Національний технічний університет «Харківський політехнічний інститут»
Міністерство освіти і науки України
Черкаський державний технологічний університет
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

Мелешко Єлизавета Владиславівна

УДК 004.89+004.942

ДИСЕРТАЦІЯ

МЕТОДОЛОГІЯ ЗАБЕЗПЕЧЕННЯ СТІЙКОСТІ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ДО ДЕСТАБІЛІЗУЮЧИХ ФАКТОРІВ У КОМП'ЮТЕРНИХ МЕРЕЖАХ

Спеціальність 05.13.05 – Комп'ютерні системи та компоненти

Подається на здобуття наукового ступеня доктора технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____Є.В. Мелешко

Науковий консультант:
Семенов Сергій Геннадійович,
доктор технічних наук, професор

Черкаси – 2021

АНОТАЦІЯ

Мелешко Є.В. Методологія забезпечення стійкості рекомендаційних систем до дестабілізуючих факторів у комп'ютерних мережах. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.05 «Комп'ютерні системи та компоненти». – Черкаський державний технологічний університет, Черкаси, 2021.

Дисертаційна робота присвячена вирішенню актуальної науково-технічної проблеми підвищення точності пропозицій рекомендаційних систем в умовах дестабілізуючих факторів у комп'ютерних мережах на основі розробки моделей та методів синтезу підсистеми забезпечення стійкості.

Проведено дослідження та порівняльний аналіз моделей та методів синтезу рекомендаційних систем віртуальних соціальних мереж та контент-орієнтованих веб-сайтів, який показав, що переважна більшість існуючих моделей і методів вразливі до дії внутрішніх та зовнішніх дестабілізуючих факторів у комп'ютерних мережах. Показано, що забезпечення стійкості рекомендаційних систем до дії дестабілізуючих факторів є важливою умовою для підвищення точності їх роботи.

Розроблено метод визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах з використанням математичного апарату марківських та напівмарківських процесів, що дає можливість встановлення зв'язку між набором щільності розподілу випадкових тривалостей перебування системи у цих станах і функціями опису динаміки ймовірностей станів для визначення ймовірностей перебування конкретної рекомендаційної системи в своїх можливих станах в довільний момент часу. На основі запропонованого методу було розроблено методику отримання аналітичних співвідношень для розрахунку

ймовірностей перебування системи у своїх можливих станах в довільний момент часу.

Розроблено математичну модель стійкої рекомендаційної системи на основі запропонованого методу визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило здійснити оптимізацію загальних витрат на обслуговування системи в умовах внутрішніх дестабілізуючих факторів. На основі розробленої математичної моделі запропоновано спосіб визначення повних витрат підсистеми збору та перерахунку вхідних даних, а також спосіб визначення оптимальної частоти перерахунку вхідних даних, при яких система має мінімальну збитковість.

Удосконалено метод колаборативної фільтрації, який відрізняється від існуючих використанням продукційних правил для визначення подоби користувачів та використанням показників активності користувачів для формування рекомендацій, що дозволило підвищити стійкість системи у випадку недостатньої кількості вхідних даних та під час холодного старту. Розроблено та реалізовано відповідні алгоритми, використання яких дозволило в 1.6 разів підвищити стійкість рекомендаційної системи в порівнянні з відомим методом колаборативної фільтрації на основі моделі сусідства та в 3.2 разів – в порівнянні з відомим методом колаборативної фільтрації на основі факторизації матриць.

Розроблено математичну модель підсистеми інформаційної безпеки стійкої рекомендаційної системи на основі запропонованого методу визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило визначити оптимальну частоту перевірки на наявність інформаційної атаки та профілів ботів. У межах математичної моделі розроблено набір можливих станів, у яких може перебувати рекомендаційна система в умовах інформаційних атак ін’єкцією профілів, визначено можливі переходи між цими станами. Розроблено методику отримання аналітичних співвідношень для розрахунку ймовірностей перебування підсистеми

інформаційної безпеки в своїх можливих станах в довільний момент часу. На основі запропонованої математичної моделі розроблено спосіб визначення повних витрат, що зазнає рекомендаційна система внаслідок моніторингу власної інформаційної безпеки, нейтралізації діяльності бот-мереж та внаслідок інформаційних атак ін'єкцією профілів. Розроблений спосіб дозволяє при відомих витратах на обчислювальні ресурси та відомих збитках при атаках бот-мереж визначати загальні витрати на обслуговування підсистеми безпеки рекомендаційної системи. Також розроблено спосіб визначення оптимальної частоти перевірки рекомендаційної системи на наявність інформаційної атаки та профілів ботів для оптимізації загальних витрат системи.

Розроблено метод імітаційного програмного моделювання користувачів та об'єктів рекомендаційної системи соціальної мережі або веб-ресурсу на основі існуючих і розроблених методів моделювання структури складних мереж та методів моделювання поведінки користувачів, що дозволило генерувати вхідні дані для тестування якості роботи алгоритмів формування рекомендацій. Розроблено спосіб моделювання змін вподобань у часі користувачів рекомендаційної системи, що дозволяє генерувати тестові набори даних, більш схожі за статистичними характеристиками на реальні. Спосіб засновано на математичній моделі нециклічних змін вподобань користувачів у часі з використанням експоненційного закону розподілу та закону радіоактивного розпаду елементів. Розроблено та реалізовано відповідні алгоритми та програмну імітаційну модель користувачів, об'єктів, бот-мереж і інформаційних процесів у рекомендаційній системі. За допомогою розробленої програмної імітаційної моделі сформовано набори даних для тестування роботи методів синтезу рекомендаційних систем та підсистем їх інформаційної безпеки.

Розроблено метод виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів рейтингів об'єктів, що дозволило знизити

кількість витрат на моніторинг безпеки системи за рахунок зняття необхідності пошуку ботів при відсутності ознак атаки. Розроблено та реалізовано відповідні алгоритми, які дозволили виявляти в середньому 76% об’єктів інформаційних атак у рекомендаційній системі. Множину виявлених ймовірних цілей атаки можна використати для подальшого пошуку профілів ботів. Це дозволить при пошуку бот-мереж перевіряти не всі профілі системи, як у відомих методах, а тільки ті, які взаємодіяли з ймовірними цілями атаки.

Розроблено метод виявлення бот-мереж у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів, що дозволило виявляти бот-мережі та розрізняти їх за множинами об’єктів атаки. Також розроблено спосіб ідентифікації окремих профілів ботів на основі нейронних мереж у рекомендаційних системах для уточнення результатів запропонованого методу виявлення бот-мереж на основі графової кластеризації. Розроблено відповідні алгоритми та реалізовано підсистему інформаційної безпеки рекомендаційної системи. Розроблена підсистема інформаційної безпеки дозволяє забезпечити вищу стійкість рекомендаційної системи до зовнішніх дестабілізуючих факторів на відміну від існуючих методів. Вона дозволяє підвищити стійкість системи в середньому у 2.5 рази до випадкової і середньої атаки та в 1.7 разів – до популярної атаки, якщо застосовується для методу колаборативної фільтрації на основі моделі сусідства або для запропонованого гібридного методу колаборативної фільтрації.

Проведена оцінка достовірності та ефективності запропонованих методів і моделей підвищення стійкості рекомендаційних систем.

Ключові слова: рекомендаційні системи, комп’ютерні мережі, проблема холодного старту, колаборативна фільтрація, аналіз даних, кластеризація, моделювання складних мереж, інформаційні атаки, бот-мережі, соціальні мережі, веб-ресурси

SUMMARY

Meleshko Ye. The methodology of ensuring the stability of recommendation systems to destabilizing factors in computer networks. – Qualifying scientific work with manuscript copyright.

Dissertation for the degree of Doctor in Technical Sciences on specialty 05.13.05 «Computer systems and components». – Cherkasy State Technological University, Cherkasy, 2021.

The dissertation is devoted to solution of an actual scientific and practical problem of accuracy increasing for propositions of recommendation systems in the conditions of destabilizing factors in computer networks on the basis of development of models and methods of synthesis of a subsystem of stability ensuring.

Research and comparative analysis of models and methods of synthesis of recommendation systems of virtual social networks and content-oriented websites were carried out. which showed that the vast majority of existing models and methods are vulnerable to internal and external destabilizing factors in computer networks. It is shown that ensuring the stability of recommendation systems to the action of destabilizing factors is an important condition for improving their accuracy.

A method for determining the dynamics of the probabilities of the recommendation system in its possible states using the mathematical apparatus of Markov and semi-Markov processes are developed. This allows to determine a relationship between the set of density distributions of random durations of staying of the system in these states and functions for describing dynamics of the specific recommendation system in its possible states at any time. On the basis of the proposed method a method of obtaining analytical relations to calculate the probabilities of the system in its possible states at any time is developed.

A mathematical model of a stable recommendation system based on the proposed method for determining the dynamics of the probabilities of the system in

its possible states is developed. This allows to optimize the total cost of maintaining the system in conditions of internal destabilizing factors. On the basis of the developed mathematical model a method for determining the total cost of the subsystem for collecting and recalculating input data is proposed, as well as a method for determining the optimal frequency of recalculation of input data, at which the system has minimum unprofitableness.

The method of collaborative filtering has been improved.This method differs from the existing ones in using production rules to determine user similarity and using user activity indicators to form recommendations. This allows to increase system stability in case of insufficient input and cold start. Appropriate algorithms were developed and implemented. Usage of them allowed to increase 1.6 times the stability of the recommendation system in comparison with the known method of collaborative filtering based on the neighborhood model and 3.2 times in comparison with the known method of collaborative filtering based on matrix factoring.

A mathematical model of the information security subsystem of a stable recommendation system is developed on the basis of the proposed method of determining the dynamics of probabilities of the system in its possible states. This allows to determine the optimal frequency of checking for information attack and bot profiles. Within the limits of the mathematical model a set of possible states of recommendation system in the conditions of information attacks by injection of profiles is developed, possible transitions between these states is determined. A method of obtaining analytical relations for calculating the probabilities of the information security subsystem in its possible states at any time is developed. On the basis of proposed mathematical model, a method for determining of the total costs suffered by the recommendation system due to monitoring of its own information security, neutralization of botnet activities and due to information attacks by injection of profiles is proposed. The developed method allows to determine the total cost of maintenance of the security subsystem of the

recommendation system with known costs for computing resources and known losses during botnet attacks. Also a method for determining of the optimal frequency of checking the recommendation system for the presence of information attacks and bot profiles is developed to optimize the overall expenses of the system.

A method of software simulation modeling of users and objects of the recommendation system of a social network or web resource based on existing and developed methods of modeling the structure of complex networks and methods of modeling user behavior is developed. This allows to generate input data to test the quality of recommendation algorithms. A method of modeling changes in user preferences over time of the recommendation system is developed. This allows to generate test data sets that are more similar in statistical characteristics to real ones. The method is based on a mathematical model of non-cyclic changes in user preferences over time using the exponential distribution law and the law of radioactive decay of elements. Appropriate algorithms are developed and implemented, and a software simulation model of users, objects, botnets and information processes in the recommendation system is implemented. Using the developed software simulation model data sets for testing the work of methods of synthesis of recommendation systems and subsystems of their information security are formed.

A method of detecting an information attack on the recommendation system based on the analysis of object rating trends is developed. This allows to reduce the cost of monitoring the security of the system by eliminating the need to search for bots in the absence of signs of attack. Appropriate algorithms are developed and implemented. This allowed to detect 76% of information attack objects in the recommendation system. Many of the identified probable targets of the attack can be used for further search for bot profiles. This will allow during search for botnets to check not all system profiles, as in known methods, but only those that interacted with the probable targets of the attack.

A method of detecting botnets in the recommendation system based on graph clustering and analysis of user actions is developed. This allows to detect botnets and distinguish them on the basis of sets of attack objects. A method for identifying individual bot profiles based on neural networks in recommendation systems to refine the results of detecting bot networks based on graph clustering was also developed. Appropriate algorithms are developed and the information security subsystem of the recommendation system is implemented. The developed information security subsystem allows to provide higher stability of the recommendation system to external destabilizing factors in contrast to existing methods. On average, it increases the stability of the recommendation system by 2.5 times to random and medium profile injection attacks and by 1.7 times to popular profile injection attacks when used with a neighborhood-based collaborative filtering method or a proposed hybrid collaborative filtering method.

An assessment of the reliability and effectiveness of the proposed methods and models to increase the stability of recommendation systems are carried out.

