Сучасний стан наукових досліджень та технологій в промисловості. 2021. № 3 (17)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

UDC 004.9
DOI: https://doi.org/10.30837/ITSSI.2021.17.023

YE. MELESHKO, M. YAKYMENKO, V. BOSKO

A METHOD OF COMPUTER SIMULATION MODELING OF USER AND BOT BEHAVIOR IN A RECOMMENDATION SYSTEM USING THE GRAPH DATABASE NEO4J

The **subject** matter of the article is the process of computer simulation modeling of complex networks. The **goal** is to develop a method of computer simulation modeling of ordinary user and bot behavior in a recommendation system based on the theory of complex networks to test the accuracy and robustness of various algorithms for generating recommendations. The tasks to be solved are: to develop a computer simulation model of user and bot behavior in a recommendation system with the ability to generate datasets for testing recommendation generation algorithms. The **methods** used are: graph theory, theory of complex networks, statistics theory, probability theory, methods of object-oriented programming and methods of working with graph databases. **Results.** A method of computer simulation modeling of users and objects in a recommender system was proposed, which consists of generating the structure of the social graph of a recommender system and simulating user and bot behavior in it. A series of experiments to test the performance of the developed computer simulation model was carried out. During the experiments, working and testing datasets were generated. Based on the working datasets, the preferences of users by the method of collaborative filtering were predicted. Based on testing datasets, the accuracy of prediction predictions was checked. The results of the experiments showed that the jitter of the investigated values of the Precision, Recall and RMSE of prediction predictions in most practical cases confidently fits within the allowable fluctuation limits, and therefore the users' behavior in computer simulation model was not random and it real users' behavior with certain preferences was simulated. This confirms the reliability of the developed computer simulation model of a recommendation system. **Conclusions.** A method of computer simulation modeling of user and bot behavior in a recommendation system, which allows generating datasets for testing the algorithms for generating recommendations, was proposed. The developed method makes it possible to simulate the behavior of both ordinary users and bots, which makes it possible to create datasets for testing the robustness of recommender systems to information attacks, as well as for testing the effectiveness of methods for detecting and neutralizing botnets. The structure of relations between users and objects of the recommender system was modeled using the theory of complex networks. Information attacks of bots were modeled on the basis of known models of profile-injection attacks on recommender systems.

Keywords: recommendation systems; computer modeling; simulation modeling; complex networks; social networks; social graph; bot network; profile-injection attacks; websites; databases.

Introduction

To conduct research in the field of recommendation systems, it is necessary to have a set of data on the actions of users and objects of a system to create and test the quality of recommendation lists. Such data sets can be obtained in the following ways: 1) have access to a database of a real web resource or application; 2) use one of the existing open datasets; 3) create an application for automatic collection of open data from web resources; 4) create a software model of the recommendation system to generate data about its users and objects.

The easiest way to do this is to use an open data set. At the same time, the most complete information for creating and testing the quality of recommendation lists is available to administrators and website owners. They also have the ability to track users' reactions to generated recommendations in real time. But the vast majority of researchers do not have access to such data. In this case, if the information from open datasets is insufficient, there are two ways - par-sing open data from websites or create a software simulation model of the recommendation system based on known information about a system (website or application).

The collection of open data from web resources can be done in two ways - either using API-functions [1, 2], or by developing and using the own web crawler [2, 3]. API functions are easy to use, but not all sites provide the opportunity to gather the necessary information thanks to API functions - such functions may be completely absent or have truncated functionality and a number of restrictions and conditions for use. Also, the interface of API functions changes from time to time, and their new versions may contain fewer features [4].

By using your own web crawler, you can collect any publicly available information from the website. But writing a web crawler is a much more difficult task, you need to understand the html-layout of someone else's website, develop your own functions for collecting data from html-code, make changes or completely rewrite the web crawler every time on the website html-layout changes. Also, large websites block automatic data collection if you notice it. One of the promising ways to parsing websites is to use the library to automate the actions of the web browser and its testing Selenium (for Java, C #, Ruby, Python, Javascript) [5]. This library creates time delays between its actions typical of human behavior, which bypasses the protection of websites from automatic data collection, but it also makes its work rather slow and unsuitable for collecting large amounts of data.

The use of open datasets, as well as the collection of data using API-functions or web-crawler from websites do not allow to study the reaction of users to the proposed recommendations and the impact of botnets.

When developing methods to increase the resilience of the recommendation system to information attacks, there is a problem of lack of open data sets to test the quality of their work. There is no bot profile information in the available datasets. Also, such information cannot be collected by parsing open data from websites. And even

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2021. No. 3 (17)

system administrators who have access to all the information and can record data on bot attacks are not able to collect such data when it is necessary to test certain algorithms in the right quantity and quality in a given period of time. Therefore, there is a need to create a software simulation model of users and objects of the recommendation system to generate data sets that will also contain bot profiles.

Based on the research conducted in this paper, it was decided to create a software simulation model of user and bot behavior in such systems for the analysis of recommendation systems.

A referral system is a network of connections between users and web resource objects. It can be considered as a kind of social graph. A social graph is a graph whose nodes are social objects (for example, users, user communities, content objects, etc.) and whose edges are social connections between them. It is convenient to present the data of the recommendation system in the form of a graph, where users and objects are the vertices of the graph, and user actions (views, evaluations, etc.) and results of the recommendation system (recommendations, similarities, etc.) are edges.

