

## A Novel Collective User Web Behavior Simulation Method

Hongri Liu<sup>1,2,3</sup>, Xu Zhang<sup>1,3</sup>, Jingjing Li<sup>1,3</sup> and Bailing Wang<sup>1,3,\*</sup>

<sup>1</sup>College of Computer Science & Technology, Harbin Institute of Technology at Weihai, Weihai, 264209, China

<sup>2</sup>Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, 518000, China

<sup>3</sup>Research Institute of Cyberspace Security, Harbin Institute of Technology, Harbin, 150001, China

\*Corresponding Author: Bailing Wang. Email: wbl@hit.edu.cn

Received: 20 June 2020; Accepted: 17 July 2020

**Abstract:** A collective user web behavior simulation is an import means for generating a large-scale user network behavior in a network testbed or cyber range. Existing studies almost focus on individual web behavior analysis and prediction, which cannot simulate human dynamics that widely exist in large-scale users' behaviors. To address these issues, we propose a novel collective user web behavior simulation method, in which an algorithm for constructing a connected virtual social network is proposed, and then a collective user web behavior simulation algorithm is designed on the virtual social network. In the simulation method, a new epidemic information dissemination algorithm based on the SIR model is proposed to drive the user web behavior with Breadth—First Search algorithm on the connected virtual social network. We specially build an experiment environment with 12 servers by using Docker container technology and then perform a wide range of experiments with different user scales to evaluate the method. The experimental results demonstrate that not only the degrees of the social network but also the time intervals of the collective users' web behavior can be well fitted to a power-law distribution and show that our simulation method can well simulate a collective user web behavior.

**Keywords:** Web behavior simulation; virtual social network; human dynamics; power-law distribution

### 1 Introduction

Facing an increasingly severe network security issues in the Internet, researchers and organizations have constructed all kinds of large-scale of network testbeds to simulate the Internet and then conduct experiments for analyzing these issues, such as Emulab Network Testbed, NCR (National Cyber Range) [1] and StarBED [2]. In large-scale network testbeds or cyber ranges, virtualization technologies are widely employed to simulate the hardware and network, which constitute the infrastructure of the network testbed. Meanwhile, the user network behavior on network testbed is generated by simulating the user behavior on the Internet [3,4]. In recently years, for predicting or simulating individual behaviors on some applications, many user network behavior models and simulation methods have been proposed [5–11]



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

and achieved promised results. However, these methods cannot be directly applied in collective user network behavior simulation because of the lack of interactions among individuals and other issues.

Based on human dynamics, many researchers have devoted the studies on behavior analysis [12–15] and have achieved fruitful results. However, a collective user network behavior simulation is rarely touched in current studies for the challenges in the scale and the method of simulation in a network testbed. Our interest arises out of the question of how to perform a collective user network behavior simulation that follows human dynamics in a real network environment. In the simulation, the key is how to drive the collective user to perform one kind of network behavior while following the laws of human dynamics.

In this paper, we study a collective user web behavior simulation method. In detail, we first construct a virtual social network in which each virtual node is modeled as an agent and the degrees of the agents follow a power-law distribution, a form of heavy-tailed distribution. Then, we simulate the information dissemination process among the agents based on an epidemic information dissemination model, in which all agents reply a message to a specific post in the forum after infected and then infects its neighbors. We finally validate that the time intervals between two consecutive replies follow a power-law distribution, which agrees with the laws of human dynamics. The experimental results demonstrate that our method is capable of simulating the collective user web behavior realistically.

The goal of our work is to study the method of simulating a collective user network behavior in a large network testbed or cyber range, instead of studying the mechanism of the heavy-tailed phenomenon in the real social network such as Twitter and Facebook. To the best of our knowledge, this work is the first study on the method of simulating a collective user network behavior in a real network environment.

In summary, the main contributions of our work are listed as follows:

1. We design a virtual social network construction method, which improves the PLOD (power-law out degree) [16] algorithm with the BA (Barabási–Albert) model [17,18] and has the ability to quickly establish a connected virtual social network whose degree distribution follows a power-law distribution.
2. An information dissemination algorithm, the key of the simulation method, is proposed to drive the individual user's network behavior. We give the calculation method of the length of waiting time and define the process of the epidemic information dissemination in the algorithm.
3. We design a real network environment to evaluate the performance of the simulation method on the Docker container swarm. As an agent, each Docker container posts a reply to the specific post of the same forum by running our proposed algorithms in the connected social network. We validate that the time intervals of consecutive replies follow a power-law distribution.

The rest of the paper is organized as follows: Section 2 reviews related work, Section 3 introduces the approach, Section 4 presents our experiments and the result, and Section 5 discusses and concludes the paper.

## 2 Background

A collective user behavior simulation plays an essential role in multiple internet applications. Many researchers have focused on the prediction and simulation of individual network behavior.

### 2.1 User Network Behavior Modeling and Predicting

Kaderka et al. [19] build a tool named BeCos, which not only allows users to specify system and component behavior but also enables system engineers to clearly define behavior in a semantically rigorous manner with behavior ontology and scenario ontology. Yang et al. [20] analyzed the influence of multiple factors on email reply behavior and developed a model to predict whether a receiver will reply

to the sender. Moreover, they characterized the influence of rich factors on email reply behavior. Incorporating structure, usage, and content data from a real website, Loyola et al. [8] proposed a new method to analyze web user behavior by using an Ant Colony Optimization algorithm, which can explain approximately 80% of real usage with a predefined similarity measure. Si et al. [21] studied the users' post and reply behaviors in Tianya, an internet forum, and found that users' posts, replies, in-degrees and out-degrees of replied message distribution all follow a power-law distribution, which is consistent with the achievements of other researchers.

In terms of the analysis approaches, Rajabi et al. [9] defined user behavior categories using an unsupervised learning algorithm and then proposed a data-driven approach to analyze user behavior. Liu et al. [10] presented a new user behavior modeling method on the user's clicking behavior of search engines, in which a convolutional neural network (CNN) architecture is constructed to build click models, and the parameters from traditional click models are used to restrict the meaning in the hidden layer of the CNN. This model is considered as the best with respect to the evaluation metric of click perplexity. Park et al. [7] analyzed the behavior of image search on a large-scale query log from Yahoo Image Search, which assumed that a user's behavior is dependent on the query type. Aiming at revealing high-level patterns in the behavior of game players, Sifa et al. [22] performed four experiments on a 6 million player dataset covering a total playtime of over 5 billion hours across more than 3000 games distributed via the Steam platform, which provided unique insights into the behavior of the game player. Xiao et al. [23] studied human behavior in Sina Microblog of China and found that not only on the individual level but also on the group level, the message releasing time intervals obey power-law distribution.