Keywords: recommendation systems, computer networks, cold start problem, collaborative filtering, data analysis, clustering, modeling of complex networks, information attacks, bot networks, social networks, web resources

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

1. Meleshko Ye., Drieiev O., Yakymenko M., Lysytsia D. Developing a model of the dynamics of states of a recommendation system under conditions of profile injection attacks // Eastern-European Journal of Enterprise Technologies (ISSN 1729-3774). – 2020. – Vol. 4, No 4(106). – P. 14-24. **(SCOPUS)**

2. Meleshko Ye., Raskin L., Semenov S., Sira O. Methodology of probabilistic analysis of state dynamics of multi-dimensional semi-Markov dynamic systems // Eastern-European Journal of Enterprise Technologies (ISSN: 1729-3774). – 2019. – Vol. 6, No 4(102). – P. 6-13. **(SCOPUS)**

3. Ulichev O., Meleshko Ye., Smirnov O., Khokh V., Goncharenko Iu. Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process // CEUR-WS, Vol 2588, Lviv, Ukraine (ISSN: 1613-0073). – 2019. – P. 215-227. **(SCOPUS)**

4. Ulichev O.S., Meleshko Ye.V., Sawicki D., Smailova S. Computer modeling of dissemination of informational influences in social networks with different strategies of information distributors // Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, Wilga, Poland (ISSN: 0277-786X). – 2019. – 111761T. **(SCOPUS)**

5. Ulichev O., Meleshko Y., Khokh V. The computer simulation method of a social network structure for the research of dissemination processes of informational influences // Scientific and Practical Cyber Security Journal (SPCSJ) 4(3). – Georgia, Tbilisi, 2019. – P. 34-47.

6. Meleshko Ye. Computer model of virtual social network with recommendation system // Innovative technologies and scientific solutions for industries. – 2019. – №2(8). – P. 80-85.

7. Meleshko Ye. Method of generating recommendations lists with considering activity indexes of users in a recommendation system // Advanced information systems. – 2019. – Т. 3, № 1. – P. 43-47.

8. Мелешко Є.В. Методи кластеризації графів соціальних мереж для побудови рекомендаційних систем // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 2 (54). – С. 129-134.

9. Meleshko Ye. Method of collaborative filtration based on associative networks of users similarity // Advanced information systems. – 2018. – Т. 2, № 4. – P. 55-59.

10. Мелешко Є.В. Проблеми сучасних рекомендаційних систем та методи їх рішення // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2018. – Т. 4 (50). – С. 120-124.

11. Мелешко Є.В. Методи оцінки якості роботи рекомендаційних систем // Системи управління, навігації та зв'язку. – Полтава: ПНТУ, 2018. – Вип. 5 (51). – С. 92-97.

12. Мелешко Е.В. Метод встраивания двухуровневых цифровых водяных знаков в медиафайлы для защиты авторских прав // Збірник наукових праць Харківського університету Повітряних сил. – 2013. – Вип. 4. – С. 127-131.

13. Meleshko Ye., Drieiev O., Drieieva H. Method of identification bot profiles based on neural networks in recommendation systems // Advanced Information Systems. – 2020. – Vol. 4, No. 2 – P. 24-28.

14. Міхав В.В., Мелешко Є.В., Якименко М.С. Метод зберігання даних рекомендаційної системи на основі бінарних діаграм рішень // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2020. – Т. 2 (60). – С. 85-89.

15. Мелешко Є.В., Хох В.Д., Босько В.В. Дослідження матричних факторизаційних моделей рекомендаційних систем // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 6 (58). – С. 58-62.

16. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження відомих моделей атак на рекомендаційні системи з колаборативною фільтрацією // Збірник наукових праць Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – №. 5 (57). – С. 67-71.

17. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження робастності рекомендаційних систем з колаборативною фільтрацією до інформаційних атак // Електронне фахове наукове видання Кібербезпека: освіта, наука, техніка.– Київ: КУБГ, 2019. – Т.1, № 5. – С. 95-104.

18. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М., Резніченко В.А. Математична модель рекомендаційної системи з врахуванням емоційного забарвлення коментарів у якості контексту // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кропивницький: ЦНТУ, 2018. – Вип. 31. – С. 181-186.

19. Улічев О.С., Мелешко Є.В. Програмне моделювання поширення інформаційно-психологічних впливів у віртуальних соціальних мережах // Сучасні інформаційні системи. – 2018. – Т. 2, № 2. – С. 35-39.

20. Мелешко Є.В., Семенов С.Г., Хох В.Д. Дослідження методів побудови рекомендаційних систем в мережі Інтернет // Збірник наукових праць "Системи управління, навігації та зв'язку". Випуск 1(47). – Полтава: ПНТУ ім. Ю. Кондратюка. – 2018. – С. 131-136.

21. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М., Резніченко В.А. Методи автоматичного аналізу тональності контенту у соціальних мережах для виявлення інформаційно-психологічних впливів // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – 2017. – Вип. 30. – С. 196-202.

22. Лысенко И.А., Смирнов А.А., Мелешко Е.В. Исследование уровней тестирования программного обеспечения инфотелекоммуникационных

систем // Наука і техніка Повітряних Сил Збройних Сил України. – 2014. – № 4. – С. 79-81.

23. Даниленко Д.О., Смірнов О.А., Мелешко Є.В. Дослідження методів виявлення вторгнень в телекомунікаційні системи та мережі // Системи озброєння і військова техніка. – 2012. – № 1. – С. 92-100.

24. Смірнов О.А., Даниленко Д.О., Мелешко Є.В. Метод обнаружения вредоносного программного обеспечения. Часть 1. Корреляционный анализ сетевого трафика // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті». – 2012. – № 4(95). – С. 8-14.

25. Дрєєв О.М., Смірнов О.А., Мелешко Є.В., Коваленко О.В. Метод прогнозування завантаженості серверу телекомунікаційної мережі // Системи обробки інформації. – 2012. – Вип. 3(2). – С. 181-187.

26. Mohammed A.S., Meleshko Y., Balaji S.B., Semenov S. Collaborative filtering method with the use of production rules // Proceedings of ICCIKE, Amity University Dubai; United Arab Emirates. – 2019. – с. 387-391. **(SCOPUS)**

27. Мелешко Е.В., Смирнов А.А. Исследование методов структурного анализа социальных сетей с точки зрения информационной безопасности // Материалы XXI Международной научно-технической конференции "Современные средства связи", 20-21 октября 2016 года, Минск, Республика Беларусь – Минск: Белорусская государственная академия связи. – 2016. – С. 175-177.

28. Шингалов Д.В., Мелешко Є.В., Босько В.В. Методи нормалізації даних для моделей машинного навчання // Збірник тез XXII Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 197-200.

29. Мелешко Є.В., Якименко М.С., Резніченко В.А. Методи оцінки якості роботи алгоритмів машинного навчання // Збірник тез XXII Міжнародного науково-практичного семінару «Комбінаторні конфігурації та

їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 90-94.

30. Мелешко Є.В., Дрєєва Г.М., Дрєєв О.М. Метод кластеризації користувачів соціальної мережі на основі нейронних мереж // Збірник тез XXII Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 87-90.

31. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження методів підвищення робастності рекомендаційних систем до інформаційних атак // Матеріали VI Міжнародної науково-практичної конференції «Актуальні питання забезпечення кібербезпеки та захисту інформації», 19-22 лютого 2020 р. – м. Київ: Вид-во Європейського університету, 2020. – С. 65-70.

32. Мелешко Є.В., Дрєєва Г., Якименко М., Хох В. Методи моделювання складних мереж // Матеріали IX Міжнародної наукової конференції "Інформація. Комунікація. Суспільство", м. Львів, 21-23 травня 2020 р. – Львів: Видавництво Львівської політехніки. – 2020. – С. 29-30.

33. Міхав В.В., Мелешко Є.В. Порівняння стратегій редагування бінарних діаграм рішень для роботи з графовими даними // Матеріали IX Міжнародної наукової конференції "Інформація. Комунікація. Суспільство", м. Львів, 21-23 травня 2020 р. – Львів: Видавництво Львівської політехніки. – 2020. – С. 17-18.

34. Мелешко Є.В., Дрєєв О.М., Дрєєва Г.М. Розробка методу ідентифікації ботів у рекомендаційних системах // Матеріали X Міжнародної науково-практичної конференції “Комплексне забезпечення якості технологічних процесів та систем”, м. Чернігів, 29-30 квітня 2020 р. – Чернігів: ЧНТУ. – 2020. – С. 165-166.

35. Шингалов Д.В., Мелешко Є.В., Босько В.В. Дослідження моделей репутації користувачів соціальної мережі // Збірник тез II Міжнародної науково-практичної конференції “Інформаційна безпека та інформаційні

технології”, м. Кропивницький, 2-3 квітня 2020 року. – Кропивницький: ЦНТУ. – 2020. – С. 29.

36. Міхав В.В., Мелешко Є.В. Метод оптимізації швидкодії бінарних діаграм рішень при представленні даних рекомендаційної системи // Збірник тез II Міжнародної науково-практичної конференції “Інформаційна безпека та інформаційні технології”, м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 17.

37. Мелешко Є.В., Хох В.Д. Дослідження моделей рекомендаційних систем на основі прихованих факторів // Збірник тез II Міжнародної науково-практичної конференції “Інформаційна безпека та інформаційні технології”, м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 46.

38. Мелешко Є.В., Дрєєва Г.М., Гермак В.С., Резніченко В.А., Шевченко О.О. Методи визначення ботів серед користувачів соціальних мереж // Збірник тез II Міжнародної науково-практичної конференції “Інформаційна безпека та інформаційні технології”, м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 44.

39. Мелешко Є., Хох В., Резніченко В., Босько В. Дослідження методів оцінювання робастності рекомендаційних систем до атак накручування рейтингів // Збірник тез IV Всеукраїнської науково-практичної конференції «Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем MEICS-2019», м. Дніпро, 27-29 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 10-11.

40. Шингалов Д.В., Мелешко Є.В., Улічев А.С. Дослідження Баєсових мереж довіри як засобів для моделювання динамічних процесів у складних мережах // Збірник тез XVII Міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем», м. Дніпро, 20-22 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 284-285.

41. Мелешко Є.В., Хох В.Д., Минайленко Р.М. Розробка експертної системи для виявлення атак на рекомендаційні мережі // Збірник тез XVII Міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем», м. Дніпро, 20-22 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 176-177.

42. Мелешко Є.В., Охотний С.М., Резніченко В.А. Дослідження методів визначення спільнот у соціальному графі // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 92-93.

43. Мелешко Є.В., Хох В.Д., Улічев О.С. Методи тестування робастності рекомендаційних систем з колаборативною фільтрацією // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 88-89.

44. Мелешко Є.В., Чабан О.О., Міхав В.В. Способи побудови рекомендаційних систем для соціальних мереж з врахуванням репутації користувачів // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 86-87.

45. Мелешко Є.В., Дрєєва Г.М. Дезінформаційні атаки на рекомендаційні системи // Збірник тез Міжнародної наукової конференції «Безпека в сучасному світі», м. Дніпро, 27-28 вересень 2019 р. – Дніпро: ДНУ

ім. Олеся Гончара. – 2019. – С. 51-53.

46. Мелешко Є.В. Методи кластеризації графів для побудови рекомендаційних систем соціальних медіа // Збірник тез VII Міжнародної науково-практичної конференції «Обробка сигналів і негаусівських процесів», присвячена пам'яті професора Кунченка Ю.П., м. Черкаси, 23-24 травня 2019 р., – Черкаси: ЧДТУ. – 2019. – С. 100-102.

47. Мелешко Є.В., Чабан О.О. Дослідження засобів парсингу веб-сайтів // Матеріали XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня 2019 року. – Кропивницький: КЛА НАУ. – 2019. – С. 163-167.

48. Мелешко Є.В. Дослідження засобів кластеризації графів у графовій СУБД Neo4j для виявлення співтовариств у соціальних мережах // Збірник тез VIII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство», смт. Чинадієво, 16-18 травня 2019 р. – Львів: Видавництво Львівської політехніки. – 2019. – С. 19-20.

49. Мелешко Є.В., Босько В.В., Резніченко В.А. Дослідження методів комп'ютерної лінгвістики для аналізу контенту веб-сайтів // Збірник тез XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня 2019 р. – Кропивницький: КЛА НАУ. – 2019. – С. 96-100.