A review of publications on various methods of modeling social graphs and networks was conducted.

There are several approaches to modeling social networks and social graphs, in particular, the following are most often used [6-20]: models of random graphs [6-11], game-theoretic models [7, 11-14], models of complex networks [7 , 15-20] and so on.

It was decided to use the theory of complex (complex) networks to model the relationships between users and objects of the recommendation system, because it allows, compared to other approaches, to model more properties of social graphs [7, 15-25]. A study of the properties of complex networks and methods of their modeling was conducted. Complex networks are stochastic networks with non-trivial topology, in particular, they differ from classical stochastic networks by the presence of a small number of nodes with a large number of connections (such vertices are called hubs) [7, 15-20].

The components of networks that reflect social connections have the following main properties, which are indicated in the studied works by approximately the same list [15-24]: scalelessness, distribution of degrees of nodes (number of connections in nodes) by power distribution, small network diameter, high clustering coefficient and high transitivity coefficient, giant connected component (more than 80% of nodes are interconnected), hierarchical connections are present, complex cluster formations are present (clicks, clans, etc.), assortivity (occurrence connections between vertices that are somewhat similar to each other).

The main known models of generation of stochastic and complex networks: Barabasi-Albert model [15, 16, 24, 26], Erdős-Renyi model [24, 26, 27], Bollobas-Riordan model [26, 28, 29].

The aim of this work is to build a software simulation model of user and bot behavior in the recommendation system of a social network or content-oriented website based on the theory of complex networks and using the Neo4j graph database to study and test methods of creating lists of recommendations.

The proposed method of software simulation of users and objects of the recommendation system consists of two parts:

- Generation of the structure of the social graph of the recommendation system, which has already been proposed in [30].
- And the simulation of the behavior of users and bots of the recommendation system, which will be considered in this paper.

The main material

In this paper, a software simulation model of the behavior of users of the recommendation system was created to test the methods of generating lists of recommendations, in particular, in order to determine and compare the accuracy and stability of different methods.

To model the structure of relations in the social graph of the recommendation system, it was decided to take as a basis, but modify taking into account the specifics of the problem, the principles on which the Barabashi-Albert model is based (namely, "growth" and "desirable accession"). easy to implement and allows you to create a stochastic graph with the properties of social networks. The structure of the social graph was modeled using the graph database Neo4j [31] and the Python programming language. The stages of the proposed method of generating the structure of the social graph of the recommendation system are considered in [30]. The developed model of the social graph of the recommendation network was tested to check the similarity of the properties of the structure of its connections to the structure of real social graphs. The average values of the parameters generated in the proposed software simulation model of social graphs of the recommendation system, which were calculated using Gephi: the average degree of nodes: 5.7, network diameter: 4.0, graph density: 0.22, average clustering factor: 0.61, average path length: 1.98 , which corresponds to the parameters of real social networks.

The next step was to model the behavior of users and bots of the recommendation system. It was decided to develop the model in such a way as to obtain a data set similar to MovieLens datasets [32] in structure and statistical properties.

Stages of the proposed method of software simulation of ordinary users and bots in the recommendation system:

Stage 1. Initialize system parameters, including the choice of the number of users and objects, the number of possible properties in them, the percentage of active users, the percentage of popular objects, the number of time iterations after which the model will complete its work, and so on. If it is necessary to simulate an information attack, the type of attack, the number of bots and the number of attack targets are selected. A set of possible properties for system elements implemented in the model

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2021. № 3 (17) ISSN 2524-2296 (online)

by hidden factors is created, and element cluster templates are generated based on these properties.

Stage 2. Creating a "Grain" of the social graph of the recommendation system - the initial number of users and objects is added to the graph, some number of objects are evaluated. Initial number of users, initial number of objects, and graph density are customizable parameters. Each new user and system object is assigned an offset value, a specific cluster, and a corresponding vector of hidden factor values, which determines the degree to which it belongs to each of the possible properties for the system elements. Estimates in the system are set based on the degree of coincidence of the hidden factors of the user and the object, as well as indicators of their displacements. The probability of viewing an object inside the Grain depends on the specified density of the graph. The probability of rating the viewed object for authentic users depends on the hidden factors of the user (determining his preferences), hidden factors of the object (determining its properties), user offset (which indicates a characteristic systemic understatement of the user or overestimation of estimates), the displacement of the object (which indicates the quality of the object, which always causes the receipt of estimates higher / lower than similar properties of objects), as well as random displacement (which occurs with a probability of 0-3 % and is a customizable parameter). In "Grain" bot profiles are absent; they begin to join the network at the 3rd stage.

Stage 3. Each time iteration of the model, a certain number of users and objects are added to the graph. This number is determined randomly and lies in the range from 0 to N, where N is less than the total number of system elements of the corresponding type. Also, at each iteration of the model time, a number of pairs of users and objects are selected for evaluation. The probability of viewing an object can be determined based on the principle of "desired connection" (1) from the Barabashi-Albert model or its modifications, for example (2):

$$P_i = \frac{k_i}{\sum_j k_j}, \quad (1)$$

$$P_i = \frac{q_{1i} + q_{2i} + q_{3i}}{\sum_j (q_{1j} + q_{2j} + q_{3j})}, \quad (2)$$

where q_{1i} – the number of views in the i-th object, q_{2i} – the number of estimates in the i-th object, q_{3i} – the number of comments in the i-th object.