## **2.2 User Network Behavior Simulation**

Agent-based models (ABM) [24,25] are computational models that simulate the actions and interactions of autonomous agents (individual or collective entities) to assess their effects within an environment. In company with the increase of computing power, ABM based user behavior simulation techniques (e.g., BDI) have been widely used in social simulation [26–29], e.g., Kampmann et al. [29] proposed an automatic parameter identification approach for personality driving behavior models by comparing the behavior data of human and artificial drivers in the same virtual environment. This method can also be applied to compare human and artificial driving behavior. Also, Flamino et al. [30] concluded that human patterns of temporal preference are consistent and resilient by analyzing users of several real-world datasets. Further, they integrated those patterns into large-scale agent-based models and simulated the users' activity to validate predictive accuracy. Xu et al. [31] build an agent-based model to simulate the knowledge transfer and integration processes among irrational individuals and then explored the effects of individual irrationality and network structure on collective performance. Although ABM is widely used to simulate human behavior in multiple research areas such as auto driving, the research on simulating user network behavior is quite rare. In [32], the BP-BDI model is proposed to simulate the user's behavior on email, in which a behavior learning model based on BP neural network is used to learn the user's mail behavior. Diallo et al. [33] first analyzed the web simulation by using the principles of service orientation, platform independence and interoperability, and then proposed a layered approach for the web simulation. In terms of Human-machine interaction (HMI) simulation, Amirkhanyan et al. [5] proposed a user behavior simulation scheme based on user behavior state graphs, which are capable to generate simple user behavior on Windows-family virtual machines.

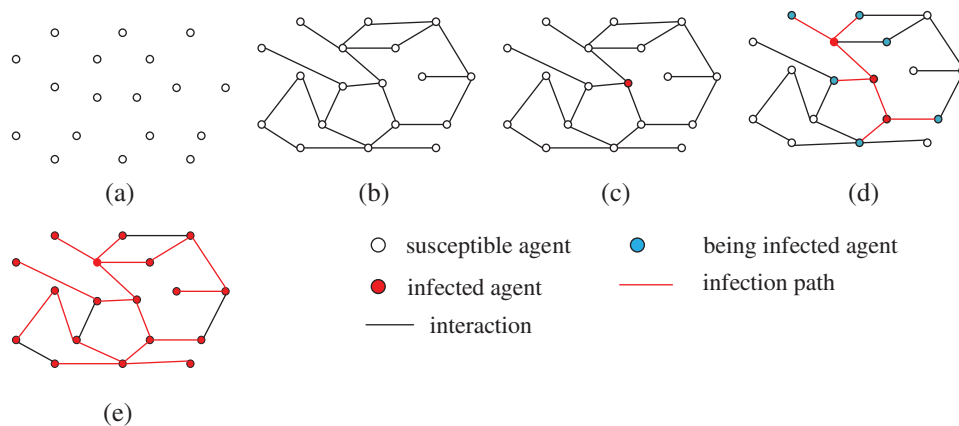
To sum up, there are many researches on individual behavior modeling and simulation based on ABM and make some notable research results, whereas, in collective user network behavior simulation, these research results cannot be applied directly. On the one hand, different from traffic simulations and crowding behavior simulation in an emergency, the collective user's network behavior is spontaneous and

lasting for several days or even months. On the other hand, the individual network behavior simulation methods are not suitable for collective user behavior simulation for lack of interactions among users.

In this paper, we propose a novel collective user web behavior simulation method based on ABM. Firstly, we construct a connected undirected virtual social network, and then use epidemic information dissemination model to simulate the information dissemination process among the agents in the virtual social network to drive individual user's web behavior simulation. The time calculation method makes the time intervals of two consecutive network behaviors follow a power-law distribution. It should be noticed that our work is different from [23], in which they simulated multi-agent interactions in a real-life social network topology to verify whether the results are consistent with empirical observations. In addition, their simulation happened in an emulation environment without any real network behavior. However, our work is to explore a simulation method that can construct a virtual social network and simulate a collective user's web behavior that follows human dynamics in a real network environment.

### 3 Research Method

In this section, we first propose a connected virtual social network construction algorithm by combing the PLOD algorithm with the BA model. In the virtual social network, each node is modeled as one agent. We divide the agents into three categories, the susceptible agent, the infected agent and the recovered agent. The SIR is modified, named as s\_SIR, to simulate the information dissemination among the agents, which drives individual network behavior simulation. Taking web behavior for example, one agent selected randomly from the social network replies to a specific post of the same forum after  $\Delta t_i$  ( $i \in [1, n]$ ) length of time and then infects its neighbors. The infected agent will repeat the same process. The abstract simulation process of the collective user web behavior is demonstrated as Fig. 1.



**Figure 1:** The abstraction process of collective user web behavior simulation

In Fig. 1, (a) shows isolated agents, (b) a virtual social network constructed by our algorithm, (c) one agent is infected (individual web behavior simulation), (d) The neighbors are infected, and (e) all the agents are infected (collective user web behavior simulation complete). We assume one infected agent will recover if it has infected all its susceptible neighbors and the recovered agent will never be infected anymore. In the information dissemination process, two connected agents communicate with each other to simulate the infection process by using UDP. Once infected, the agent will perform web behavior simulation after  $\Delta t_i$  ( $i \in \{1, 2, \dots, n\}$ ) length of time.

### 3.1 Virtual Social Network Construction Algorithm

There are two classical virtual social network generation algorithms, one is the PLOD, and the other is the BA model. The degree distribution resulting from the BA model follows a power-law distribution of the form  $P(k) \sim k^{-3}$ , which cannot generate other degree distributions directly. The PLOD algorithm can construct a virtual social network that the degrees of the agents obey power-law distribution by assigning a degree to each agent. However, it cannot guarantee that the virtual social network is connected. To construct a connected virtual social network with specified degree centrality, we improve the PLOD algorithm with the BA model, which makes the degrees of the agents follow a power-law distribution. The algorithm is described as Algorithm 1.

