-checker, 8% Believer and 22% Susceptible agents. The results obtained here and the ones described in a similar work on hoax diffusion [15], [31] are equivalent. By modifying network configurations as well as formula parameters, these convergence results constitute a verification of the proposed agent-based simulation.

V. Conclusions and future work

This paper introduced a tool to simulate agent-based diffusion processes of fake-news in a social network. We considered a known model based on epidemic spreading process in which misinformation and fact-checking compete on a population of agents that can believe or not to the hoax, verifying or forget the news. The parameters of the model are the spreading rate, the credibility of the hoax, a forgetting probability, a verifying probability, the population size and the initial seeds (number of believers at time $t = 0$). The behaviour has been already extensively studied on random and scale-free networks: here we offer a tool to run the model on any desired topology tuning the values of all the parameters.

The results of our implementation in NetLogo match with the ones obtained by simulations in different platforms and analytical computations. On the other hand, NetLogo offers the possibility of a very intuitive framework to run personalised version of the model, and this opens the way to improve scenario analysis and dynamics of simulations. In order to make the tool more usable, we shared the code of our model [3], a large online collection of NetLogo models.

A first extension of this work would involve performing multiple executions about the phenomena of interest with the same configuration to detect mean values and standard deviation for a consistent number of executions. In particular, we plan to perform sensitivity analysis on the top of the here described model. The exploration of parameter sweep from simulation allows us to sweep the values in predefined increments over a specified range. This effort can be easily achieved by the adoption of *BehaviorSpace*[4], a specific tool integrated in NetLogo. Such tool automatically performs multiple replications (using multiple processor cores) for each of a number of different settings for the model input parameters.

A second kind of future work includes the exploration of adding a specific behavior to agents. This approach paves the way toward solving optimization problems, e.g., by the application of search heuristic technique such as genetic algorithms [32]. In particular we plan to check what happens with the introduction of movements and interactions with other agents (and patches) in a traditional agent-based modeling perspective.

References

[1] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild *et al.*, "The science of fake news," *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.

[2] D. Lazer, M. Baum, N. Grinberg, L. Friedland, K. Joseph, W. Hobbs, and C. Mattsson, "Combating fake news: An agenda for research and action," *Harvard Kennedy School, Shorenstein Center on Media, Politics and Public Policy*, vol. 2, 2017.

[3] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[3]The model adopted in the current paper is available here: http://www.di.unito.it/~sulis/diffusion-processes, as well as on Modeling Commons: http://modelingcommons.org/browse/one_model/6403

[4]ccl.northwestern.edu/netlogo/docs/behaviorspace.html

[4] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[5] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016.

[6] B. Nyhan and J. Reifler, "When corrections fail: The persistence of political misperceptions," *Political Behavior*, vol. 32, no. 2, pp. 303–330, 2010.

[7] S. Lewandowsky, U. K. Ecker, C. M. Seifert, N. Schwarz, and J. Cook, "Misinformation and its correction: Continued influence and successful debiasing," *Psychological science in the public interest*, vol. 13, no. 3, pp. 106–131, 2012.

[8] C. Silverman, "Lies, damn lies and viral content," *Tow Center for Digital Journalism, Columbia University*, 2015.

[9] M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, "The spreading of misinformation online," *Proceedings of the National Academy of Sciences*, vol. 113, no. 3, pp. 554–559, 2016.

[10] C. Shao, G. L. Ciampaglia, A. Flammini, and F. Menczer, "Hoaxy: A platform for tracking online misinformation," in *Proceedings of the 25th international conference companion on world wide web*, 2016, pp. 745–750.

[11] Y. Moreno, M. Nekovee, and A. F. Pacheco, "Dynamics of rumor spreading in complex networks," *Physical Review E*, vol. 69, no. 6, p. 066130, 2004.

[12] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 665–674.

[13] D. Trpevski, W. K. Tang, and L. Kocarev, "Model for rumor spreading over networks," *Physical Review E*, vol. 81, no. 5, p. 056102, 2010.

[14] F. Jin, E. Dougherty, P. Saraf, Y. Cao, and N. Ramakrishnan, "Epidemiological modeling of news and rumors on twitter," in *Proceedings of the 7th workshop on social network mining and analysis*, 2013, pp. 1–9.

[15] M. Tambuscio, G. Ruffo, A. Flammini, and F. Menczer, "Fact-checking effect on viral hoaxes: A model of misinformation spread in social networks," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 977–982.

[16] U. Wilensky and W. Rand, *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. The MIT Press, 2015.

[17] S. F. Railsback and V. Grimm, *Agent-based and individual-based modeling: a practical introduction*. Princeton university press, 2019.

[18] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the national academy of sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002.

[19] J. H. Holland, *Signals and boundaries: Building blocks for complex adaptive systems*. Mit Press, 2012.

[20] E. Sulis and A. Di Leva, "An agent-based model of a business process: The use case of a hospital emergency department," in *International Conference on Business Process Management*. Springer, 2017, pp. 124–132.

[21] D. J. Daley and D. G. Kendall, "Epidemics and rumours," *Nature*, no. 204, p. 1118, 1964.

[22] D. P. Maki and M. Thompson, "Mathematical models and applications: with emphasis on the social life, and management sciences," Tech. Rep., 1973.

[23] L. Zhao, X. Qiu, X. Wang, and J. Wang, "Rumor spreading model considering forgetting and remembering mechanisms in inhomogeneous networks," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 4, pp. 987–994, 2013.

[24] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[25] J. C. Thiele, "R marries netlogo: Introduction to the rnetlogo package," *Journal of Statistical Software*, vol. 58, no. 2, pp. 1–41, 2014.

[26] M. Jaxa-Rozen and J. H. Kwakkel, "Pynetlogo: Linking netlogo with python," *Journal of Artificial Societies and Social Simulation*, vol. 21, no. 2, 2018.

[27] W. Rand, J. Herrmann, B. Schein, and N. Vodopivec, "An agent-based model of urgent diffusion in social media," *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 2, p. 1, 2015.

[28] K. Lee, S. Kim, C. O. Kim, and T. Park, "An agent-based competitive product diffusion model for the estimation and sensitivity analysis of

social network structure and purchase time distribution," *Journal of Artificial Societies and Social Simulation*, vol. 16, no. 1, p. 3, 2013.

[29] Z. Chen, "An agent-based model for information diffusion over online social networks," *Papers in Applied Geography*, vol. 5, no. 1-2, pp. 77–97, 2019. [Online]. Available: https://doi.org/10.1080/23754931.2019.1619193

[30] M. Tambuscio, D. F. Oliveira, G. L. Ciampaglia, and G. Ruffo, "Network segregation in a model of misinformation and fact-checking," *Journal of Computational Social Science*, vol. 1, no. 2, pp. 261–275, 2018.

[31] M. Tambuscio and G. Ruffo, "Fact-checking strategies to limit urban legends spreading in a segregated society," *Applied Network Science*, vol. 4, no. 1, p. 116, 2019.

[32] E. Sulis, P. Terna, A. Di Leva, G. Boella, and A. Boccuzzi, "Agent-oriented decision support system for business processes management with genetic algorithm optimization: an application in healthcare," *Journal of Medical Systems*, vol. 44, no. 9, pp. 1–7, 2020.