50. Мелешко Є.В. Метод побудови рекомендаційних систем на основі асоціативних мереж користувачів // Збірник тез XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня 2019 р. – Кропивницький: КЛА НАУ. – 2019. – С. 91-95.

51. Мелешко Є.В., Охотний С.М., Босько В.В. Розробка програмного забезпечення для збору та аналізу даних із соціальних мереж // Збірник тез IX Міжнародної науково-практичної конференції «Комплексне забезпечення якості технологічних процесів та систем», Т.2, м. Чернігів, 14-16 травня 2019

р. – Чернігів: ЧНТУ. – 2019. – С. 225-226.

52. Мелешко Є.В. Метод визначення подоби між користувачами у рекомендаційних системах з колаборативною фільтрацією // Збірник тез Науково-практичної конференції «Інформатика, математика, автоматика (ІМА-2019)», м. Суми, 23-26 квітня 2019 р. – Суми: СДУ. – 2019. – С. 213-214.

53. Мелешко Є.В. Розробка програмного забезпечення для виділення співтовариств у соціальній мережі // Збірник тез ІІІ Всеукраїнської науково-практичної конференції «Перспективні напрямки сучасної електроніки, інформатики і комп'ютерних систем», м. Дніпро, 21-23 листопада 2018 р. – Дніпро: ДНУ. – 2018. – С. 42-43.

54. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М. Дослідження програмних засобів для аналізу та візуалізації соціальних графових структур // Збірник тез V Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії (IT&I-2018)», м. Київ, 20-21 листопада 2018 р. – Київ: КНУ. – 2018. – С. 159-160.

55. Мелешко Є.В., Босько В.В., Резніченко В.А. Розробка рекомендаційної системи на базі СУБД Neo4j // Збірник тез V Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії (IT&I – 2018)», м. Київ, 20-21 листопада 2018 р. – Київ: КНУ. – 2018. – С. 351-352.

56. Мелешко Є.В., Босько В.В., Резніченко В.А. Застосування асоціативних мереж для побудови рекомендаційних систем // Збірник тез Міжнародної науково-практичної Інтернет-конференції «Автоматика, комп'ютерно-інтегровані технології та проблеми енергоефективності в промисловості і сільському господарстві (АКІТ-2018)», м. Кропивницький, 15-16 листопада 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 165-166.

57. Мелешко Є.В. Дослідження проблем сучасних рекомендаційних систем та методів їх рішення // Збірник тез Міжнародної науково-практичної конференції «Контроль і управління в складних системах (КУСС-2018)», м.

Вінниця, 15-17 жовтня 2018 р. – Вінниця: ВНТУ. – 2018. – С. 126.

58. Мелешко Є.В. Методи оцінки точності прогнозування вподобань користувачів веб-ресурсів рекомендаційними системами // Збірник тез X Всеукраїнської науково-практичної конференції «Стан та удосконалення безпеки інформаційно-телекомунікаційних систем (SITS'2018)», с. Коблево, 21-23 червня 2018 р. – Миколаїв: НАУ та МІПРО. – 2018. – С. 58-61.

59. Мелешко Є.В., Дреєва Г.М. Дослідження проблем сучасних рекомендаційних систем // Збірник тез VII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство (ICS-2018)», 17-19 травня 2018 р., Україна, смт. Чинадієво. – Львів: НУ «Львівська політехніка». – 2018. – С. 31-32.

60. Мелешко Є.В. Загрози інформаційній безпеці у рекомендаційних системах соціальних медіа // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м. Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 24-25.

61. Мелешко Є.В. Методи оцінки якості роботи рекомендаційних систем // Матеріали XX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 13-14 квітня 2018 р. – Кропивницький: КЛА НАУ. – 2018. – С. 68-72.

62. Мелешко Є.В. Дослідження методів побудови рекомендаційних систем заснованих на фільтрації контенту // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький, 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 234-237.

63. Улічев О.С., Мелешко Є.В. Програмна модель соціальної мережі та стратегій поширення інформаційно-психологічних впливів // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький, 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 136-220.

64. Охотний С.М., Мелешко Є.В. Визначення центральностей у соціальному графі засобами графової бази даних Neo4j // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 247-248.

65. Улічев О.С., Мелешко Є.В. Моделювання розповсюдження інформаційно-психологічних впливів у сегменті соціальної мережі // Збірник тез VII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство (ICS-2018)», 17-19 травня 2018 р., Україна, смт. Чинадієво. – Львів: НУ «Львівська політехніка». – 2018. – С. 29-30.

66. Мелешко Є.В., Хох В.Д., Сидоренко В.В. Розробка автоматизованої системи виявлення, оцінки та розробки заходів по усуненню загроз в інформаційних системах // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м. Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 38-39.

67. Улічев О.С., Мелешко Є.В. Програмна модель розповсюдження інформаційно-психологічних впливів в сегменті соціальної мережі // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м. Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 34-35.

68. Улічев О.С., Мелешко Є.В. Математична модель розповсюдження інформації в сегменті соціальної мережі // Матеріали XX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 13-14 квітня 2018 р. – Кропивницький: КЛА НАУ. – 2018. – С. 68-72.

69. Мелешко Є.В., Гермак В.С. Дослідження впливу структури соціальної мережі на захищеність від поширення вірусної інформації // Збірник тез доповідей III Міжнародної науково-практичної конференції "Актуальні питання забезпечення кібербезпеки та захисту інформації". с.

Верхнє Студене, 22-25 лютого 2017 р. – Київ: Видавництво Європейського університету. – 2017. – С. 118-119.

70. Охотний С.М., Мелешко Є.В., Константинова А.А. Розробка бота для соціальної мережі Facebook на основі фреймворка Selenium // Збірник тез Всеукраїнської науково-практичної Інтернет-конференції "Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті", м. Кропивницький, 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 202-203.

71. Мелешко Є.В., Якименко М.С. Методи виявлення інформаційно-емоційних впливів у текстовій інформації // Збірник тез VI Міжнародної наукової конференції «Інформація. Комунікація. Суспільство (ICS-2017)», м. Львів, 18-20 травня 2017 р. – Львів: Національний університет "Львівська політехніка". – 2017. – С. 30-31.

72. Охотний С.М., Мелешко Є.В. Збирання даних про користувачів віртуальної соціальної мережі за допомогою web-кроулера // Збірник тез II Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький, 20-22 квітня 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 157-159.

73. Мелешко Є.В., Шингалов Д.В., Минайленко Р.М. Методи автоматизації побудови графових структур спільнот у соціальних мережах // Матеріали XIX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» присвяченого пам'яті д.ф.-м.н., професора Петренюка Анатолія Яковича, м. Кропивницький, 7-8 квітня 2017 року. – Кропивницький: КЛА НАУ. – 2017. – С. 162-164.

74. Мелешко Є.В. Дослідження методів динамічного аналізу віртуальних соціальних мереж з точки зору інформаційної безпеки // Матеріали Всеукраїнської науково-практичної конференції "Кібербезпека в Україні: правові та організаційні питання", м. Одеса, 21 жовтня 2016 р. – Одеса: ОДУВС. – 2016. – С. 154-155.

75. Мелешко Є.В. Аналіз структури соціальної мережі з точки зору інформаційної безпеки // Збірник тез XVIII міжнародного науково-практичного семінару "Комбінаторні конфігурації та їх застосування", м. Кіровоград, 15-16 квітня 2016. – Кіровоград: Кіровоградський національний технічний університет. – 2016. – С. 93-97.

76. Мелешко Є.В. Методи протидії деструктивним інформаційним впливам в соціальних мережах в умовах інформаційної війни // Збірник тез Всеукраїнської науково-практичної конференції «Інформаційна безпека держави суспільства та особистості», м. Кіровоград, 16 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 139-142.

ЗМІСТ

ВСТУП

Актуальність. Актуальність дослідження зумовлена стрімким збільшенням у комп'ютерних мережах кількості контент-орієнтованих веб-ресурсів та віртуальних соціальних мереж, які використовують рекомендаційні системи. За допомогою рекомендаційних систем користувач швидше знаходить потрібний саме йому контент, а власник збільшує відвідуваність свого веб-ресурсу, а отже, і власний прибуток. Тому вони стають такою ж важливою частиною веб-сайтів, як і пошукові підсистеми, інколи доповнюючи їх, а інколи створюючи їм альтернативу [76, 209].

В той же час, проведені дослідження показали, що рекомендаційні системи вразливі до ряду внутрішніх [9, 34, 76, 146, 147, 157, 163, 171, 209] та зовнішніх [22, 28, 40, 46, 47, 60, 61, 69, 148, 176, 177] дестабілізуючих факторів, що значно впливають на точність їх роботи, а отже, й ефективність використання, як для користувачів, так і для власників веб-ресурсів. Прикладами внутрішніх дестабілізуючих факторів у рекомендаційних системах можуть бути проблема холодного старту та проблема недостатньої кількості і якості вхідних даних [9, 34, 76]. Основним зовнішнім дестабілізуючим фактором у рекомендаційних системах є інформаційні атаки ін'єкцією профілів [22, 28, 40, 46, 60, 69, 76].

Найбільш важливими роботами в галузі дослідження, розробки та вдосконалення рекомендаційних систем є роботи зарубіжних та вітчизняних науковців, серед яких варто відзначити наступних: Річчі Ф. [76], Рокач Л. [76], Шапіра Б. [76], Кантор П.Б. [76], Джонс М. [36, 37], Берк Р. [15, 60-62, 98], Мобашер Б. [60-62, 98], Вільямс Ч. [60, 61, 98], Фанк С. [27], Бернарді Л. [9], Кастеллс П. [21], Варгас С. [21], Пасічник В.В. [4, 200], Артеменко О.І. [4, 200], Лобур М.В. [82, 137, 138], Стех Ю.В. [81, 82, 137, 138].

Для тестування моделей та методів синтезу рекомендаційних систем досить важливим є створення програмних імітаційних моделей користувачів, об'єктів та інформаційних процесів веб-ресурсів та соціальних мереж для

генерації навчальних і тестових вибірок даних, а також симуляції реакції користувачів на створені списки рекомендацій. Водночас для рекомендаційних систем таких моделей практично немає, переважно вони тестуються на відкритих наборах даних [30, 76, 171, 209], що не дозволяє в повній мірі досліджувати реакцію користувачів на запропоновані рекомендації та вплив бот-мереж.

Найбільш важливими роботами у напрямку моделювання поведінки користувачів та інформаційних процесів у комп'ютерних мережах, спрямованими на широке застосування, є роботи зарубіжних та вітчизняних науковців, серед яких слід відзначити наступних: Барабаши А.-Л. [2, 5-8], Альберт Р. [2, 6-8], Трааг В. [88], Губанов Д.О. [119, 120], Новіков Д.О. [119, 120], Чхартішвілі О.Г. [119, 120], Пасічник В.В. [201], Ланде Д.В. [118, 136, 211], Снарський А.О. [136, 211], Додонов О.Г. [118].

Проведене дослідження відомих моделей та методів синтезу рекомендаційних систем [157, 171, 173, 178] показало, що на даний момент не в повній мірі вирішено питання забезпечення стійкості рекомендаційних систем до дестабілізуючих факторів, внаслідок чого вплив таких факторів значно знижує точність формування рекомендацій користувачам.

Таким чином, на сьогоднішній день в теорії і практиці функціонування рекомендаційних систем загострилося **протиріччя** між підвищенням вимог до точності пропозицій користувачам рекомендаційних систем, збільшенням ризиків впливу на цей процес внутрішніх і зовнішніх дестабілізуючих факторів та існуючим станом теоретичного обґрунтування, синтезу і практичної реалізації підсистем забезпечення стійкості до цих негативних впливів.