---

**Algorithm 1.** A connected social network construction with specified degree

---

Input:  $n$  is the number of agents in the virtual social network.

$degree\_max$  is the maximum degree of the social network.

$S_1 = \{n_1, n_2, \dots, n_n\}$ ,  $S_2 = \{\}$ ,  $S_3 = \{\}$ ,  $n_i$ , ( $i \in \{1, 2, \dots, n\}$ ) represent the agents in the social network.

Output: a virtual connected social network in which the degrees of the agents follow power-law distribution.

Description:

**Begin:**

Step 1. Generate a set of  $n$  integers  $\{d_1, d_2, \dots, d_n\}$  that are consistent with the power-law distribution, where  $d_i$  is the degree of  $n_i$  in  $S_1$  and  $\max \{d_1, d_2, \dots, d_n\} \leq degree\_max$ ;

Step 2. Randomly select an agent  $n_i$  from  $S_1$  as the initial network and add it to  $S_2$ , then delete  $n_i$  from  $S_1$ ;

Step 3. Randomly select an agent  $n_i$  from  $S_1$  and link it to a random agent  $n_k$  in  $S_2$  with probability  $\alpha = (d_i + d_k) / \text{SUM}(d_j, \text{where } n_j \in S_1)$ .  $d_i = d_i - 1$ ,  $d_k = d_k - 1$  ( $i \in [1, \text{length}(S_1)]$  and  $k \in [1, \text{length}(S_2)]$ );

Step 4. if  $d_g == 0$  ( $n_g \in \{n_i, n_k\}$ ) then add  $n_g$  to  $S_3$ , else add agent  $n_i$  to  $S_2$ ;

Step 5. if  $S_1 \neq \{\}$  then jump to Step 3 else jump to Step 6;

Step 6. Random select two agents  $n_i$  and  $n_k$  from  $S_2$  whose degrees are not 0 and link them with probability  $\alpha = (d_i + d_k) / \text{SUM}(d_j, \text{where } n_j \in S_2)$ .  $d_i = d_i - 1$ ,  $d_k = d_k - 1$ ;

Step 7. if  $d_g == 0$  ( $n_g \in \{n_i, n_k\}$ ) then add  $d_g$  to  $S_3$ ;

Step 8. jump to Step 6 until  $S_2 = \{\}$ ;

**End**

---

In Algorithm 1, the agents are divided into three non-intersecting collections,  $S_1$ ,  $S_2$ , and  $S_3$ .  $S_1$  is the collection of isolated agents in the social network,  $S_2$  is the collection of the connected agents whose degrees are not zero in the virtual network, and  $S_3$  is the collection of agents whose degrees are 0. The virtual social network grows up until  $S_1 = \{\}$  and  $S_2 = \{\}$ . The  $\alpha \rightarrow 1$  when  $|S_2| \rightarrow 0$ . Therefore, the virtual social network constructed is a connected network. Also, the time complexity of Algorithm 1 is  $O(n^2)$ . Furthermore, we implement the method proposed in [34] to calculate the set  $\{d_i\}$  for the degrees of the agents.

We assume that the  $\{d_i\}$  satisfies a power-law distribution  $p(d_i) \propto d_i^{-\beta}$ , and an upper limit  $M$  is assigned to  $degree\_max$  in Algorithm 1. The purpose is to generate a set  $\{d_i | d_i \in [1, M] \wedge d_i \text{ is an integer} \wedge p(d_i) \propto d_i^{-\beta}\}$ . The cumulative distribution is

$$F(d) = \sum_{i=1}^d P(d) \quad (1)$$

Accordingly, Eq. (1) can be approximated by a continuous form, as

$$F(d) = \int_1^d p(d) \quad (2)$$

The ratio of  $F(\Delta t)$  ( $\Delta t \in [1, M]$ ) to  $F(M)$  is denoted by  $r$ , as

$$r = \frac{\int_1^{\Delta t} x^{-\beta} dx}{\int_1^M x^{-\beta} dx} = \frac{\frac{\Delta t^{1-\beta} - 1}{1-\beta}}{\frac{M^{1-\beta} - 1}{1-\beta}} = \frac{\Delta t^{1-\beta} - 1}{M^{1-\beta} - 1} \quad (3)$$

$0 \leq r \leq 1$  and

$$\Delta t = (r \times (M^{1-\beta} - 1) + 1)^{\frac{1}{1-\beta}} \quad (4)$$

If  $M \rightarrow \infty$ , then  $M^{1-\beta} \rightarrow 0$  and hence (4) can be approximated as

$$\Delta t \approx (1 - r)^{\frac{1}{1-\beta}}$$

For different  $\beta$ , the average intervals  $\text{avg}(\Delta t)$  is different. To exclude the bias influenced caused by  $\text{avg}(\Delta t)$ , a correction factor  $u(\beta)$  is introduced to Eq. (5), as

$$\Delta t' = u(\beta) \times (r \times (M^{1-\beta} - 1) + 1)^{\frac{1}{1-\beta}} \quad (6)$$

where  $u(\beta)$  satisfies

$$\frac{\text{avg}(\Delta t)_{\beta_1}}{\text{avg}(\Delta t)_{\beta_2}} = \frac{u(\beta_2)}{u(\beta_1)} \quad (7)$$

For simplicity but without loss of generality, we set  $u(1) = 1$ , hence

$$u(\beta) = \frac{\text{avg}(\Delta t)_1}{\text{avg}(\Delta t)_\beta} \quad (8)$$

To sum up, given  $M$  and  $\beta$ ,  $u(\beta)$  can be estimated directly. The set  $\{d_i\}$  can be generated by selecting a random  $r \in [0, 1]$  and then calculating  $\Delta t_i$  by using Eq. (6). Furthermore, if  $\Delta t'$  is not an integer, we further rectify  $\Delta t'$ . Namely,  $\Delta t' = \lceil \Delta t' \rceil$  with probability  $\Delta t' - \lfloor \Delta t' \rfloor$  while  $\Delta t' = \lfloor \Delta t' \rfloor$  with probability  $\lceil \Delta t' \rceil - \Delta t'$ . The virtual relationship is established and maintained by the communication among agents.