The probability of rating a viewed object for authentic users depends on hidden user factors, hidden object factors, user offset, object offset, and random offset. The probability of viewing an object and rating it for bots is determined by the attack model used to create them.

Stage 4. Stopping the simulation model, saving the generated data set to a file for further use in methods of forming and testing lists of recommendations.

The software simulation model consists of the following elements: "System user model", "System object model" and "Information process model in the recommendation system".

The system user model has the following parameters:

- Id - user identification number containing a digit;
- Active - takes True - if the user is active (puts more ratings than others) and False – if the activity level is normal;
- Bot - takes True – if the user is a bot (profile is involved in an attack on the referral system) and False - otherwise;
- Type - takes the value "Normal", if the user is not a bot, otherwise it contains the name of the attack applied to the re-recommendation system;
- Character - contains the type of character of the user in relation to the style of grading: "Normal", "Often gives good grades", "Gives only good grades";
- Registration time – contains the time of user registration in the system;
- Time of last activity – contains the time of the last activity of the user;
- Activity stop time - the time when the user stopped using the system;
- Hidden factors - a list of random variables that take values from -1 to 1, length K, which simulates the influence of certain characteristics of the object on the user's judgment about it;
- User offset - a random variable in the range from -1.0 to 1.0, which simulates the user's tendency to underestimate or overestimate the values of objects;
- Time of last preference change - used when necessary to simulate changes in user preferences over time, contains the time of the last overwrite of the list of hidden user factors.

The system object model has the following parameters:

- Id - identification number of the object, containing a digit;
- Popular - takes True - if the object is popular (gets more ratings than others) and False - if the popularity level is normal;
- Type for bots - takes the value "Target" - if the bots need to change the rating of the object in a certain direction and False - otherwise;
- Attack type - takes the value "None" if the object is not targeted, or indicates the type of attack "To decrease the rating", "To increase the rating";
- Add time - contains the time to add the object to the database of the recommendation system;
- Hidden factors - a list of random variables that take values from -1 to 1, length K, which simulates the expression of certain characteristics in the object, which affect the level of interest in the user;
- Object offset - a random variable in the range from -1.0 to 1.0, which simulates the overall quality of the object, which affects user ratings and leads to more low ratings due to low quality or more often high scores due to high quality. project.

The parameters of users and system objects are set during the creation and initialization of the corresponding instances of objects, and some of them may change during the operation of the program.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2021. No. 3 (17)

The model of information processes in the system has the following parameters:

- Rating matrix – contains a matrix of adjacency of the graph, in which the vertices are users and objects, and the evaluation edges are set by users to objects.

- Time matrix – contains a matrix of adjacency of the graph, in which the vertices are users and objects, and the edges are the time of evaluation of user-supplied objects.

- Set of possible estimates – contains a set of estimates that users can expose to objects, in the developed model the set of estimates is represented by the following list [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0], which means the ability to rate in the form of the number of stars, a maximum of 5 stars, you can choose half a star.

- User list - contains a list of system users, instances of the User class.

- Object list – contains a list of system objects, instances of the Object class.

- Start time of the model – contains time in Unix time stamp format.

- Current model run time – contains the time in Unix time stamp format, which indicates the time of the last action in the system.

- List of bots – contains a list of bots in the developed model.

- Target list – contains a list of bot attack targets in the developed model.

- Number of clusters - contains a specified number of clusters, which can be divided into elements of the system based on data about their hidden factors.

- Cluster templates – contains a list of templates for modeling hidden factors of users / objects. In total, 19 random templates were created in the model to generate elements that may belong to 19 different clusters. Also in the model you can choose the option of modeling a certain percentage of elements that do not belong to the generated clusters.

The following functions were implemented to model the behavior of users and objects and information processes of the recommendation system:

- Model launch – starts the process of modeling users and objects of the recommendation system.

- Save model data to a file – saves all data and parameters of the developed software simulation model of users, objects and processes of the recommendation system to a file.

- Generating system elements – creating a given number of users and system objects and assigning them parameters.

- Generation of elements of clusters of elements – allows to generate templates of the hidden factors of elements for creation in the subsequent set of the elements relating to certain clusters.

- Generation of hidden factors of elements on the basis of cluster templates – at the input receives a cluster template, at the output provides a list of hidden factors that randomly differ from the data in the template, so as not to go beyond the cluster, but have their own unique values of factors.

- Generation of hidden factors of elements outside of clusters - generation of the list of hidden factors of an element randomly without use of templates.

- Grain generation of a social graph – creates an initial social graph of the recommendation system based on a given initial number of users, objects and graph density.

- Iterative model construction – allows you to simulate the change of time.

- One iteration of the model – within this function are called all the functions that implement the behavior of users and the work of the recommendation system at the current time.

- Generation of the current operating time of the model - the generation of the time interval between two events in the system – a random variable that lies in a given range.

- Adding a user to the system – modeling the process of user registration in the recommendation system, the user gets the time of registration and the opportunity to view objects and give them ratings.

- Adding an object to the system – modeling the process of adding an object to the database of the recommendation system, the object gets the time of adding to the database and the ability to receive views and ratings.

- Determining the probability of viewing an object – based on the principle of "Preferred connection" is determined by the probability of viewing a particular object by a particular user.

- Determining the probability of occurrence of the assessment - based on the principle of "Desired accession" is determined by the probability of assessment of a particular object by a particular user.