Подолати цю суперечність можна шляхом вирішення актуальної **науково-практичної проблеми** підвищення точності пропозицій рекомендаційних систем в умовах дестабілізуючих факторів у комп'ютерних мережах на основі розробки моделей та методів синтезу підсистеми забезпечення стійкості.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана у межах пріоритетних наукових напрямів, які охоплюють актуальні проблеми, відповідно до рішення Ради президентів академій наук України від 30 січня 2019 року «Про Основні наукові напрями та найважливіші проблеми фундаментальних досліджень у галузі природничих, технічних, суспільних і гуманітарних наук Національної академії наук України на 2019-2023 роки», «Інформатика» за темами: «Розроблення обчислювальних алгоритмів і процедур з метою вирішення практичних задач міждисциплінарного характеру для застосувань, що належать до науково-технічної та соціально-економічної сфер діяльності людини», «Розроблення математичних методів та систем моделювання об'єктів та процесів». Дисертаційну роботу виконано у межах зареєстрованих НДР Центральноукраїнського національного технічного університету: «Методи застосування штучних нейронних мереж в телекомунікаційних системах для обробки та аналізу даних» (ДР № 0116U008161) та «Моделювання та аналіз складних мереж та інформаційних систем» (ДР № 0119U003587), а також НДР Національного технічного університету «Харківський політехнічний інститут»: «Дослідження методів управління та захисту даних в комп'ютеризованих інформаційно-вимірювальних та розподілених системах» (ДР № 0119U002603).

Мета і задачі дослідження. Мета дисертаційної роботи – підвищення стійкості рекомендаційних систем соціальних мереж та веб-сервісів до внутрішніх та зовнішніх дестабілізуючих факторів у комп'ютерних мережах.

Мета дисертаційної роботи визначає необхідність розв'язання таких **основних задач:**

1. Дослідження моделей та методів синтезу рекомендаційних систем для соціальних мереж та контент-орієнтованих веб-сайтів у комп'ютерних мережах.

2. Розробка методу визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах та методики отримання

аналітичних співвідношень для безпосереднього розрахунку цих ймовірностей з метою створення на їх основі математичних моделей конкретних рекомендаційних систем.

3. Розробка математичної моделі динаміки ймовірностей перебування стійкої рекомендаційної системи в своїх можливих станах для створення на її основі методів формування списків рекомендацій, стійких до внутрішніх дестабілізуючих факторів.

4. Розробка методу та алгоритмів формування списків рекомендацій, стійких до внутрішніх дестабілізуючих факторів рекомендаційної системи.

5. Розробка математичної моделі динаміки ймовірностей перебування підсистеми інформаційної безпеки стійкої рекомендаційної системи в своїх можливих станах для створення на її основі методів формування списків рекомендацій, стійких до зовнішніх дестабілізуючих факторів.

6. Розробка методу, алгоритмів та способів програмного імітаційного моделювання користувачів, об’єктів та інформаційних процесів рекомендаційної системи для тестування алгоритмів її роботи.

7. Розробка методу та алгоритмів виявлення інформаційних атак на рекомендаційну систему.

8. Розробка методу, алгоритмів та способів ідентифікації профілів ботів і виявлення бот-мереж у рекомендаційних системах.

9. Обґрунтування достовірності одержаних результатів наукових досліджень.

Об’єктом дослідження є процес функціонування рекомендаційних систем соціальних мереж та веб-сайтів у комп’ютерних мережах.

Предметом дослідження є методологія забезпечення стійкості рекомендаційних систем в умовах дестабілізуючих факторів.

Методи дослідження. Для вирішення завдань математичного моделювання рекомендаційної системи використано теорію марківських та напівмарківських процесів, теорію графів, теорію складних мереж та теорію ймовірностей. Для створення програмної імітаційної моделі використано

методи об’єктно-орієнтованого програмування та методи роботи з графовими базами даних. Для розробки методів синтезу рекомендаційних систем та підсистем забезпечення стійкості також було використано теорію статистичної обробки даних, теорію штучного інтелекту, теорію інформаційної безпеки та теорію технічного аналізу.

Наукова новизна одержаних результатів полягає у такому:

– **Вперше розроблено** метод визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах з використанням математичного апарату марківських та напівмарківських процесів, що дає можливість встановлення зв’язку між набором щільності розподілу випадкових тривалостей перебування системи у цих станах і функціями опису динаміки ймовірностей станів для визначення ймовірностей перебування конкретної рекомендаційної системи в своїх можливих станах в довільний момент часу.

– **Вперше розроблено** математичну модель стійкої рекомендаційної системи на основі запропонованого методу визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило здійснити оптимізацію загальних витрат на обслуговування системи в умовах внутрішніх дестабілізуючих факторів.

– **Удосконалено** метод колаборативної фільтрації, який відрізняється від існуючих використанням продукційних правил для визначення подоби користувачів та використанням показників активності користувачів для формування рекомендацій, що дозволило підвищити стійкість системи у випадку недостатньої кількості вхідних даних та під час холодного старту.

– **Вперше розроблено** математичну модель підсистеми інформаційної безпеки стійкої рекомендаційної системи на основі запропонованого методу визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило визначити оптимальну частоту перевірки на наявність інформаційної атаки та профілів ботів.

– **Вперше розроблено** метод імітаційного програмного моделювання користувачів та об’єктів рекомендаційної системи соціальної мережі або веб-ресурсу на основі існуючих і розроблених методів моделювання структури складних мереж та методів моделювання поведінки користувачів, що дозволило генерувати вхідні дані для тестування якості роботи алгоритмів формування рекомендацій.

– **Вперше розроблено** метод виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів рейтингів об’єктів, що дозволило знизити кількість витрат на моніторинг безпеки системи за рахунок зняття необхідності пошуку ботів при відсутності ознак атаки.

– **Вперше розроблено** метод виявлення бот-мереж у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів, що дозволило виявляти бот-мережі та розрізняти їх за множинами об’єктів атаки.

Практичне значення одержаних результатів. Отримані в дисертаційній роботі результати дають змогу підвищити стійкість рекомендаційних систем до внутрішніх та зовнішніх дестабілізуючих факторів, що в свою чергу дозволяє забезпечити достатній рівень точності та інших показників якості формування списків рекомендацій.

Практична цінність роботи полягає у такому:

– розроблено алгоритми програмного імітаційного моделювання користувачів, об’єктів та інформаційних процесів рекомендаційної системи, які дозволяють генерувати вхідні дані для тестування алгоритмів формування списків рекомендацій;

– розроблено вдосконалені алгоритми колаборативної фільтрації даних для формування більш точних списків рекомендацій користувачам веб-ресурсів на основі продукційних правил та використання показників активності користувачів;

– розроблено алгоритми виявлення наявності інформаційної атаки на рекомендаційну систему на основі аналізу трендів рейтингів об’єктів системи;

– розроблено алгоритми виявлення окремих профілів ботів на основі нейронних мереж та алгоритми виявлення бот-мереж на основі графової кластеризації та аналізу дій користувачів у рекомендаційній системі;

– розроблено методику одержання аналітичних співвідношень для розрахунку ймовірностей перебування стійкої рекомендаційної системи в своїх можливих станах в довільний момент часу для оптимізації частоти перерахунку вхідних даних для формування списків рекомендацій;

– розроблено методику одержання аналітичних співвідношень для розрахунку ймовірностей перебування підсистеми інформаційної безпеки стійкої рекомендаційної системи в своїх можливих станах для визначення оптимальної частоти перевірки на наявність інформаційної атаки та ботів.

Практичне значення отриманих результатів підтверджено відповідними актами впровадження.

Результати дисертації впроваджені і використовуються у діяльності Компанії «Line Up», Державного підприємства «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості», Державного підприємства «Харківський науково-дослідний інститут технологій машинобудування», Національного наукового центру «Інститут судових експертиз ім. Засл. проф. М.С. Бокаріуса», а також використано у навчальному процесі Центральноукраїнського національного технічного університету та Національного технічного університету «Харківський політехнічний інститут».

Особистий внесок здобувача. Усі наукові результати дисертаційної роботи автор отримав самостійно. У друкованих працях, опублікованих у співавторстві, здобувачеві належать: [125] – дослідження стійкості серверу комп’ютерної мережі до завантаженості даними; [122, 141, 162, 175, 210, 216, 230] – дослідження методів виявлення інформаційних атак та впливів на системи обробки даних; [139] – дослідження методів тестування обчислювальних алгоритмів; [160, 161, 163, 171-173, 181, 189, 226, 229] – дослідження методів побудови рекомендаційних систем; [89, 91, 165, 182,

215-219, 231] – дослідження методів програмного імітаційного моделювання складних систем та інформаційних процесів; [90, 142, 166, 167, 176-179] – дослідження інформаційних загроз та методів захисту від них у рекомендаційних системах; [58] – розробка методу визначення динаміки ймовірностей перебування системи в своїх можливих станах з використанням математичного апарату марківських та напівмарківських процесів; [188] – розробка способу програмного імітаційного моделювання рекомендаційних систем для проведення тестування якості роботи їх алгоритмів; [56, 164, 174] – розробка способу ідентифікації ботів у рекомендаційних системах на основі нейронних мереж; [57] – розробка математичної моделі підсистеми інформаційної безпеки стійкої рекомендаційної системи; [64] – розробка методу колаборативної фільтрації на основі продукційних правил; [169, 183, 196] – розробка способу формування тестових вибірок для систем аналізу даних; [159, 168, 170, 180, 184, 190, 196, 227, 228] – дослідження методів збору та обробки інформації з соціальних мереж та контент-орієнтованих веб-сайтів.

З робіт, що опубліковані у співавторстві, у дисертаційній роботі використовуються виключно результати, отримані особисто здобувачем.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися та обговорювалися на таких наукових конференціях та семінарах: IEEE International Conference on Computational Intelligence and Knowledge Economy ICCIKE-2019 (United Arab Emirates, Dubai, 2019 p.); Міжнародна науково-технічна конференція «Сучасні засоби зв'язку» (Республіка Білорусь, Мінськ, 2016 р.); Всеукраїнська науково-практична конференція «Інформаційна безпека держави суспільства та особистості» (Кіровоград, 2015 р.); Міжнародний науково-практичний семінар «Комбінаторні конфігурації та їх застосування» (Кропивницький, 2016-2020 рр.); Всеукраїнська науково-практична конференція «Кібербезпека в Україні: правові та організаційні питання» (Одеса, 2016 р.); Міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології»

(Кропивницький, 2017-2018 рр.); Міжнародна наукова конференція «Інформація. Комунікація. Суспільство» (Львів, 2017-2020 рр.); Всеукраїнська науково-практична Інтернет-конференція "Автоматика та комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті (АКІТ)", (Кропивницький, 2017-2018 рр.); Міжнародна науково-практична конференція «Актуальні питання забезпечення кібербезпеки та захисту інформації» (с. Верхнє Студене – Київ, 2017, 2020 рр.); Міжнародна науково-технічна конференція «ITSEC» (Київ, 2018 р.); Всеукраїнська науково-практична конференція «Стан та удосконалення безпеки інформаційно-телекомунікаційних систем (SITS’2018)» (Миколаїв-Коблево, 2018 р.); Міжнародна науково-практична конференція «Контроль і управління в складних системах (КУСС-2018)» (Вінниця, 2018 р.); Міжнародна науково-практична конференція «Інформаційні технології та взаємодії (IT&I)» (Київ, 2018 р.); Всеукраїнська науково-практична конференція «Перспективні напрямки сучасної електроніки, інформатики і комп’ютерних систем» (Дніпро, 2018 р.); Науково-практична конференція «Інформатика, математика, автоматика (ІМА)» (Суми, 2019 р.); Міжнародна науково-практична конференція «Комплексне забезпечення якості технологічних процесів та систем» (Чернігів, 2019-2020 р.); Міжнародна науково-практична конференція «Обробка сигналів і негаусівських процесів» (Черкаси, 2019 р.); Міжнародна наукова конференція «Безпека в сучасному світі» (Дніпро, 2019 р.); Всеукраїнська науково-практична Інтернет-конференція «Перспективні напрямки інформаційних і комп’ютерних систем та мереж, комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті» (Кропивницький, 2019 р.); Міжнародна науково-практична конференція «Математичне та програмне забезпечення інтелектуальних систем» (Дніпро, 2019 р.); Всеукраїнська науково-практична конференція «Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем MEICS» (Дніпро, 2019 р.); Міжнародна науково-

практична конференція «Інформаційна безпека та інформаційні технології» (Кропивницький, 2020 р.).

Публікації. Основні положення дисертації опубліковано в 76 наукових працях, у тому числі: 25 наукових статтях (з них 4 проіндексовано у базі даних Scopus; 2 – опубліковані у закордонних рецензованих виданнях; 22 – опубліковані у вітчизняних фахових наукових журналах, з яких 7 статей одноосібні), а також 51 тезах доповідей (з них 1 проіндексовано у базі даних Scopus).