### 3.2 Web Behavior Simulation Algorithm

We apply the SIR model to simulate the information dissemination process among the agents and meanwhile trigger the web behavior simulation, which is named the s-SIR algorithm as described in Algorithm 2. More specifically, the status of agents in the virtual social network is divided into three types, namely, susceptible agents, infected agents, recovered agents.

Before describing our s-SIR algorithm, we first give the following definitions:

1. In the initial state, each agent in the social network is susceptible.
2. Each agent is irrational; in other words, it will infect all susceptible neighbors.

3. Each infected agent will become a recovered agent after infecting all neighbors and will never be infected anymore.

The infection process is implemented by one agent transferring a message to its neighbors one by one in the virtual social network. The agent is viewed as infected once it receives a specified message from a neighbor, and then replies to the same post after waiting a specified length of time.

---

**Algorithm 2.** The algorithm of collective user web behavior simulation based on the SIR model

---

Input: A virtual social network  $G = (V, E)$ , where  $v_i$  is the agent node and  $e_i$  is the connection. A collection  $\{ \langle v_i, t_i, p_i \rangle, i = 1, \dots, n \}$ , where  $v_i$  is the agent name,  $t_i$  is the length of the waiting time of  $v_i$  from being infected to performing web behavior simulation, and  $p_i$  is the probability of being infected.

Output: The collective user web behavior.

Description:

**Begin**

1. Randomly select an agent  $v_i$  from  $G$  as the first infected agent;
2.  $v_i$  waits for a time period of length  $t_i$  and then performs web behavior simulation, namely, replies to the specific post;
3.  $v_i$  infects each susceptible neighbor agent  $v_j$  with probability  $p_j$  ( $j = 1, \dots, k$ , where  $k$  is the number of susceptible neighbor agents), and  $v_i$  becomes a recovered agent;
4. for each infected agent  $v_j$ , jumps to **Step 2** until all the agents in the  $G$  become recovered agents;

**End**

---

In Algorithm 2, each infected agent  $v_i$  ( $i \in \{1, 2, \dots, n\}$ ) will wait for the length of  $t_i$  before replying a message to an specific post of the same forum, and then send a message to each susceptible neighbors  $v_j$  with Breadth—First Search. At some point of time, multiple agents are performing web behavior simulations simultaneously. The length of waiting time  $t_i$  ( $i = 1, \dots, n$ ) also follows a power-law distribution. The calculation method of  $t_i$  is given in the Eqs. (1)–(7). To simulate the web behavior realistically, we use an expansion factor  $\gamma$  to extend the web behavior simulation process. Namely, the length of the waiting time  $t_i$  is expanded by  $\gamma$  times.

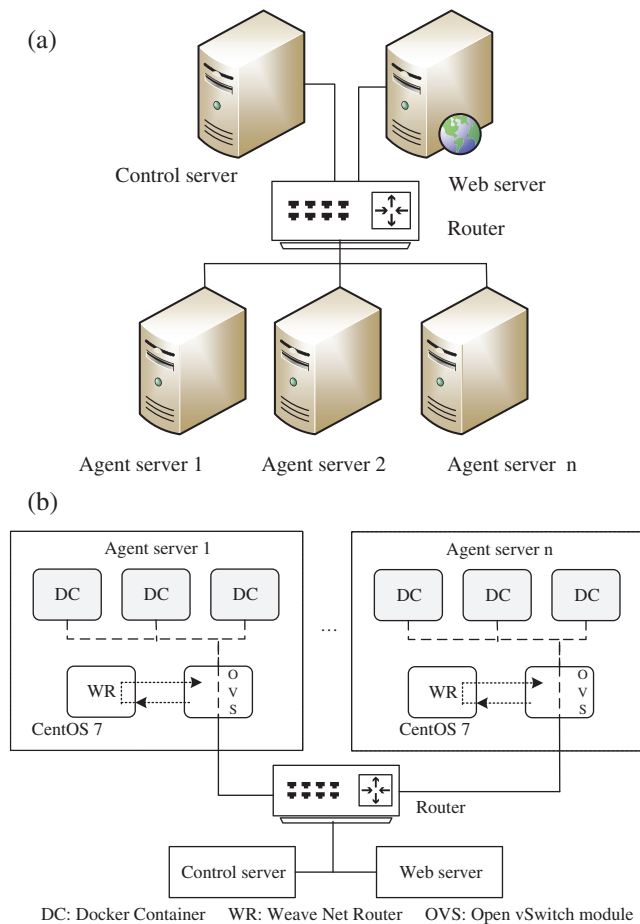
To simulate collective user web behavior, a construction method of a virtual social network is proposed, of which the degrees of nodes follow a power-law distribution. Next, a web behavior simulation method named s-SIR based on SIR has been designed to drive each agent performing a reply to the post in the virtual social network. Not only the distribution of agents' degrees but also the time intervals of post behaviors follow a power-law distribution. In other words, we abstract an undirected graph  $G = \{v_i, d_i, t_i\}$  to describe the agents in the virtual social network, where  $d_i$  is the degree of  $v_i$ , and  $t_i$  is the length of waiting time from being infected to perform the simulation. Once the  $G$  is determined, the collective user web behaviors can be simulated, and finally, the degree and waiting time intervals follow a power-law distribution.

## 4 Experiments

In our experiments, we use a Docker container swarm to simulate the collective user. We deploy 6000 and 10000 Docker containers respectively to evaluate our proposed simulation method. First, the virtual social network constructed method is validated, and then we perform web behavior simulation in the virtual social network with our proposed s-SIR algorithm. Finally, a power-law function is used to fit test data, and the standard error is employed to evaluate the realistic of the simulation method. Our experiments demonstrate that the method is capable of simulating the collective user web behavior realistically.

#### 4.1 Experiment Setup

We deploy 12 high-performance servers (Intel Xeon E5-2650 V4, 64G memory, 2 NICs and 2T hard disk) to conduct the experiments, of which 10 servers running Docker containers are used to perform collective user web behavior simulation and another server serves as a web server where a forum is deployed and the 12th server is used to generate the virtual social network and calculate the time of replying to the post of the forum. Each agent replies a message to the specific assigned post of the forum in the web server. The agent is implemented by one Docker container, and the Weave Net networking toolkit is employed to connect all the agents. The physical and logical topologies of the experiment is shown in Fig. 2.