- Generation of an estimate based on the hidden factors of the respective object and user. The score for a user-object pair is determined by the following formulas:

$$d_{u,m} = \frac{\sum_{i=0}^{n} \left| f_{u,i} - f_{m,i} \right|}{n}, \quad (3)$$

$$r_{u,m} = \Psi\left(5d_{u,m} + b_u + b_m\right), \quad (4)$$

where $d_{u,m}$ – the distance between user u and object m in the multidimensional space of hidden factors, can take values from 0 to 1; n – the number of hidden factors in the system; $f_{u,i}$ – i-th hidden user u factor; $f_{m,i}$ – the i-th hidden factor of the object m; b_u – user offset in object evaluation (content level); b_m – object offset in obtaining assessments (content quality level); $\psi()$ – a function that converts the resulting fractional number into a discrete number from a set of estimates [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0], for example, if the number is in the range from (4.000 - k) to (4.500 - k), where k is a small number (in the model was taken k = 0.05), it is converted to a score of 4.0.

- The decision to make an assessment and its adjustment. After the user "viewed" the object, and based

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2021. № 3 (17) ISSN 2524-2296 (online)

on hidden factors, it was determined which rating for the object corresponds to his preferences, this function is used, which determines the fact that the user will decide to put the viewed object evaluation. Also in this function, a slight adjustment of the assessment based on random factors is possible. With the help of this function, the probability of occurrence of estimates can be approximated to the frequencies of occurrence of estimates in the data set MovieLens datasets.

The model also developed bot rating generators for different types of attacks. Three known models of attacks on the recommendation system by injection of profiles [33-37] were used to generate bot estimates: random, medium and popular attacks.

- Random attack estimate generation – creates estimates for bot object pairs, where the bot randomly attacks the referral network. Generating a score for a random attack to increase the rating is as follows:

$$r_{u,m} = \begin{cases} \text{randomPattern}(), if\ an\ object\ is\ "random" \\ 5.0, if\ an\ object\ is\ "target" \end{cases}, \quad (5)$$

where randomPattern() is a function that generates random estimates with given values of the probability of occurrence.

- Estimation for average attack – creates estimates for bot object pairs, where the bot performs an average attack on the referral network. The generation of the score for the average attack on the rating upgrade is as follows:

$$r_{u,m} = \begin{cases} average\ Pattern(), if\ an\ object\ is\ "random" \\ 5.0, if\ an\ object\ is\ "target" \end{cases}, (6)$$

where *average* Pattern() is a function that generates an average estimate for a randomly selected object.

- Generate a rating for a popular attack – creates ratings for bot object pairs, where the bot performs a popular attack on the referral network. The rating for a popular rating upgrade attack is generated as follows:

$$r_{u,m} = \begin{cases} popular\ Pattern(), if\ an\ object\ is\ "random" \\ 5.0, if\ an\ object\ is\ "target" \end{cases}, (7)$$

where *popular* Pattern() – function that generates for a randomly selected object from a set of popular objects its average rating.

- Determining the probability of changing preferences - used when necessary to simulate changes in preferences of system users over time, determines the probability that at the current time the specified user will change their preferences, if the time is known when he last changed preferences.

- Change user preferences – replaces the hidden factors of the specified user.

Therefore, based on the proposed method of modeling the structure of the links between the elements and the behavior of users of the recommendation system of the content-oriented website, a software simulation model of the recommendation system was developed.

The software simulation model of user and bot behavior in the recommendation system was developed in the Python programming language using the principles of object-oriented programming. All data of the software simulation model were written to the graph database Neo4j, and also at the end of work of model were stored in a file of own format.

The Neo4j database has the following structural elements that allow you to build a social graph:

- node - node of the graph, the number of nodes is limited to 235, ie ~ 34 billion.

- node label - node label, labels can be used to filter data.

- relation - an edge, a connection between two nodes. The number of edges is also limited to 235.

- relation identifier – connection type (edges), maximum number of connection types 32767.

- properties - properties of nodes and edges, allow to add information fields to them.

- node ID - a unique node identifier, analogous to the key field in relational databases.

The advantages of the Neo4j database are a flexible data model, real-time analysis, easy data retrieval and filtering capabilities, scalability and reliability, the presence of a visual interface, as well as the existence of the Neo4j Aura application for use as a cloud service.

Thus, the Neo4j database provides all the necessary tools to create a software simulation model of the social graph, which can be further used to model and study the behavior of users and bots in referral systems.

The social graph of the recommendation system in the software simulation model had the following format: vertices - users and objects, edges – connections such as "rated", "like", "recommended", etc. Both vertices and edges contain sets of parameters that correspond to the available information about them. For example, the "rated" edge contains the evaluation value and timestamp, and the user type vertex contains all the data fields that correspond to the parameters of the user model in the system.

A series of experiments was performed to verify the operability of the developed software simulation model. Many datasets were generated during the experiments. Each data set was divided into a working and a test sample. Based on the test sample, the accuracy of prediction of preferences was checked. The results of the experiments showed that the jitter of the studied values of accuracy, completeness and RMSE prediction of preferences in most practical cases confidently fits into the allowable limits of fluctuations, and therefore user behavior in the software model was not random, but simulated the behavior of real users with certain preferences. This confirms the reliability of the developed software simulation model of the recommendation system.