Структура роботи та її обсяг. Дисертація складається із анотації, вступу, шести розділів, загальних висновків, списку використаної літератури та додатків і містить 300 сторінок основного тексту, 42 рисунки, 31 таблицю, 231 джерело у списку літератури та 23 сторінки додатків. Загальний обсяг роботи 323 сторінки.

РОЗДІЛ 1.
ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА МЕТОДІВ СИНТЕЗУ РЕКОМЕНДАЦІЙНИХ СИСТЕМ. ПОСТАНОВКА НАУКОВО-ТЕХНІЧНОЇ ПРОБЛЕМИ

На сьогоднішній день у сфері інформаційних технологій для просування товарів та послуг все частіше використовують так звані рекомендаційні системи – інструменти автоматичної генерації рекомендацій на основі вивчення персональних потреб користувачів веб-сайтів чи додатків [36, 37, 76, 171, 209]. Рекомендаційні системи застосовують, наприклад, такі популярні веб-сайти як Netflix, Amazon, Spotify, YouTube, Facebook тощо.

Основною сферою використання рекомендаційних систем на сьогоднішній день є цифровий маркетинг контенту веб-сайтів, альтернативні методи пошуку даних, а також ранжування пошукової видачі у класичних пошукових системах.

В останні десятиліття кількість веб-сайтів, що застосовують рекомендаційні системи істотно збільшилася. Це пов'язано, перш за все, з розвитком електронної комерції та збільшенням числа Інтернет-магазинів і товарів у них, а також зі збільшенням числа контент-орієнтованих сайтів та появою нового типу сайтів – рекомендаційних мереж. Крім того безперервне збільшення обсягу даних у мережі Інтернет гостро ставить питання оптимізації алгоритмів пошуку за різними критеріями та оптимізації ранжування даних у пошуковій видачі для користувача. Тож використання рекомендаційних систем на веб-сайтах та в мобільних і комп’ютерних додатках стає традиційним на рівні із звичайним пошуком даних.

В той же час рекомендаційні системи вразливі до ряду зовнішніх та внутрішніх дестабілізуючих факторів [9, 34, 40, 46, 47, 76, 147, 157, 163, 176], зокрема, проблеми холодного старту, проблеми недостатньої кількості та якості вхідних даних, інформаційних атак. Треті особи можуть за допомогою мережі ботів впливати на вміст списків рекомендацій користувачів веб-сайтів

з метою просування свого контенту в обхід інтересів користувачів та власників рекомендаційної системи.

У даному розділі розглянуто основну термінологію в області рекомендаційних систем, проведено дослідження та здійснено класифікацію існуючих моделей та методів їх синтезу, притаманних їм проблем та вразливостей, а також загроз у сфері інформаційної безпеки.

На основі проведеного дослідження робиться висновок про актуальність науково-практичної проблеми підвищення точності пропозицій рекомендаційних систем в умовах дестабілізуючих факторів у комп'ютерних мережах на основі розробки моделей та методів синтезу підсистеми забезпечення стійкості.

1.1. Дослідження моделей та методів синтезу рекомендаційних систем

Під рекомендаційними системами будемо розуміти програмне забезпечення, що використовується для прогнозування того, які об'єкти (товари, послуги, публікації, фільми, новини тощо) будуть цікаві користувачу у майбутньому, на основі зібраної раніше інформації про нього та його дії у системі (веб-сайті чи додатку). Елементами рекомендаційної системи будемо називати сукупність користувачів та об'єктів системи.

Перш ніж перейти до розгляду моделей та методів синтезу рекомендаційних систем, проведемо їх класифікацію за призначенням та способами збору даних про користувачів.

Основні сфери застосування рекомендаційних систем на сьогоднішній день можна розділити на наступні [4, 9, 19, 92, 76, 102, 194, 209]:

1. *Рекомендаційні системи в електронній комерції* – виконують завдання по збільшенню відсотка продажів товарів та послуг в Інтернет-магазинах. Крім того, призначення рекомендаційної системи в електронній комерції – скоротити час пошуку потрібних товарів та послуг відвідувачам веб-сайту.

2. *Рекомендаційні системи на контент-орієнтованих веб-сайтах та в соціальних мережах* – покликані збільшувати час перебування відвідувача на сайті, глибину перегляду і залученість, полегшують пошук та доступ користувача до потрібної йому інформації.

3. *Рекомендаційні системи в пошукових роботах* – застосовуються для підвищення релевантності та покращення ранжування пошукової видачі для користувача.

За типом одержувача рекомендацій рекомендаційні системи можна класифікувати на [76, 209]:

1. *Індивідуальні рекомендаційні системи* – для кожного користувача системи формується окремий список рекомендацій.

2. *Групові рекомендаційні системи* – список рекомендацій формується не для окремого користувача, а для певної групи користувачів. Розбиття користувачів на групи може відбуватися за різними критеріями.

Рекомендаційні системи формують свої пропозиції користувачам на основі зібраної про них інформації. Інформацію про поточні вподобання користувачів можна збирати по-різному. За методами збору даних про користувачів рекомендаційні системи можна класифікувати на [76, 186, 209]:

1. *Системи з явним збором даних.* Користувач добровільно надає системі необхідні для її роботи дані. Явний збір даних може полягати в проханні користувачу поставити диференційовану оцінку тому чи іншому об’єкту, створити список «улюблених» об’єктів або заповнити анкету, пройти тест. Основний недолік методу – користувачу необхідно здійснювати деякий набір дій, в чому він може бути незацікавленим або не мати часу на їх виконання.

2. *Системи з неявним збором даних.* Відбувається за допомогою спостереження за поведінкою користувача – оцінками, покупками, переходами за посиланнями тощо. Перевагою методу є те, що для збору даних користувачу не треба здійснювати ніяких додаткових дій, все, що йому потрібно – просто користуватися веб-сайтом в своєму звичайному режимі.

Але даний метод підіймає ряд етичних питань, наприклад, таких як приватність даних.

3. *Системи, що поєднують обидва типи збору даних.* Частина даних одержується явно від користувача, а інша частина збирається непомітно для нього.

На сьогоднішній день найбільш поширеними є рекомендаційні системи з неявним збором даних про користувачів, що формують індивідуальні рекомендації та використовуються переважно на контент-орієнтованих веб-сайтах, в Інтернет-магазинах та віртуальних соціальних мережах. Користувачів важко мотивувати надавати дані про свої вподобання у явній формі, тож набагато легше їх зібрати неявно, і таких даних, що можна використати для створення рекомендацій найбільше на веб-сайтах великих Інтернет-магазинів та популярних соціальних мереж, чим вони і користуються, переважно у маркетингових цілях.

Розглянемо основні моделі та методи роботи рекомендаційних систем, що існують на сьогоднішній день.

Після дослідження робіт [15, 17-19, 31, 35-37, 50, 72, 76, 78, 81, 82, 94, 100, 101, 137, 138, 186, 209, 224], було виявлено, що існують наступні моделі роботи рекомендаційних систем:

– *Моделі сусідства* (Neighborhood models) – досить поширені моделі, що використовують коефіцієнти подоби між елементами системи (користувачами, об’єктами) для того, щоб формувати рекомендації на основі ступеню схожості елементів системи.

– *Матричні факторизаційні моделі* (Matrix factorization models) – також популярні моделі, які використовують для формування рекомендацій приховані фактори, які одержують шляхом факторизації матриці рейтингів та/або матриці даних профілю користувача.

– *Моделі на основі класифікації та кластеризації даних* – такі моделі розділяють дані на кластери на основі різних статистичних методів, методів кластеризації або методів машинного навчання, використовуються у

системах, де є багато інформації про елементи системи, які можна представити у вигляді векторів ознак.

– *Моделі на основі знань* (Knowledge-based models) – рідковикористовувані моделі, що застосовують знання про предметну область, для якої розробляється рекомендаційна система та дозволяють формувати набори правил та базу знань, на основі яких розробляються інтелектуальні чи експертні системи.

– *Моделі, засновані на знаннях про соціальні зв'язки* (community-based models) – рідковикористовувані моделі, що застосовують знання про соціальні зв'язки користувачів для формування рекомендацій.

– *Моделі зміни вподобань користувачів у часі* – моделі, що дозволяють прогнозувати періодичні та неперіодичні зміни у вподобаннях користувачів з часом, як правило, є доповненням до попередніх моделей, зокрема, існують матричні факторизаційні моделі, що враховують фактор часу.

Моделі робастних рекомендаційних систем – передбачають наявність підсистеми інформаційної безпеки, така підсистема може мати деякі або усі з наступних функцій: виявлення атак та ідентифікація профілів ботів, рандомізація списків рекомендацій для зашумлення даних з метою захисту приватності вподобань користувачів, виявлення та видалення зі списку рекомендацій або маркування ризикованих об'єктів. Є доповненням до попередніх моделей, будь-яку рекомендаційну систему можна доповнити підсистемою інформаційної безпеки.

Дослідження робіт [15, 17-19, 31, 35-37, 50, 72, 76, 78, 81, 82, 94, 100, 101, 137, 138, 186, 209, 224] показало, що існує велика кількість різних методів роботи рекомендаційних систем, а саме:

– *Методи колаборативної фільтрації (collaborative filtering)* – застосовують інформацію про рейтинги, які користувачі виставляють об'єктам, та засновані на визначенні схожості користувачів чи об'єктів. Дані методи, як правило, застосовують моделі сусідства або матричні факторизаційні моделі. Можуть використовувати моделі, засновані на

знаннях.

– *Методи контентної фільтрації (content-based filtering)* – застосовують описи об'єктів та дані з профілю користувача, часто засновані на комп'ютерній лінгвістиці. Використовують моделі класифікації та кластеризації даних. Можуть використовувати моделі, засновані на знаннях.

– *Методи фільтрації, заснованої на знаннях про предметну область (knowledge-based filtering)* – використовують знання про поведінку користувачів у системі, для якої розробляється рекомендаційна система, використовують базу знань для формувань рекомендацій. Використовують моделі, засновані на знаннях.

– *Методи фільтрації, заснованої на знаннях про соціальні зв'язки користувачів (community-based filtering)* – можуть застосовуватися, якщо на веб-сайті чи в додатку є елементи соціальної мережі, які дозволяють отримати граф соціальних зв'язків користувачів, в такому разі, при побудові рекомендацій користувачу, буде враховуватися інформація про вподобання його друзів. Використовують моделі, засновані на знаннях про соціальні зв'язки.

– *Методи контекстної фільтрації, зокрема, на основі демографічної інформації (context-based filtering)* – формування рекомендацій відбувається на основі контекстної інформації, наприклад, демографічних даних про користувача. Як правило, використовуються як доповнення до інших методів, наприклад, методів колаборативної чи контентної фільтрації, зокрема, для подолання проблеми холодного старту. Можуть використовувати моделі класифікації та кластеризації даних та моделі на основі знань.

– *Гібридні методи* – поєднують у собі декілька методів та/або моделей роботи рекомендаційних систем.

Розглянемо найбільш поширені комбінації існуючих на сьогоднішній день моделей та методів синтезу рекомендаційних систем для контент-орієнтованих веб-сайтів та віртуальних соціальних мереж.

1.1.1. Дослідження моделей рекомендаційних систем на основі сусідства для методів колаборативної фільтрації

Методи колаборативної фільтрації, що використовують моделі сусідства, здійснюють прогнозування вподобань користувачів на основі ступеню сусідства між елементами рекомендаційної системи [36, 37, 76, 171, 209]. Ступінь сусідства визначається за допомогою коефіцієнтів подоби, обчислення яких здійснюється на основі різних параметрів та метрик.

Тож першим кроком методів, заснованих на колаборативній фільтрації, є обчислення *коефіцієнтів подоби* користувачів та/або об’єктів системи, обчислення яких здійснюються найчастіше на основі оцінок, які користувачі виставляють об’єктам. Крім оцінок можуть використовуватися дані про здійснені перегляди, поставлені теги, написані коментарі тощо.

У найбільш простому випадку збираються дані про поставлені користувачами оцінки та записуються у матрицю рейтингів (рис. 1.1).

	Об’єкт 1	**Об’єкт 2**	**…**	**Об’єкт n**
Користувач 1	5	3	…	3
Користувач 2	4	–	…	2
…	…	…	…	…
Користувач m	–	4	…	4

Рис. 1.1. Матриця рейтингів

В матриці рейтингів значення є оцінками конкретного користувача для конкретного об’єкту. Відсутні значення в таблиці рейтингів є невідомими, тобто, для певного об’єкту певний користувач не виставив оцінку.