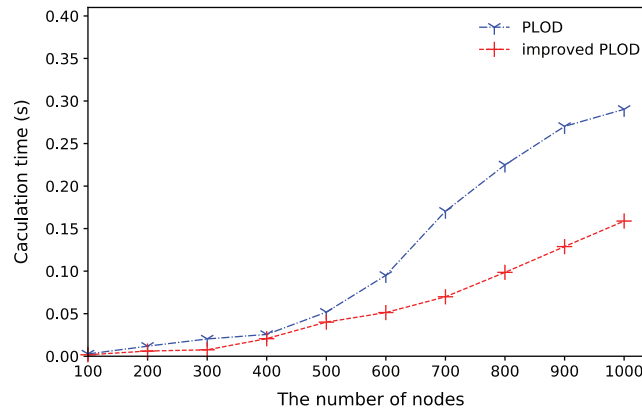
**Figure 2:** (a) The physical topology of the experiment network. (b) The logical topology of the agent server

As shown in Fig. 2, the servers are connected by a router in physical while the agents (Docker containers) are connected by Open vSwitch, a virtual switch, controlled by Weave Net in logical. The implementations of Algorithms 1 and 2 are deployed in the control server and the agents, respectively, to perform the simulation. We deploy 10000 agents at most to evaluate our proposed method, and all the system clocks of the servers are synchronized by NTP before each round of the experiment. Furthermore, the clock synchronization error of 1000 Docker containers in one server is tested and the maximum error is less than 1 s.



#### 4.2 Calculation Time Comparison

First, we compare the calculation time between the PLOD algorithm and the improved PLOD algorithm (Algorithm 1). The number of nodes ranges from 100 to 1000 with the same degree distribution function. The experimental result is shown in Fig. 3.



**Figure 3:** The comparison of the calculation time between the PLOD and improved PLOD

As shown in Fig. 3, the calculation time increases as the number of nodes increase. However, the improved PLOD algorithm grows slowly compared with the PLOD. For the improved PLOD algorithm, the maximum calculation time is 0.15 s with 1000 nodes, which is much less than that of the PLOD algorithm.

#### 4.3 Virtual Social Network Construction Experiment

Next, we evaluate the virtual social network construction method with 1000 agents where  $\beta = 2.1$  in Algorithm 1. The degree distribution and generated graph are demonstrated in Fig. 4.

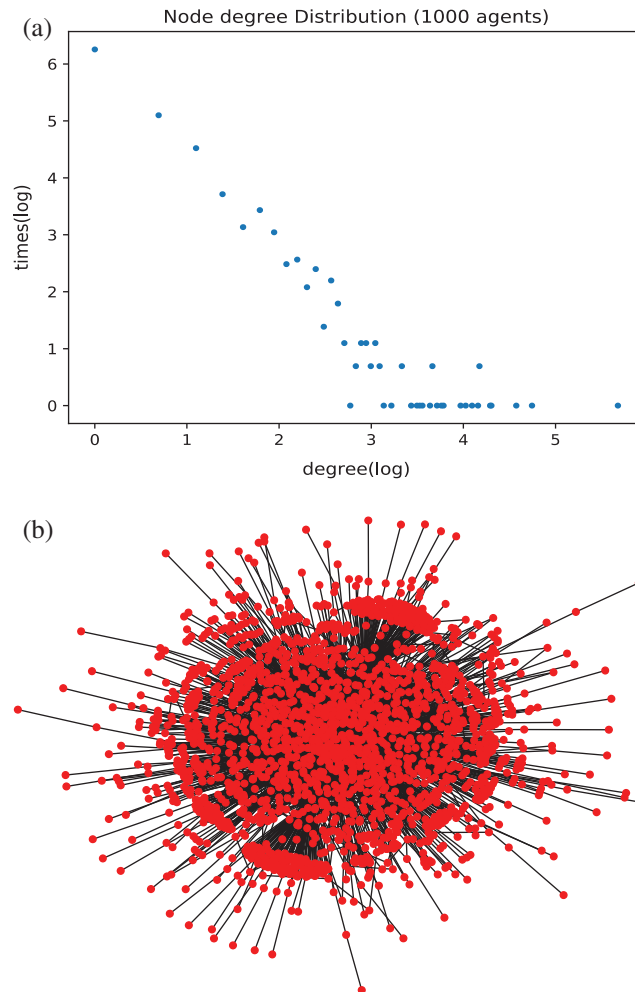
In Fig. 4a, the x-axis is the degrees of the agents, and the y-axis represents the number of times of the same degree that appears in the experiment. In the experiment, there are 44 different degrees from  $\{1, 2, \dots, 292\}$  for the 1000 agents. Furthermore, most degrees of the nodes are smaller and follow heavy-tailed distribution in the log-log plot. The power-law function appears as a straight line with slope  $k = -2.1$ .

#### 4.4 Time Interval Distribution Experiment

In this section, we construct a virtual network with 6000 and 10000 agents, respectively. Firstly, one agent is randomly selected as the source of the infection. Then we drive the web behavior simulation according to the network topology and the length of waiting time. In this section, we set all the infected probability  $p_j = 1$  ( $j = 1, \dots, n$ ), and the length of each waiting time  $t_i$  ( $i = 1, \dots, n$ ) is multiplied by expansion factor  $\gamma$  ( $\gamma = 1500$ ) to extend the waiting time. Finally, we calculate the time intervals of the reply to the post. The time intervals are indirectly obtained from the database of the webserver of the forum. The time interval distribution of 10000 users is illustrated in Fig. 5.

As in Fig. 5, the x-axis is the time interval and the y-axis is the number of counts of the time interval. We can conclude:

1. Most of the time interval of agents are located in the tail part,
2. The smaller the time interval, the larger the counts and
3. The larger the time interval, the smaller the counts.