The results of the experiments are shown in fig. 1-3. During the experiments, the values of the following indicators were calculated: $P_{prec}^{(i)}$ – precision prediction of preferences, $P_{rec}^{(i)}$ – recall completeness and RMSE preference recognition error.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

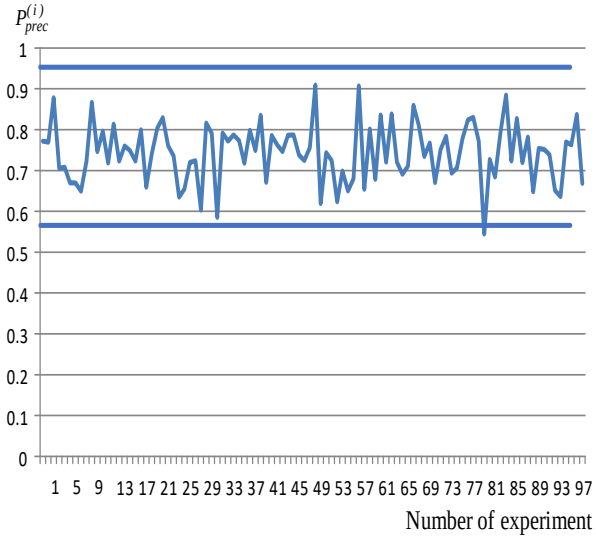

Fig. 1. Graph of changes in random characteristics of the frequency of correct prediction of preferences $P_{prec}^{(i)}$

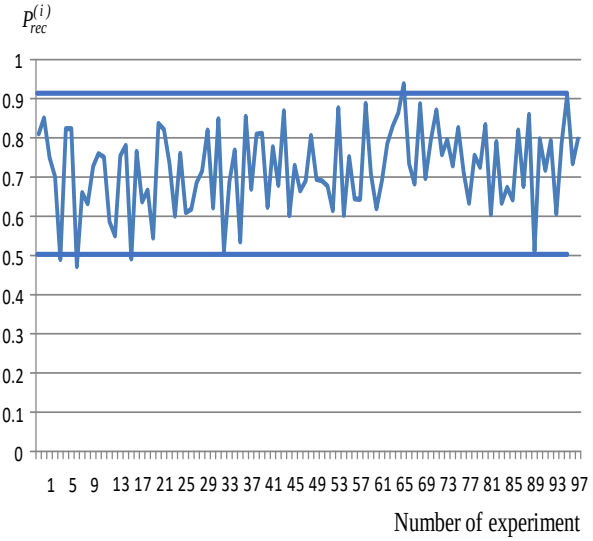

Fig. 2. Graph of changes in random characteristics of the frequency of correct prediction of preferences $P_{rec}^{(i)}$

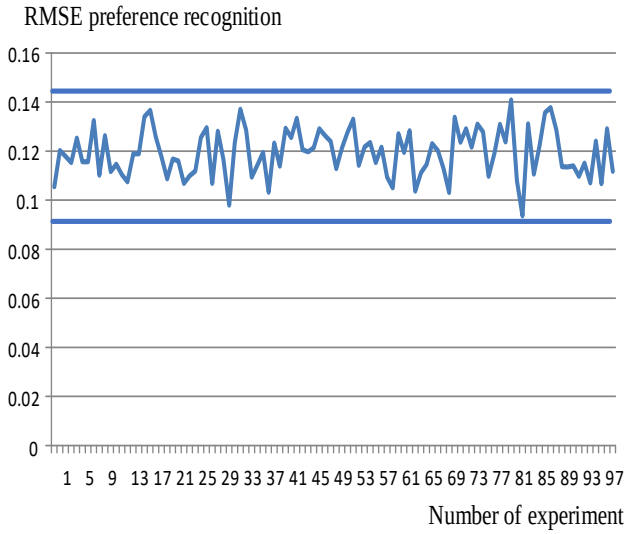

Fig. 3. Graph of changes in random frequency characteristics of RMSE preference recognition

Fig. 1-3 shows graphs of changes in random frequency characteristics $P_{prec}^{(i)}$, $P_{rec}^{(i)}$ and RMSE preference recognition, respectively. Also, these figures show the allowable limits of oscillation of these random variables, obtained as a result of the simulation.

As can be seen from the figures, jitter studied the random characteristics $P_{prec}^{(i)}$, $P_{rec}^{(i)}$ and RMSE preference recognition in most practical cases confidently fits within acceptable limits. This confirms the reliability of the developed software simulation model of the recommendation system.

Conclusions

This paper proposes a method of software simulation of users and objects of the recommendation system for a social network or for a content-oriented website, which allows you to generate data sets for testing algorithms for generating recommendations. The developed method allows modeling the behavior of both regular users and bots, which makes it possible to create data sets to test the resilience of recommendation systems to information attacks, as well as the effectiveness of methods for detecting and neutralizing botnets. The structure of relationships between users and objects of the recommendation system was modeled using the theory of complex networks. Information attacks of bots were modeled on the basis of known models of attacks by injection of profiles on recommendation systems.

Further research in this area may be aimed at building software simulations of social graphs based on the theory of complex networks and using the graph database Neo4j to study not only the recommendation systems of social networks and content-oriented websites,

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2021. № 3 (17) ISSN 2524-2296 (online)

but also other applications and processes. occurring in social media, in particular, the processes of dissemination of information and information influences, applications with reputational systems, etc.