Розглянемо метрики, що можуть використовуватися для визначення коефіцієнтів подоби у колаборативній фільтрації.

Якщо об’єкт (або користувач), що описується m ознаками, представити точкою у k-мірному просторі, то подібність об'єктів один з одним буде

визначатися як відстань в даному метричному просторі. У випадку з матрицею рейтингів таке представлення можливе – ознаками будуть в такому разі оцінки об’єктам. Найбільш поширені метрики подоби, що використовуються в такому випадку: евклідова відстань (1.1), зважена евклідова відстань (1.2), відстань Хемінга (Манхеттенська відстань) (1.3), відстань Мінковського (1.4), відстань Махаланобіса (1.5), кореляція Пірсона (1.6), косинусна подоба (1.7):

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{m} (x_{1i} - x_{2i})^2}, \tag{1.1}$$

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{m} w_i (x_{1i} - x_{2i})^2}, \tag{1.2}$$

$$d(x_1, x_2) = \sum_{i=1}^{m} |x_{1i} - x_{2i}|, \tag{1.3}$$

$$d(x_1, x_2) = \left(\sum_{i=1}^{m} |x_{1i} - x_{2i}|^p \right)^{1/p}, \tag{1.4}$$

$$d(x_1, x_2) = \sqrt{(\overline{X}_1 - \overline{X}_2)^{\delta} \sum^{-1} (\overline{X}_1 - \overline{X}_2)}, \tag{1.5}$$

$$d(x_1, x_2) = \frac{\sum_{i=1}^{m} (x_{1i} - \overline{x}_1)(x_{2i} - \overline{x}_2)}{\sqrt{\sum_{i=1}^{m} (x_{1i} - \overline{x}_1)^2} \sqrt{\sum_{i=1}^{m} (x_{2i} - \overline{x}_2)^2}}, \tag{1.6}$$

$$d(x_1, x_2) = \frac{\overline{X}_1 \cdot \overline{X}_2}{\|\overline{X}_1\| \cdot \|\overline{X}_2\|} = \frac{\sum_{i=1}^{m} x_{1i} \cdot x_{2i}}{\sqrt{\sum_{i=1}^{m} (x_{1i})^2} \sqrt{\sum_{i=1}^{m} (x_{2i})^2}}, \tag{1.7}$$

де $d(x_1, x_2)$ – відстань між об'єктами x_1 та x_2; x_{1i}, x_{2i} – значення i-ї ознаки відповідно у 1-го та 2-го об'єкту; w_i – вага, що привласнюється i-ій змінній; $\sum^{-1}$ – матриця зворотна коваріаційній матриці, розрахованій по всій вибірці; $\overline{X}_1$, $\overline{X}_2$ – вектори значень ознак у 1-го та 2-го об'єкту; $\overline{x}_1$, $\overline{x}_2$ – середні значення ознак відповідно у 1-го та 2-го об'єкту; m – кількість ознак.

Оцінювати схожість об'єктів (або користувачів) за допомогою мір відстані зручно при використанні числових ознак. Але часто зустрічаються ознаки, що виміряються в інших шкалах (наприклад, в ранговій або номінальній). І це може трапитися, якщо використовувати в колаборативній фільтрації не оцінки, а інші відомості про дії користувача. В цьому випадку всі ознаки, які використовуються для класифікації, представляються у вигляді двійкового коду. Тобто, кожен об'єкт описується вектором $\overline{X}_i = (x_{i1}, x_{i2}, ..., x_{im})$, де $i = \overline{1,n}$, кожна з компонент якого приймає значення 0 або 1. Для визначення подоби *i*-го та *j*-го об'єктів в такому випадку найбільш часто застосовують наступні метрики: коефіцієнт Рао (1.8), коефіцієнт Хаммана (1.9), коефіцієнт Роджерса та Танімото (1.10), коефіцієнт Джекарда (1.11), коефіцієнт Дейка (1.12), коефіцієнт композиційної подоби (1.13):

$$S_{ij} = \frac{n_{ij}^{(1,1)}}{m}, (0 \le S_{ij} \le 1), \tag{1.8}$$

$$S_{ij} = \frac{p_{ij} - q_{ij}}{m}, (p_{ij} = q_{ij} \to S_{ij} = 0), \tag{1.9}$$

$$S_{ij} = \frac{n_{ij}^{(1,1)}}{n_i^{(1)} + n_j^{(1)} + n_{ij}^{(1,1)}}, (0 \le S_{ij} \le 1), \tag{1.10}$$

$$S_{ij} = \frac{n_{ij}^{(1,1)}}{n_{ij}^{(1,1)} + q_{ij}}, (0 \le S_{ij} \le 1), \tag{1.11}$$

$$S_{ij} = \frac{2n_{ij}^{(1,1)}}{2n_{ij}^{(1,1)} + q_{ij}}, (0 \le S_{ij} \le 1), \tag{1.12}$$

$$S_{ij} = \frac{p_{ij}}{2m - p_{ij}} = \frac{p_{ij}}{m - q_{ij}}, (0 \le S_{ij} \le 1), \tag{1.13}$$

де $p_{ij} = n_{ij}^{(1,1)} + n_{ij}^{(0,0)}$ – загальна кількість співпадаючих ознак; $q_{ij} = n_{ij}^{(0,1)} + n_{ij}^{(1,0)}$ – загальна кількість неспівпадаючих ознак; $n_{ij}^{(1,1)}$– число співпадаючих одиничних ознак у обох пар об'єктів (пар (1,1)); $n_{ij}^{(0,0)}$– число співпадаючих нульових ознак у обох пар об'єктів (пар (0,0)); $n_{ij}^{(1,0)}$ – кількість співпадаючих

одиничних ознак у *i*-го та нульових ознак у *j*-го об'єктів (пар (1,0)); $n_{ij}^{(0,1)}$ – кількість співпадаючих нульових ознак у *i*-го та одиничних ознак у *j*-го об'єктів (пар (0,1)); $n_i^{(1)}$, $n_j^{(1)}$ – число одиничних ознак у *i*-го та одиничних ознак у *j*-го об'єктів відповідно; *m* – загальна кількість ознак, за якими здійснюється порівняння.

Методи колаборативної фільтрації на основі моделі сусідства поділяється на два види [76, 78, 205, 209]:

1. Засновані на схожості користувачів (User/User, User-based).
2. Засновані на схожості об’єктів (Item/Item, Item-based).

Колаборативна фільтрація, заснована на схожості користувачів (user-based). Розглянемо найпростіший спосіб реалізації колаборативної фільтрації, заснованої на схожості користувачів. Всі інші способи є ускладненнями даного.

Після того, як для рекомендаційної системи зібрані дані та побудована матриця користувачів-об’єктів, для формування рекомендацій певному користувачу необхідно здійснити наступну послідовність дій:

1. Обчислити множину коефіцієнтів подоби даного користувача з усіма іншими користувачами (1.14):

$$K_i = \{k_{i,1}, k_{i,2}, \ldots, k_{i,j}\}\ , \tag{1.14}$$

де $k_{i,j}$ – коефіцієнт подоби між *i*-тим та *j*-тим користувачами.

2. Ранжувати користувачів відносно *i*-того користувача за ступенем подоби на нього. Обрати TopN найбільш схожих на нього користувачів або усіх користувачів, для яких вдалося обчислити коефіцієнт подоби.

3. Обчислити для об’єктів системи, які ще не переглядав (не оцінював) *i*-тий користувач, зважену суму оцінок інших користувачів (1.15):

$$S_{i,q} = \sum_{j=1}^{n} r_{q,j} \cdot k_{i,j}\ , \tag{1.15}$$

де $r_{q,j}$ – оцінка *q*-того об’єкту, поставлена *j*-тим користувачем; *n* – довжина списку TopN схожих на *i*-того користувача користувачів.

4. Розділити кожну зважену суму на суму коефіцієнтів подоби всіх користувачів, що ставили оцінки відповідному об'єкту (1.16), щоб об'єкти, що отримали більше оцінок не одержали перевагу:

$$\widetilde{r}_{i,q}=\frac{S_{i,q}}{\sum\limits_{j=1}^{n}k_{i,j}}. \qquad (1.16)$$

Рівняння (1.16) дає прогноз оцінки *i*-того користувача *q*-тому об'єкту.

5. Відсортувати множину об'єктів, одержану після третього кроку на основі значень прогнозованих оцінок та рекомендувати користувачу перші *N* об'єктів у списку.

В реальних системах для кожної рекомендації неможливо використовувати в обчисленнях дані всіх сотень тисяч чи навіть мільйонів користувачів системи, тому у формулах використовуються *K* найближчих сусідів – користувачів максимально схожих на користувача, якому обчислюються рекомендації (згаданий вище TopN).

Один з варіантів цієї групи методів, що має назву алгоритм GroupLens [75], замість кроків 3 та 4 розглянутих вище, для прогнозування оцінки використовує наступну формулу:

$$\widetilde{r}_{i,q}=\overline{r}_i+\frac{\sum\limits_{j=1}^{n}\left(r_{j,q}-\overline{r}_j\right)\cdot k_{i,j}}{\sum\limits_{j=1}^{n}\left|k_{i,j}\right|}, \qquad (1.17)$$

де $\overline{r}_i$ – середня оцінка *i*-того користувача; $\overline{r}_j$ – середня оцінка *j*-того користувача.

Методи колаборативної фільтрації засновані на схожості користувачів потребують постійних перерахунків коефіцієнтів подоби, так як дані про дії користувачів постійно оновлюються і їх вподобання з часом змінюються.

Колаборативна фільтрація, заснована на схожості об'єктів (item-based). Основна ідея даних методів полягає у тому, щоб для кожного об'єкту заздалегідь визначити множину схожих на нього об'єктів. Тоді для формування рекомендацій певному користувачу достатньо буде знайти ті

об’єкти, яким він поставив найбільші оцінки, та створити зважений список N об’єктів, максимально схожих на них. Результати порівняння об’єктів змінюються не так часто, як результати порівняння користувачів. Тож на першому кроці необхідно дослідити всі наявні дані, а подальші перерахунки можна робити рідко, вибираючи моменти часу, коли навантаження на веб-сайт мінімальне.

Методи колаборативної фільтрації засновані на схожості об’єктів працюють швидше, ніж методи засновані на схожості користувачів, так як багато обчислень можна здійснити заздалегідь. Тож їх можна з успіхом використовувати на великих об’ємах даних.

Для прогнозування оцінки на основі даного підходу треба знайти зважене середнє для оцінених користувачем об’єктів:

$$\widetilde{r}_{i,q}=\overline{r}_i+\frac{\sum\limits_{p=1}^{n}\left(r_{j,p}-\overline{r}_i\right)\cdot k_{q,p}}{\sum\limits_{p=1}^{n}\left|k_{q,p}\right|}, \qquad (1.18)$$

де $k_{q,p}$ – коефіцієнт кореляції між q-тим об’єктом та p-тим об’єктом; n – довжина списку схожих на q-тий об’єкт об’єктів.

Використовуючи методи колаборативної фільтрації засновані на схожості об’єктів можна рідше перераховувати коефіцієнти подоби, ніж в методах заснованих на схожості користувачів, адже коефіцієнти подоби для об’єктів змінюються рідше, ніж для користувачів. Також Item-based методи мають більшу стійкість до інформаційних атак, ніж User-based методи.

Методи колаборативної фільтрації на основі моделі сусідства не потребують використання складних ресурсозатратних алгоритмів, показують високу точність прогнозування вподобань для користувачів, які уже виставили певну, мінімально необхідну, кількість оцінок об’єктам системи. Чим більше оцінок користувач виставить об’єктам системи, тим точніше ці методи сформують для нього списки рекомендацій. Недоліками цих методів є вразливість до проблеми холодного старту та інформаційних атак.

1.1.2. Дослідження матричних факторизаційних моделей рекомендаційних систем для методів колаборативної фільтрації

Методи колаборативної фільтрації, що використовують матричні факторизаційні моделі, здійснюють прогнозування вподобань користувачів на основі прихованих факторів елементів системи, що впливають на інтереси користувачів та можуть бути одержані за допомогою факторизації матриці рейтингів або змішаної матриці, що містить рейтинги та ознаки елементів системи [19, 27, 35, 41, 76, 173].

Приховані фактори, виявлені за допомогою факторизації, дозволяють заповнити відсутні у матриці рейтингів комірки, тобто, прогнозувати відсутні рейтинги.