**Figure 4:** (a) The distribution of the virtual social network log–log plot. (b) The graph of virtual social network

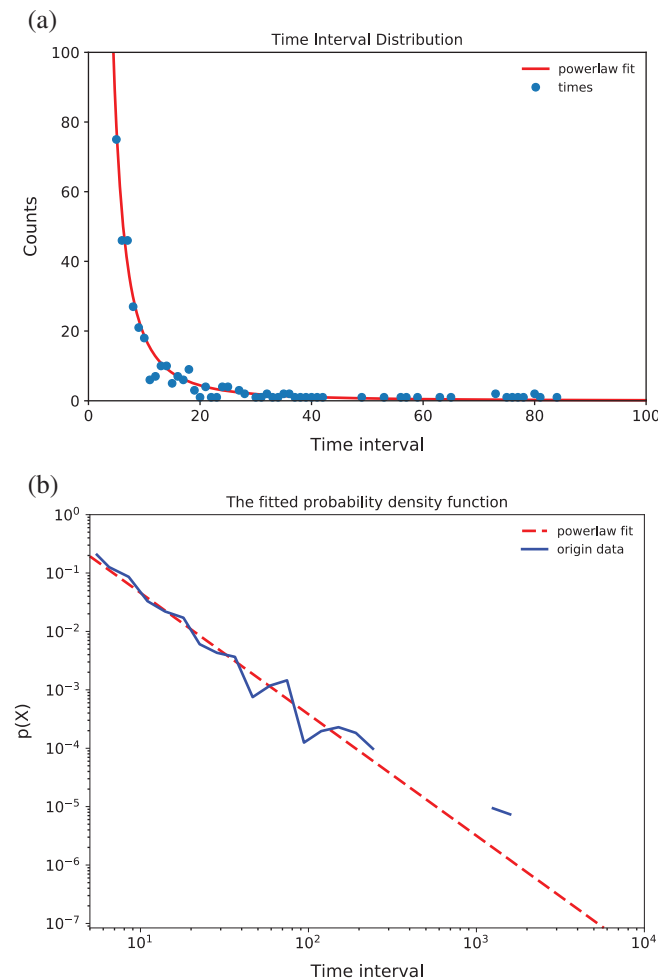
We fit the pow-law function, which is  $y = 6640 \times x^{-2.1}$ , and the probability density function using logarithmic operations is  $y = -2.1 \times x + 3.8$  with standard error 0.0153.

To compare with the simulation result of 10000 users, the experimental result of 6000 users is shown in Fig. 6.

As demonstrated in Fig. 6a, the time intervals also exhibit a power-law distribution, which is similar to the 10000 user's simulation. The probability density function using logarithmic operations is  $y = -2.0 \times x + 3.7$  with standard error 0.0156. Comparing Fig. 6 with Fig. 5, we can conclude that the standard error does not change obviously (less than 2%). Hence, the performance of the simulation method is stable.

#### 4.5 Standard Error Comparisons Experiment

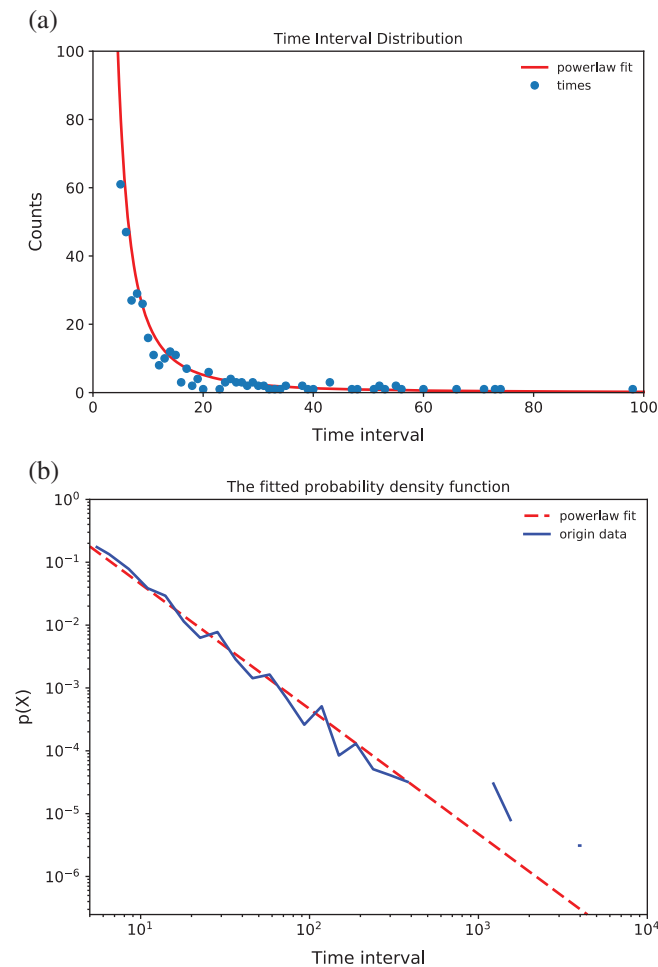
Finally, we perform extensive experiments to evaluate the standard error of the fitted pow-law function, in which  $\beta = 3, 4, 5$  in Algorithm 2 and the waiting time expansion factor  $\gamma = 200, 300, 500, \text{ and } 1500$  with 6000 agents and 10000 agents, respectively. The experimental results are shown in Fig. 7.



**Figure 5:** (a) The distribution of time intervals of replies to the post with 10000 users. (b) The logarithmic plot of the time intervals of replies to the post with 10000 users

From Fig. 7, we can conclude that the standard error does not increase obviously with the increase of the agents' number. The max standard error appears when  $\gamma = 200$  and  $\beta = 5$  (Algorithm 1) in both figures. Furthermore, the standard error grows as  $\beta$  increases when  $\gamma$  is fixed. The standard error decreases as  $\gamma$  increases when  $\beta$  is fixed. Furthermore, the simulation process lasts 37.1 h when  $\beta = 3$  and  $\gamma = 1500$  with 10000 agents, and 14.6 h with 6000 agents. So, the longer the waiting time, the better the fitting effect and the closer to the real collective user web behavior in a specific range of the length of waiting time.

In our experiments, we first validate the connected social network construction algorithm with specified degree. Compared with the PLOD algorithm, the calculation cost of our proposed method is smaller and ensures that the generated social networked is connected. Base on the virtual social network, we then deploy 6000 and 10000 Docker containers with ten servers to simulate the collective user web behavior with the s-SIR algorithm respectively. By using UDP, we simulate the infection process that occurs among virtual users. The time intervals of the web behavior of the collective user are fitted to pow-law function with a lower standard error, which fits the human dynamics. From the experiments, we can conclude that the standard error is stable when increasing the scale of the users, and the standard error grows when increases  $\beta$  or decreases  $\gamma$ .