References

1. Segaran, T. (2011), *Programming Collective Intelligence*, Translated from English, Saint Petersburg : Symbol-Plus, 368 p.
2. Hogan, B. (2012), "Analysis of social networks on the Internet", PostNauka website about modern fundamental science and scientists, available at : https://postnauka.ru/longreads/20259
3. Meleshko, Ye. V., Okhotnyi, S. M., Bosko, V. V. (2019), "Development of software for collecting and analyzing data from social networks", *Collection of abstracts of the IX International scientific-practical conference "Integrated quality assurance of technological processes and systems"*, Vol. 2, Chernihiv, May 14-16, 2019, Chernihiv : ChNTU, P. 225–226.
4. Meleshko, Ye. V., Semenov, S. H., Khokh, V. D. (2018), "Research of methods of construction of recommendation systems on the Internet", *Collection of scientific works "Control, navigation and communication systems"*, Issue 1 (47), Poltava : PNTU, P. 131–136.
5. "The Selenium Browser Automation Project" (2021), available at : https://www.selenium.dev/documentation/
6. Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., Upfal, E. (2000), "Stochastic models for the web graph", *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00). IEEE Computer Society*, Redondo Beach, CA, USA, P. 57–65. DOI: https://doi.org/10.1109/SFCS.2000.892065
7. Traag, V. A. (2014), "Algorithms and Dynamical Models for Communities and Reputation in Social Networks", *Springer International Publishing*, P. 229. DOI: https://doi.org/10.1007/978-3-319-06391-1
8. Li, Q. L. (2016), "Nonlinear Markov processes in big networks", *Special Matrices*, Vol. 4 (1), P. 202–217.
9. Rajgorodskij, A. M. (2012), "Mathematical models of the Internet", *Jornal "Kvant"*, No. 4, P. 12–16, available at : https://elementy.ru/nauchno-populyarnaya_biblioteka/431792
10. Rajgorodskij, A. M. (2010), "Random graph models and their applications", *Proceedings Moscow Institute of Physics and Technology*, Vol. 2, No. 4, P. 130–140.
11. Suslova, V. A., Gorodov, A. A. (2015), "Methods for modeling social networks", *Reshetnev readings*, No. 2 (19), P. 133–134, available at : http://cyberleninka.ru/article/n/metody-modelirovaniya-sotsialnyh-setey
12. Gubanov, D. A., Novikov, D. A., Chhartishvili, A. G. (2009), "Influence models on social networks", *Management of big systems*, No. 27, P. 205–281, available at : http://cyberleninka.ru/article/n/modeli-vliyaniya-v-sotsialnyh-setyah
13. Gubanov, D. A., Novikov, D. A., Chhartishvili, A. G. (2010), *Social networks: models of information influence, control and confrontation*, Physics and Mathematics Literature Publishing House, 228 p.
14. Melikov, S., Musatov, D., Savvateev, A. (2013), "Social networks' modeling", available at : https://kpfu.ru/docs/F117464271/MMS_socnet_cities.pdf
15. Barabási, A.-L. (2018), *Network science*, Cambridge University Press, 475 p., available at : http://networksciencebook.com/
16. Albert, R., Barabási, A.-L. (2002), "Statistical mechanics of complex networks", *Reviews of Modern Physics*, Vol. 74, P. 47–97. DOI: 10.1103/RevModPhys.74.47
17. Evin, I. A. (2010), "Introduction to the theory of complex networks", *Computer Research and Modeling*, Vol. 2, No. 2, P. 121–141.
18. Lande, D. V., Snarskij, A. A., Bezsudnov, I. V. (2009), *Internetics: Navigation in complex networks: models and algorithms*, Moscow : Librokom (Editorial URSS), 264 p., available at : http://dwl.kiev.ua/art/internetica/
19. Pasichnyk, V. V., Ivanushchak, N. M. (2010), "Research and modeling of complex networks", *Eastern European Journal of Advanced Technology*, Vol. 2, No. 3 (44), P. 43–48.
20. Snarskij, A. A., Lande, D. V. (2015), *Modeling complex networks: a tutorial*, Kyiv : Engineering, 212 p., available at : http://dwl.kiev.ua/art/mss/
21. Barabási, L.-A., Albert, R., Jeong, H. (2000), "Scale-free characteristics of random networks: the topology of the world-wide web", *Physica*, A281, P. 69–77.
22. Haidai, B., Artiukh, R., Malyeyeva, O. (2018), "Analysis and modelling the preferences of social networks users", *Innovative Technologies and Scientific Solutions for Industries*, No. 1 (3), P. 5–12. DOI: https://doi.org/10.30837/2522-9818.2018.3.005
23. Watts, D. J., Strogatz, S. H. (1998), "Collective dynamics of "small-world" networks", *Nature*, Vol. 393 (6684), P. 440–442, available at : https://www.nature.com/articles/30918
24. Gusarova, N. F. (2016), *Analysis of social networks. Basic concepts and metrics*, Saint Petersburg : ITMO University, 67 p.
25. Barabási, A.-L., Albert, R. (1999), "Emergence of scaling in random networks", *Science*, Vol. 286, No. 5439, P. 509–512. DOI: https://doi.org/10.1126/science.286.5439.509
26. Bernovskij, M. M., Kuzjurin, N. N. (2012), "Random graphs, models and generators of scaleless graphs", *Proceedings of the Institute for System Programming of the Russian Academy of Sciences*, Vol. 22, P. 419–432, available at : https://cyberleninka.ru/article/n/sluchaynye-grafy-modeli-i-generatory-bezmasshtabnyh-grafov
27. Erdös, P., Rényi, A. (1960), "On the evolution of random graphs", *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, Vol. 5, P. 17–61.
28. Bollobás, B., Borgs, C., Chayes, T., Riordan, O. M. (2003), "Directed scale-free graphs", *Proceeding SODA '03 Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, P. 132–139.
29. Bollobás, B., Riordan, O. (2003), "Mathematical results on scale-free random graphs", *Handbook of graphs and networks*, Weinheim : Wiley-VCH, P. 1–34.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2021. No. 3 (17)