Факторизація – це процес декомпозиції об'єкту (зокрема, матриці) в набір інших об'єктів (факторів), добуток яких дає початковий об'єкт [44]. Факторизація дозволяє виділити ключові компоненти об'єкту факторизації.

Основна ідея факторизації матриці рейтингів рекомендаційної системи зображена на рис. 1.2.

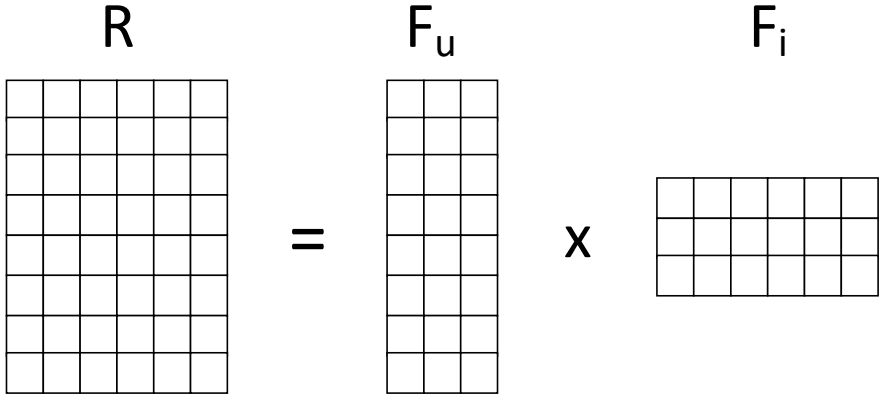


Рис. 1.2. Принцип факторизації матриці рейтингів

Матриця рейтингів R розмірності $n \times m$, де n – кількість користувачів, m – кількість об'єктів, факторизується на дві матриці: матрицю прихованих факторів користувачів F_u, розмірності $n \times k$, де k – кількість прихованих факторів, та матрицю прихованих факторів об'єктів F_i, розмірності $k \times m$.

Загальний алгоритм факторизації матриці рейтингів виглядає наступним чином:

1. Ініціалізуємо матриці F_u та F_i випадковими значеннями.

2. Перемножуємо матриці F_u та F_i і порівнюємо результат з R, обчислюємо помилки для кожної комірки та загальну помилку.

3. Мінімізуємо помилки за допомогою деякого алгоритму машинного навчання (наприклад, градієнтного спуску, методу найменших квадратів тощо), поки загальна помилка не знизиться до прийнятного значення.

Найбільш відомими методами колаборативної фільтрації, що застосовують матричну факторизаційну модель є FunkSVD, SVD++, Asymmetric SVD, timeSVD [19, 27, 35, 41, 42, 76, 87]. Усі ці моделі одержали назву від методу факторизації матриць Singular value decomposition (сингулярний розклад матриць), хоча безпосередньо його вони не використовують, а лише засновані на спільній з ним ідеї – одержати для певної матриці деякі матриці, добуток яких дасть матрицю наближену до початкової. У випадку з матрицями рейтингів, ця одержана наближена матриця буде містити наближені дані у відомих рейтингах, а в комірках, де в початковій матриці рейтинги були невідомі, з’являться прогнозовані рейтинги.

Перший метод з цієї групи методів був запропонований Сімоном Фанком під час конкурсу від Netflix у 2006. У своїй публікації у власному блозі [27], Фанк визначив матрицю рейтингів як добуток двох матриць пониженого рангу, перша – рядки прихованих факторів для користувачів, а друга – стовпчики прихованих факторів для об’єктів. Множення рядка користувача на стовпчик об’єкта дає прогнозований рейтинг для відповідної пари користувач-об’єкт.

Усі наступні методи, засновані на матричних факторизаційнійних моделях є покращеними модифікаціями моделі FunkSVD. Розглянемо всі ці методи детальніше.

FunkSVD. Найперший метод роботи рекомендаційних систем, що

застосовує матричну факторизацію [27].

Даний метод полягає у наступному. Спочатку треба визначити базові предиктори (зміщення) $b_{u,i}$, які складаються з базових предикторів окремих користувачів b_u і базових предикторів окремих об'єктів b_i, а також просто загального середнього рейтингу об'єктів у системі μ:

$$b_{u,i} = \mu + b_u + b_i. \tag{1.19}$$

Прогнозування оцінки для пари користувач-об'єкт здійснюється за наступною формулою:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i \cdot p_u. \tag{1.20}$$

де q_i – вектор факторів об'єкту *i*, а p_u – вектор факторів користувача *u*.

На початку роботи алгоритму треба обчислити глобальну середню оцінку та усі предиктори. Потім треба знайти найкращі предиктори та фактори, що дозволяють прогнозувати рейтинги з найменшою помилкою.

Для визначення помилки використовується сума квадратів відхилень:

$$E = \sum_{(u,i)\in D} (r_{u,i} - \hat{r}_{u,i})^2, \tag{1.21}$$

$$E = \sum_{(u,i)\in D} (r_{u,i} - \mu - b_u - b_i - q_i \cdot p_u)^2, \tag{1.22}$$

де $r_{u,i}$ – справжній рейтинг об'єкту *i* у користувача *u*; $\hat{r}_{u,i}$ – прогнозований рейтинг.

Дана функція оптимізується градієнтним спуском, беруться часткові похідні по кожному аргументу, а рух під час градієнтного спуску відбувається у сторону, зворотню напрямку цих похідних. Для одержання адекватних результатів при роботі з реальними даними необхідно враховувати ймовірність оверфітінгу [14, 59, 71] (перенавчання системи) та виконувати регуляризацію [14, 67], щоб подолати дану проблему.

Регуляризація – додавання деякої додаткової інформації, щоб знайти рішення некоректно поставленої задачі, або щоб уникнути перенавчання.

Загалом регуляризуючий вираз $R(f)$ додається до значення помилки,

перед тим як визначати аргумент, що дає найменше значення помилки:

$$f_* = \arg\min_f \sum_{i=1}^{n} E\left(f\left(\hat{x}_i\right), \hat{y}_i\right) + \lambda \cdot R\left(f\right), \tag{1.23}$$

де E – функція, що визначає похибку передбачення $f(x)$ для значень y, а параметр λ визначає важливість доданка для регуляризації. Зазвичай $R(f)$ визначається як штраф за складність функції f. Зокрема, поняття складності включає обмеження на гладкість та на норму векторного простору.

В FunkSVD оптимізаційний вираз можна записати наступним чином:

$$\begin{gathered} b_*, q_*, p_* = \arg\min_{b,q,p} \sum_{(u,i)}^{n} (r_{u,i} - \mu - b_u - b_i - q_i \cdot p_u)^2 + \\ + \lambda\left(\sum_u b_u^2 + \sum_i b_i^2 + \|q_i\|^2 + \|p_u\|^2 \right) \end{gathered}, \tag{1.24}$$

де λ – параметр регуляризації.

Якщо взяти від (1.24) часткові похідні по кожній із змінних, що оптимізуються, отримаємо прості правила для градієнтного спуску.

Під час градієнтного спуску у FunkSVD використовуються наступні правила для оптимізації змінних, що впливають на результат:

$$b_u = b_u + \gamma\left(e_{u,i} - \lambda b_u\right), \tag{1.25}$$

$$b_i = b_i + \gamma\left(e_{u,i} - \lambda b_i\right), \tag{1.26}$$

$$q_{i,k} = q_{i,k} + \gamma\left(e_{u,i} \cdot p_{u,k} - \lambda q_{i,k}\right), \tag{1.27}$$

$$p_{u,k} = p_{u,k} + \gamma\left(e_{u,i} \cdot q_{i,k} - \lambda p_{u,k}\right), \tag{1.28}$$

де $e_{u,i} = r_{u,i} - \hat{r}_{u,i}$ – помилка на навчальному наборі; γ – швидкість навчання.

Можна очікувати більшої точності, виділивши окремі швидкості навчання γ_n і регуляризації λ_n для кожного типу досліджуваного параметра. Так, наприклад, рекомендується використовувати різні швидкості навчання для зсувів користувачів, зсувів об’єктів і самих факторів.

Важливо, що при такому підході невідомо, які саме характеристики об’єктів відповідають факторам. Тому дані моделі є неінтерпретуємими.

SVD++. Відрізняється від FunkSVD тим, що крім рейтингів (явного

зворотного зв’язку від користувача) використовує також неявну інформацію про вподобання користувачів, наприклад, перегляди об’єктів, написання коментарів тощо [19, 35, 76].

Точність прогнозування у SVD++ поліпшується за рахунок врахування неявного зворотного зв'язку, який забезпечує додаткову індикацію вподобань користувачів. Це особливо корисно для тих користувачів, які надали більше неявного зворотного зв’язку, ніж явного.

Для врахування неявного зворотного зв’язку (одного типу) від користувачів використовується другий набір факторів об’єктів, що пов’язує кожен об’єкт з вектором факторів $y_i \in R(u)$. Ці нові фактори об’єктів використовуються для характеристики користувачів на основі набору об’єктів, які вони неявно оцінили. У такій моделі рейтинги прогнозуються наступним чином:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i \cdot \left(p_u + \left|R(u)\right|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j \right). \tag{1.29}$$

Набір $R(u)$ містить усі об’єкти, що були неявно оцінені користувачем u.

Характеристики користувача u моделюються як $p_u + \left|R(u)\right|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j$. Оскільки y_j – центровані навколо нуля (регуляризацією), сума нормалізується на $\left|R(u)\right|^{-\frac{1}{2}}$, щоб стабілізувати її дисперсію в межах діапазону спостережуваних значень $\left|R(u)\right|$.

Параметри моделі визначаються шляхом мінімізації відповідної регуляризованої квадратичної функції помилок за допомогою стохастичного градієнтного спуску. Оптимізація змінних, що впливають на результат прогнозу, обчислюється наступним чином:

$$b_u = b_u + \gamma\left(e_{u,i} - \lambda_1 b_u\right), \tag{1.30}$$

$$b_i = b_i + \gamma\left(e_{u,i} - \lambda_1 b_i\right), \tag{1.31}$$

$$q_{i,k} = q_{i,k} + \gamma\left(e_{u,i} \cdot \left(p_{u,k} + \left|R(u)\right|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j \right) - \lambda_2 q_{i,k} \right), \tag{1.32}$$

$$p_{u,k} = p_{u,k} + \gamma\left(e_{u,i} \cdot q_{i,k} - \lambda_2 p_{u,k}\right), \tag{1.33}$$

$$\forall j \in R(u) : y_j \leftarrow y_j + \gamma(e_{u,i} \cdot \left|R(u)\right|^{-\frac{1}{2}} q_{i,k} - \lambda_2 y_j), \tag{1.34}$$

Можна враховувати і декілька типів неявного зворотного зв'язку. Наприклад, якщо користувач u має два типи неявного оцінювання об'єктів: додавання в обране $N_1(u)$ та перегляд сторінки $N_2(u)$, тоді рейтинги можна прогнозувати наступним чином:

$$\hat{r}_{u,i} = \mu + b_u + b_i + q_i \cdot \\ \cdot \left(p_u + \left|N_1(u)\right|^{-\frac{1}{2}} \sum_{j \in N_1(u)} y_{1j} + \left|N_2(u)\right|^{-\frac{1}{2}} \sum_{j \in N_2(u)} y_{2j} \right). \tag{1.35}$$

Відносна важливість кожного джерела неявного зворотного зв'язку буде автоматично визначена алгоритмом шляхом встановлення відповідних значень параметрів моделі.

Asymmetric SVD. Асиметричний SVD дозволяє додавати до моделі нових користувачів з декількома рейтингами, без необхідності перенавчати всю модель [41, 76, 87]. При додаванні нового користувача, його приховані фактори обчислюються наступним чином:

$$p_{u,k} = \sigma\left(b + \sum_{j \in R(u)} w_{r_{uj}} s_j \right), \tag{1.36}$$

де σ – сигмоїдна функція: $\sigma(x) = \frac{1}{1+e^{-x}}$; r_{uj} – рейтинг, який користувач u поставив об'єкту j; $R(u)$ – множина об'єктів, оцінених користувачем u, з відомими рейтингами; w, b, s – параметри, що шукаються градієнтним спуском.