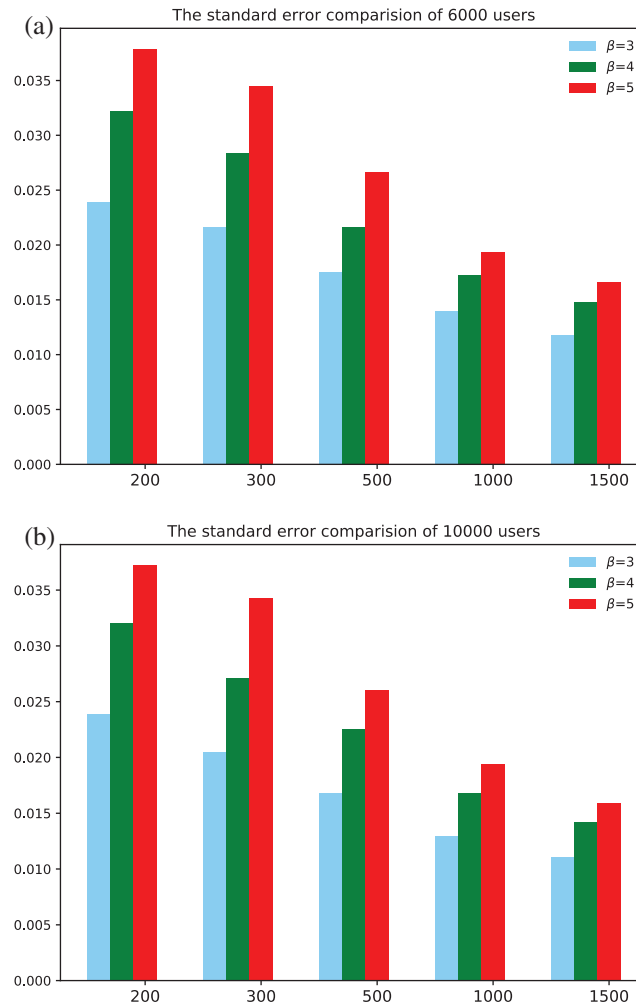
**Figure 6:** (a) The distribution of time intervals of replies to the post with 6000 users. (b) The logarithmic plot of the time intervals of replies to the post with 6000 users

## 5 Discussion and Conclusion

In this work, we have studied the collective user's web behavior simulation method and validated our proposed method by driving 6000 and 10000 agents to reply to a specific post of the same forum. First, a construction method of a virtual social network whose degrees follow a power-law distribution is proposed. Then, we use a new data structure  $G = \{v_i, d_i, t_i\}$  to describe the agents in the virtual network. Based on this network, we design an algorithm named s-SIR to drive each agent to perform web behavior simulation. A calculation method, which can generate a data collection following a power-law distribution, is implemented to create the degrees and the waiting time of the agents in the virtual social work. The degree collection is employed to construct the virtual social network, and the waiting time collection is used to control the message dissemination process by our s-SIR algorithm. We have ignored the time of information dissemination and the replies to the post due to the expansion factor  $\gamma$  is much longer than those time. The experimental results show that the time intervals of the web behavior follow a power-law distribution.

To the best of our knowledge, this is the first work to study on the method of simulating the collective user web behavior in a real network environment. The study results can be widely employed in all kinds of

network testbeds to construct a real network scenario in the construction of a large-scale users' network behavior that follows human dynamics, such as mail behavior and web behavior.



**Figure 7:** (a) The standard error comparison with different  $\beta$  and  $\gamma$  under 6000 users. (b) The standard error comparison with different  $\beta$  and  $\gamma$  under 10000 users

Since our work is the first research in the area of collective user network simulation, it comes with certain limitations. One is that we only simulate 1,0000 users' web behavior because of the limitation of the hardware. Another possible limitation is that we cannot directly calculate the collection of power-law distribution for degree and waiting time given the distribution function of collective user behavior time interval. In the future, we may want to study the power-law distributions calculation method of degree and waiting time, given the collective user behavior distribution.