30. Meleshko, Ye. (2019), "Computer model of virtual social network with recommendation system", *Innovative technologies and scientific solutions for industries*, No. 2 (8), P. 80–85. DOI: https://doi.org/10.30837/2522-9818.2019.8.080
31. "Neo4j Documentation" (2021), Official website of the graph database Neo4j, available at : https://neo4j.com/docs/
32. Harper, F. M., Konstan, J. A. (2015), "The MovieLens Datasets: History and Context", *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 19 p. DOI: https://doi.org/10.1145/2827872
33. Chirita, P. A., Nejdl, W., Zamfir, C. (2005), "Preventing shilling attacks in online recommender systems", *In Proceedings of the ACM Workshop on Web Information and Data Management*, P. 67–74.
34. Gunes, I., Kaleli, C., Bilge, A., Polat, H. (2014), "Shilling attacks against recommender systems: a comprehensive survey", *Artificial Intelligence Review*, Vol. 42, P. 767–799. DOI: https://doi.org/10.1007/s10462-012-9364-9
35. Mobasher, B., Burke, R., Bhaumik, R.,Williams, C. (2007), "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness", *ACM Transactions on Internet Technology*, Vol. 7 (4), P. 41. DOI: https://doi.org/10.1145/1278366.1278372
36. O'Mahony, M. P., Hurley, N. J., Silvestre, G. C. M. (2002), "Promoting recommendations: An attack on collaborative filtering" *DEXA, Lecture Notes in Computer Science*, Vol. 2453, P. 494–503.
37. Ricci, F., Rokach, L., Shapira, B., Kantor, P. B. (Editors) (2011), *Recommender Systems Handbook*, Boston : Springer, 842 p. DOI: https://doi.org/10.1007/978-0-387-85820-3

Received 18.08.2021

Відомості про авторів / Сведения об авторах / About the Authors

Мелешко Єлизавета Владиславівна – доктор технічних наук, доцент, Центральноукраїнський національний технічний університет, доцент кафедри кібербезпеки та програмного забезпечення, Кропивницький, Україна; email: elismeleshko@gmail.com; ORCID: https://orcid.org/0000-0001-8791-0063.

Мелешко Елизавета Владиславовна – доктор технических наук, доцент, Центральноукраинский национальный технический университет, доцент кафедры кибербезопасности и программного обеспечения, Кропивницкий, Украина.

Meleshko Yelyzaveta – Doctor of Sciences (Engineering), Associate Professor, Central Ukrainian National Technical University, Associate Professor of the Department of Cybersecurity and Software, Kropyvnytskyi, Ukraine.

Якименко Микола Сергійович – кандидат фізико-математичних наук, доцент, Центральноукраїнський національний технічний університет, завідувач кафедри вищої математики та фізики, Кропивницький, Україна; email: m.yakymenko@gmail.com; ORCID: https://orcid.org/0000-0003-3290-6088.

Якименко Николай Сергеевич – кандидат физико-математических наук, доцент, Центральноукраинский национальный технический университет, заведующий кафедрой высшей математики и физики, Кропивницкий, Украина.

Yakymenko Mykola – PhD (Physical and Mathematical Sciences), Associate Professor, Central Ukrainian National Technical University, Head of the Department of Higher Mathematics and Physics, Kropyvnytskyi, Ukraine.

Босько Віктор Васильович – кандидат технічних наук, доцент, Центральноукраїнський національний технічний університет, доцент кафедри кібербезпеки та програмного забезпечення, Кропивницький, Україна; email: victorvv2@ukr.net; ORCID: https://orcid.org/0000-0002-4933-9676.

Босько Виктор Васильевич – кандидат технических наук, доцент, Центральноукраинский национальный технический университет, доцент кафедры кибербезопасности и программного обеспечения, Кропивницкий, Украина.

Bosko Viktor – PhD (Engineering Sciences), Associate Professor, Central Ukrainian National Technical University, Associate Professor of the Department of Cybersecurity and Software, Kropyvnytskyi, Ukraine.

МЕТОД ПРОГРАМНОГО ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ПОВЕДІНКИ КОРИСТУВАЧІВ ТА БОТІВ У РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ З ВИКОРИСТАННЯМ ГРАФОВОЇ БАЗИ ДАНИХ NEO4J

Об'єктом дослідження є процес програмного імітаційного моделювання складних мереж. **Метою** даної роботи є розробка методу програмного імітаційного моделювання поведінки звичайних користувачів та ботів у рекомендаційній системі на основі теорії складних мереж для можливості тестування точності та стійкості різних алгоритмів формування рекомендацій. **Задача:** розробити програмну імітаційну модель поведінки користувачів та ботів у рекомендаційній системі з можливістю генерації наборів даних для тестування алгоритмів формування рекомендацій. **Методи дослідження:** теорія графів, теорія складних мереж, теорія статистики, теорія ймовірностей, методи об'єктно-орієнтованого програмування та методи роботи з графовими базами даних. **Результати.** Запропоновано метод програмного імітаційного моделювання користувачів та об'єктів рекомендаційної системи, що складається з генерації структури соціального графу рекомендаційної системи та симуляції поведінки користувачів і ботів у ній. Було проведено серію експериментів для перевірки працездатності розробленої програмної імітаційної моделі. У ході експериментів було згенеровано множину робочих та тестових наборів даних. На основі робочої вибірки здійснювалося прогнозування вподобань користувачів методом колаборативної фільтрації. На основі тестової вибірки перевірялася точність прогнозування вподобань. Результати проведених експериментів показали, що джиттер досліджуваних значень точності, повноти та RMSE прогнозування вподобань у більшості практичних випадків впевнено вкладається в допустимі межі коливань, а отже поведінка користувачів у програмній моделі не була

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2021. № 3 (17) *ISSN 2524-2296 (online)*

випадковою, а імітувала поведінку реальних користувачів з певними вподобаннями. Це підтверджує достовірність розробленої програмної імітаційної моделі рекомендаційної системи. **Висновки.** Запропоновано метод програмного імітаційного моделювання користувачів у рекомендаційній системі, що дозволяє генерувати набори даних для тестування алгоритмів формування рекомендацій. Розроблений метод дозволяє моделювати поведінку як звичайних користувачів, так і ботів, що дає можливість створювати набори даних для тестування стійкості рекомендаційних систем до інформаційних атак, а також для тестування ефективності методів виявлення та нейтралізації бот-мереж. Структура зв'язків між користувачами та об'єктами рекомендаційної системи моделювалася за допомогою теорії складних мереж. Інформаційні атаки ботів моделювалися на основі відомих моделей атак ін'єкцією профілів на рекомендаційні системи.

Ключові слова: рекомендаційні системи; програмне моделювання; імітаційне моделювання; складні мережі; соціальні мережі; соціальний граф; мережа ботів; атаки ін'єкцією профілів; веб-сайти; бази даних.

МЕТОД ПРОГРАММНОГО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ И БОТОВ В РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЕ С ИСПОЛЬЗОВАНИЕМ ГРАФОВОЙ БАЗЫ ДАННЫХ NEO4J

Объектом исследования является процесс программного имитационного моделирования сложных сетей. **Целью** данной работы является разработка метода программного имитационного моделирования поведения обычных пользователей и ботов в рекомендательной системе на основе теории сложных сетей для возможности тестирования точности и стойкости различных алгоритмов формирования рекомендаций. **Задача:** разработать программную имитационную модель поведения пользователей и ботов в рекомендательной системе с возможностью генерации наборов данных для тестирования алгоритмов формирования рекомендаций. **Методы** исследования: теория графов, теория сложных сетей, теория статистики, теория вероятностей, методы объектно-ориентированного программирования и методы работы с графовыми базами данных. **Результаты.** Предложен метод программного имитационного моделирования пользователей и объектов рекомендательной системы, состоящей из генерации структуры социального графа рекомендательной системы и симуляции поведения пользователей и ботов в ней. Была проведена серия экспериментов для проверки работоспособности разработанной программной имитационной модели. В ходе экспериментов было сгенерировано множество рабочих и тестовых наборов данных. На основе рабочей выборки осуществлялось прогнозирование предпочтений пользователей методом коллаборативной фильтрации. На основе тестовой выборки проверялась точность прогнозирования предпочтений. Результаты проведенных экспериментов показали, что джиттер исследуемых значений точности, полноты и RMSE прогнозирования предпочтений в большинстве практических случаев уверенно укладывается в допустимые пределы колебаний, а следовательно поведение пользователей в программной модели не было случайным, а имитировало поведение реальных пользователей с определенными предпочтениями. Это подтверждает достоверность разработанной программной имитационной модели рекомендательной системы. **Выводы.** Предложен метод программного имитационного моделирования пользователей в рекомендательной системе, позволяющий генерировать наборы данных для тестирования алгоритмов формирования рекомендаций. Разработанный метод позволяет моделировать поведение как обычных пользователей, так и ботов, что дает возможность создавать наборы данных для тестирования стойкости рекомендательных систем к информационным атакам, а также для тестирования эффективности методов выявления и нейтрализации бот-сетей. Структура связей между пользователями и объектами рекомендательной системы моделировалась с помощью теории сложных сетей. Информационные атаки ботов моделировались на основе известных моделей атак инъекцией профилей на рекомендательные системы.

Ключевые слова: рекомендательные системы; программное моделирование; имитационное моделирование; сложные сети; социальные сети; социальный граф; сеть ботов; атаки инъекцией профилей; веб-сайты; базы данных.

Бібліографічні описи / Bibliographic descriptions

Мелешко Є. В., Якименко М. С., Босько В. В. Метод програмного імітаційного моделювання поведінки користувачів та ботів у рекомендаційній системі з використанням графової бази даних NEO4J. *Сучасний стан наукових досліджень та технологій в промисловості*. 2021. № 3 (17). С. 23–31. DOI: https://doi.org/10.30837/ITSSI.2021.17.023

Meleshko, Ye., Yakymenko, M., Bosko, V. (2021), "A method of computer simulation modeling of user and bot behavior in a recommendation system using the graph database NEO4J", *Innovative Technologies and Scientific Solutions for Industries*, No. 3 (17), P. 23–31. DOI: https://doi.org/10.30837/ITSSI.2021.17.023