**Funding Statement:** This research was supported by National Key Research and Development Plan under Grant 2017YFB0801804, Key Research and Development Plan of Shandong Province under Grant 2017CXGC0706, Peng Cheng Laboratory Project of Guangdong Province PCL2018KP004, frontier science and technology innovation of China under Grant 2016QY05X1002-2, national regional innovation center scientific and technological special projects Grant 2017QYCX14, University Co-construction Project in Weihai City.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] B. Ferguson, A. Tall and D. Olsen, “National cyber range overview,” in *IEEE Military Communications Conf.*, Baltimore, MD, pp. 123–128, 2014.
- [2] T. Miyachi, K. Chinen and Y. Shinoda, “StarBED and SpringOS architectures and their performance,” in *Int. Conf. on Testbeds and Research Infrastructures*, Berlin, Heidelberg, pp. 43–58, 2011.
- [3] B. Fang, Y. Jia, A. Li and W. Zhang, “Cyber ranges: State-of-the-art and research challenges,” *Journal of Cyber Security*, vol. 1, no. 3, pp. 1–9, 2016.
- [4] H. Kavak, J. J. Padilla and D. Vernon-Bido, “A characterization of cybersecurity simulation scenarios,” in *Proc. of the 19th Communications & Networking Symposium, Society for Computer Simulation International*, Pasadena, California, USA, pp. 1–8, 2016.
- [5] A. Amirkhanyan, A. Sapegin, M. Gawron, F. Cheng and C. Meinel, “Simulation user behavior on a security testbed using user behavior states graph,” in *Int. Conf. on Security of Information & Networks*, Sochi, Russia, pp. 217–223, 2015.
- [6] N. Li, W. Cushing, S. Kambhampati and S. Yoon, “Learning probabilistic hierarchical task networks as probabilistic context-free grammars to capture user preferences,” *ACM Transactions on Intelligent Systems & Technology*, vol. 5, no. 2, pp. 1–32, 2014.
- [7] J. Park, N. O’Hare, R. Schifanella, A. Jaimes and C. Chung, “A large-scale study of user image search behavior on the web,” in *Proc. of the 33rd Annual ACM Conf. on Human Factors in Computing Systems—CHI’15*, Seoul, Republic of Korea, pp. 985–994, 2015.
- [8] P. Loyola, P. E. Román and J. D. Velásquez, “Clustering-based learning approach for ant colony optimization model to simulate web user behavior,” in *IEEE/WIC/ACM Int. Conf. on Web Intelligence & Intelligent Agent Technology*, Lyon, France, pp. 457–464, 2011.
- [9] Z. Rajabi, A. Shehu and H. Purohit, *User Behavior Modelling for Fake Information Mitigation on Social Web*. Cham: Springer International Publishing, 2019.
- [10] Y. Liu, C. Wang, M. Zhang and S. Ma, “User behavior modeling for better Web search ranking,” *Frontiers of Computer Science*, vol. 11, no. 6, pp. 923–936, 2017.
- [11] J. Liu, M. Mitsui, N. Belkin and C. Shah, “Task, information seeking intentions, and user behavior: Toward a multi-level understanding of Web search,” in *Proc. of the Conf. on Human Information Interaction and Retrieval*, Glasgow, Scotland UK, pp. 123–132, 2019.
- [12] Y. Virkar and A. Clauset, “Power-law distributions in binned empirical data,” *Annals of Applied Statistics*, vol. 8, no. 1, pp. 89–119, 2014.
- [13] L. Leiva, “Automatic web design refinements based on collective user behavior,” in *CHI’12 Extended Abstracts on Human Factors in Computing Systems*, Austin, Texas, USA, pp. 1607–1612, 2012.
- [14] M. Mitrović, G. Paltoglou and B. Tadić, “Quantitative analysis of bloggers’ collective behavior powered by emotions,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 1, pp. 2005–2019, 2011.
- [15] D. Yang, D. Zhang and B. Qu, “Participatory cultural mapping based on collective behavior data in location-based social networks,” *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 3, pp. 1–23, 2016.
- [16] J. Esquivel-Gómez, E. Stevens-Navarro, U. Pineda-Rico and J. Acosta-Elias, “A growth model for directed complex networks with power-law shape in the out-degree distribution,” *Scientific Reports*, vol. 5, 7670, 2015.
- [17] V. Zadorozhnyi and E. Yudin, “Growing network: Nonlinear extension of the Barabasi-Albert model,” *Communications in Computer and Information Science*, vol. 487, pp. 432–439, 2014.
- [18] V. N. Zadorozhnyi and E. B. Yudin, “Structural properties of the scale-free Barabasi-Albert graph,” *Automation and Remote Control*, vol. 73, no. 4, pp. 702–716, 2012.

- [19] J. Kaderka, M. Rozek, J. Arballo, D. Wagner and M. Ingham, “The behavior, constraint, and scenario (BeCoS) tool: A web-based software application for modeling behaviors and scenarios,” in *AIAA Aerospace Sciences Meeting*, Kissimmee, Florida, USA, 2018.
- [20] L. Yang, S. Dumais, P. Bennett and A. Awadallah, “Characterizing and predicting enterprise email reply behavior,” in *Proc. of the 40th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Shinjuku, Tokyo, Japan, pp. 235–244, 2017.
- [21] X. Si and Y. Liu, “Power-law distribution of human behaviors in internet forums,” in *Int. Sym. on Intelligence Information Processing and Trusted Computing*, Huanggang, China, pp. 286–289, 2010.
- [22] R. Sifa, A. Drachen and C. Bauckhage, “Large-scale cross-game player behavior analysis on steam,” in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conf.*, Dortmund, Germany, pp. 1–8, 2015.
- [23] Y. Xiao, B. Wang, Y. Liu, Z. Yan, X. Chen *et al.*, “Analyzing, modeling, and simulation for human dynamics in social network,” *Abstract & Applied Analysis*, vol. 2012, no. 6684, pp. 552–582, 2012.
- [24] P. Caillou, B. Gaudou, A. Grignard, C. Truong and P. Taillandier, “A simple-to-use BDI architecture for agent-based modeling and simulation,” in *Advances in Social Simulation*, Springer, Cham, pp. 15–28, 2017.
- [25] C. Adam and B. Gaudou, “BDI agents in social simulations: A survey,” *Knowledge Engineering Review*, vol. 31, no. 3, pp. 207–238, 2016.
- [26] S. Chandramohan, M. Geist, F. Lefèvre and O. Pietquin, “User simulation in dialogue systems using inverse reinforcement learning,” in *Interspeech*, Florence, Italy, pp. 1025–1028, 2011.
- [27] S. Gottwalt, W. Ketter, C. Block, J. Collins and C. Weinhardt, “Demand side management—A simulation of household behavior under variable prices,” *Energy Policy*, vol. 39, no. 12, pp. 8163–8174, 2011.
- [28] K. Mezuman, S. Bauer and K. Tsigaridis, “Human behavior in global fire simulations,” in *AGU Fall Meeting Abstracts*, Washington, USA, 2018.
- [29] S. Kampmann, S. Seele, R. Herpers, P. Becker and C. Bauckhage, “Automatic mapping of human behavior data to personality model parameters for traffic simulations in virtual environments,” in *IEEE Conf. on Computational Intelligence and Games*, Tainan, Taiwan, pp. 336–343, 2015.
- [30] J. Flamino, W. Dai and B. K. Szymanski, “Modeling human temporal dynamics in agent-based simulations,” in *Proc. of the 2019 ACM SIGSIM Conf. on Principles of Advanced Discrete Simulation*, Chicago, IL, USA: ACM, pp. 99–102, 2019.
- [31] B. Xu, R. Liu and Z. He, “Individual irrationality, network structure, and collective intelligence: An agent-based simulation approach,” *Complexity*, vol. 21, no. S1, pp. 44–54, 2016.
- [32] Y. Sheng, J. Rong and W. Xiang, “Simulation of the users’ email behavior based on BP-BDI model,” in *2015 Int. Conf. on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Xi’an, China, pp. 16–22, 2015.
- [33] S. Y. Diallo, R. Gore, J. J. Padilla, H. Kavak and C. J. Lynch, “Towards a world wide web of simulation,” *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 14, no. 2, pp. 159–170, 2017.
- [34] Z. Yang, A. X. Cui and T. Zhou, “Impact of heterogeneous human activities on epidemic spreading,” *Physica A: Statistical Mechanics and Its Applications*, vol. 390, no. 23–24, pp. 4543–4548, 2011.