Якщо необхідно використовувати неявні зворотні зв'язки, тоді для Asymmetric SVD можна застосувати наступну формулу:

$$p_{u,k} = \left|R(u)\right|^{-\frac{1}{2}} \sum_{j \in R(u)} r'_{u,j} \cdot s_j, \tag{1.37}$$

де $r'_{u,j} = r_{i,j} - (\mu + b_u + b_i)$; r_{uj} – рейтинг, який користувач u поставив об'єкту j; $R(u)$ – множина об'єктів, оцінених користувачем u, з відомими рейтингами; μ, b, s – параметри, що шукаються градієнтним спуском.

TimeSVD++. Це факторизаційна модель з врахуванням часу [42, 76]. Вподобання користувачів можуть залежати від часу, саме це враховує дана модель. Зміни можуть мати як циклічний, так і не циклічний характер.

Популярність товару може змінюватися з часом. Це можна врахувати, трактуючи зміщення об'єкту b_i як функцію часу. Користувачі з часом можуть змінювати свої вподобання. Це можна також врахувати, прийнявши зміщення користувача b_u як функцію часу. Тоді базові предиктори будуть розраховуватися наступним чином:

$$b_{u,i} = \mu + b_u(t_{u,i}) + b_i(t_{u,i}). \tag{1.38}$$

Тут, $b_u(t_{u,i})$ та $b_i(t_{u,i})$ – реальні функції, які змінюються з часом. Точний спосіб побудови цих функцій повинен відображати розумний спосіб параметризації задіяних часових змін. Наприклад, у випадку рейтингу фільму очікується, що прихильність до фільму щодня трохи коливатиметься, а сильно змінюватиметься протягом більш тривалих періодів. З іншого боку вподобання користувачів можуть змінюватися щодня, відображаючи природню для поведінки клієнтів мінливість. Це вимагає обрання менших проміжків часу при моделюванні поведінки користувача та більших проміжків часу для моделювання часових ефектів, пов'язаних з об'єктами.

Базові предиктори окремих об'єктів можна визначити за наступною формулою:

$$b_i(t) = b_i + b_{i,Bin(t)}, \tag{1.39}$$

де b_i – незмінна у часі частина предиктору об'єкта; $b_{i,Bin(t)}$ – частина предиктору об'єкта, що змінюється у часі, $bin(t)$ – номер часового проміжку, на які поділені усі наявні для навчання системи дані.

Для параметризації часової поведінки користувача з різною складністю та точністю можна розглядати різні функції. Простий спосіб моделювання

використовує лінійну функцію для фіксації можливого поступового зміщення вподобань користувачів. Для кожного користувача u позначаємо середню дату рейтингу за допомогою t_u. Тепер, якщо користувач оцінив фільм у момент часу t, то пов'язане з цим відхилення часу визначається як:

$$dev_u(t) = sign(t - t_u) \cdot |t - t_u|^{\beta}, \tag{1.40}$$

де $|t - t_u|$ вимірює кількість умовних часових одиниць (наприклад, днів) між датами t і t_u. Значення β встановлюється шляхом перехресної перевірки.

Базові предиктори окремих користувачів можна визначити за допомогою формули (1.41) або (1.42).

$$b_i(t) = b_u + \alpha_u \cdot dev_u(t), \tag{1.41}$$

де α_u – параметр алгоритму, який слід визначати під час градієнтного спуску.

Ця проста лінійна модель для наближення поведінки користувача вимагає визначення двох параметрів b_u та α_u для кожного користувача.

Більш гнучку параметризацію пропонують сплайни. Нехай u є користувачем, пов’язаним з n_u рейтингами. Часові точки k_u розподілені рівномірно по датах рейтингів користувача u як ядра, які керують такою функцією:

$$b_i(t) = b_u + \frac{\sum_{l=1}^{k_u} e^{-\sigma|t-t_l^u|} b_{t_l}^u}{\sum_{l=1}^{k_u} e^{-\sigma|t-t_l^u|}}, \tag{1.42}$$

де параметри b_u та t_l асоціюються з контрольними точками (ядрами) і автоматично дізнаються з даних. Таким чином, вподобання користувача формується як зважена в часі комбінація цих параметрів. Кількість ядер k_u врівноважує гнучкість та ефективність обчислень. Постійна σ визначає плавність сплайну.

При використанні формул (1.39-1.41) для визначення базових предикторів, одержимо наступну формулу для запису оптимізаційного виразу:

$$b_{u*}, b_{u,t*}, b_{i*}, b_{i,Bin(t)*}, \alpha_{u*} =$$

$$= \arg\min \sum_{(u,i,t)} \left(r_{u,i} - \mu - b_u - \alpha_u dev_u\left(t_{u,i}\right) - b_{u,t} - b_i - b_{i,Bin(t)}\right)^2 + , \qquad (1.43)$$
$$+ \lambda\left(b_u^2 + \alpha_u^2 + b_{u,t}^2 + b_i^2 + b_{i,Bin(t)}^2\right)$$

Можна використовувати ту саму методологію для отримання більшої кількості часових ефектів. Наприклад, для фіксації періодичних ефектів. Деякі об'єкти можуть бути більш популярними в конкретні пори року або впродовж певних свят. Періодичні ефекти можна знайти і для користувача. Наприклад, користувач може мати різні схеми купівлі протягом вихідних у порівнянні з робочим тижнем. Спосіб моделювання таких періодичних ефектів – виділити параметр для комбінацій періодів часу з об'єктами або користувачами. В такому разі базові предиктори можна розраховувати за наступними формулами:

$$b_i(t) = b_i + b_{i,Bin(t)} + b_{i,period(t)}, \qquad (1.44)$$

$$b_u(t) = b_u + \alpha_u \cdot dev_u(t) + b_{u,t} + b_{u,period(t)}. \qquad (1.45)$$

Приховані фактори користувачів для timeSVD без врахування періодичності, а лише з врахуванням зміщень у часі, можна визначити так:

$$p_{u,k}(t) = p_{u,k} + \alpha_{u,k} \cdot dev_u(t) + p_{u,k,t}, \ k = 1,..., f. \qquad (1.46)$$

Рейтинги у TimeSVD++ можна прогнозувати за наступною формулою:

$$\hat{r}_{u,i} = \mu + b_u(t) + b_i(t) + q_i \cdot \left(p_u(t) + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j \right). \qquad (1.47)$$

Отже, існує багато різних моделей поведінки користувача рекомендаційної системи, заснованих на матричній факторизації для визначення прихованих факторів, що впливають на його вподобання.

Перевагами таких моделей є висока стійкість до інформаційних атак та висока точність прогнозування вподобань. До недоліків слід віднести погану масштабованість, довгий час навчання, а також необхідність перенавчання рекомендаційної системи при появі нових даних, цей останній недолік частково вирішено у Asymmetric SVD.

1.1.3. Дослідження моделей рекомендаційних систем для методів контентної фільтрації

Такі моделі передбачають, що вподобання користувачів визначаються властивостями об’єктів, які вони переглядали раніше [76, 209]. Рекомендаційні системи з фільтрацією на основі змісту (контентною фільтрацією) беруть до уваги схожість об'єктів з інформацією відомою про користувача.

Інформація про користувача може бути отримана з його профілю та/або зібрана з його дій на веб-сайті – написаних відгуків та коментарів, придбаних товарів, переглянутих веб-сторінок тощо. Така інформація може бути зібрана як явними способами (дані з профілю користувачів та сторінок опису об’єктів), так і неявними способами (вилучення ключових слів та фраз [83, 146, 159], а також їх емоційного забарвлення [74, 85, 183, 229, 230] з коментарів та постів користувачів).

Для створення рекомендацій такі системи аналізують інформацію про користувача, формують ключові слова про його інтереси та вподобання. Також формуються ключові слова для об’єктів системи з їх описів, отриманих тегів тощо. Рекомендуватися будуть об’єкти з бази даних, вибрані на основі визначених ключових слів. Тому таку фільтрацію ще можна назвати фільтрацією по ключовим словам.

Методи контентної фільтрації застосовують методи кластеризації, статистичні методи та методи машинного навчання, наприклад, [76, 209]:

1. Класифікатори на основі Байєсівських мереж.
2. Класифікатори на основі нейронних мереж.
3. Класифікатори на основі дерев рішень.
4. Класифікатори на основі алгоритмів кластеризації.

Класифікатори на основі Байєсівських мереж. Одним з найвідоміших класифікаторів є наївний байєсівський класифікатор [59, 116, 209]. В його основі лежить ймовірнісна модель теореми Байєса. Для роботи алгоритмів з використанням даного класифікатора необхідно створити модель Байєса для

кожного користувача, який оцінював будь-які об'єкти, на основі ознак цих об'єктів (для фільмів це можуть бути актори або жанри, для новин – ключові слова тощо). Для знаходження найбільш ймовірної категорії необхідно обчислити умовні ймовірності приналежності будь-якого об’єкту до кожної категорії і вибрати ту, яка має найбільшу ймовірність (1.48):

$$Pr(C \mid O) = \frac{Pr(O \mid C) \cdot Pr(C)}{Pr(O)}, \tag{1.48}$$

$$Pr(O \mid C) = \prod_{i=1}^{n} Pr(W_i \mid C), \tag{1.49}$$

де O – об’єкт (товар, послуга тощо); C – категорія; W – слово (ознака); n – кількість слів (ознак); $Pr(C)$ – повна ймовірність того, що випадково обраний об’єкт потрапляє у категорію C; $Pr(O)$ – повна ймовірність появи об’єкту O; $Pr(W_i|C)$ – умовна ймовірність того, що слово W_i наявне в описі об’єкту O, якщо об’єкт O відноситься до категорії C.

Для кожної категорії можна задати порогові значення. Тоді, щоб новий об’єкт був віднесений до деякої категорії, ймовірність його віднесення до цієї категорії повинна бути більша ймовірності його потрапляння в будь-яку іншу категорію на величину порогового значення.

Перевагою байєсівських класифікаторів є можливість навчання, простота алгоритму навчання, можливість перегляду даних про важливість ознак та одержану в процесі навчання. Основний недолік – неможливість враховувати залежність результату від поєднань ознак.

Класифікатори на основі нейронних мереж. Штучні нейронні мережі можуть мати різну архітектуру та різні алгоритми навчання, але всі вони складаються з нейронів [59, 116, 193, 209], загальна структурна схема яких наведена на рис. 1.3.

На рис. 1.3 використані наступні позначення: $x_1 \ldots x_n$ – входи нейрона (синапси); $w_1 \ldots w_n$ – вагові коефіцієнти входів; S – зважена сума входів нейрона; $F(S)$ – функція активації нейрона; T – порогове значення (значення, після якого нейрон переходить у стан збудження), є не у всіх типів штучних

нейронів; Y – вихід нейрона (аксон).

Зважена сума S обчислюється за наступною формулою:

$$S=\sum_{i=1}^{n} x_i \cdot w_i \ , \tag{1.50}$$

Функція активації $F(S)$ – визначає залежність сигналу на виході нейрона від зваженої суми сигналів на його входах. В якості функції активації можуть використовуватися: лінійна функція, порогова функція, сигмоїдальна функція тощо.

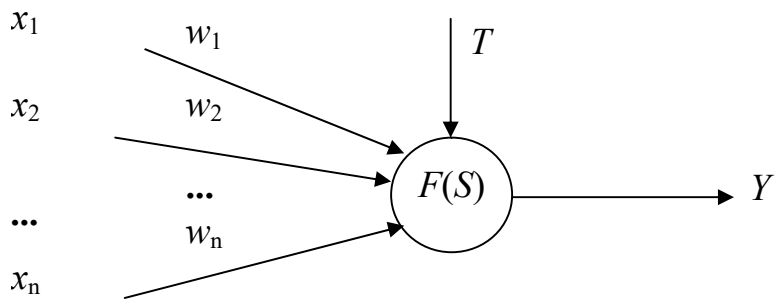


Рис. 1.3. Загальна структурна схема нейрона
у штучній нейронній мережі [59, 116, 193]

В найпростіших випадках для контентної фільтрації можна застосовувати багатошаровий перцептрон, якщо є дані для попереднього навчання і мережу Кохонена, якщо доведеться вчити мережу в процесі використання. Розглянемо загальну структуру перцептрону для використання у контентній фільтрації (рис. 1.4).

Входами нейромережі будуть коди ознак об’єктів, а виходами – коди категорій об’єктів. Нейромережа буде містити декілька прихованих прошарків, кількість прошарків та нейронів у них зазвичай визначаються експериментальним шляхом. Загальною є рекомендація, що кількість нейронів у прихованих прошарках має бути більшою за кількість нейронів у вхідному прошарку.

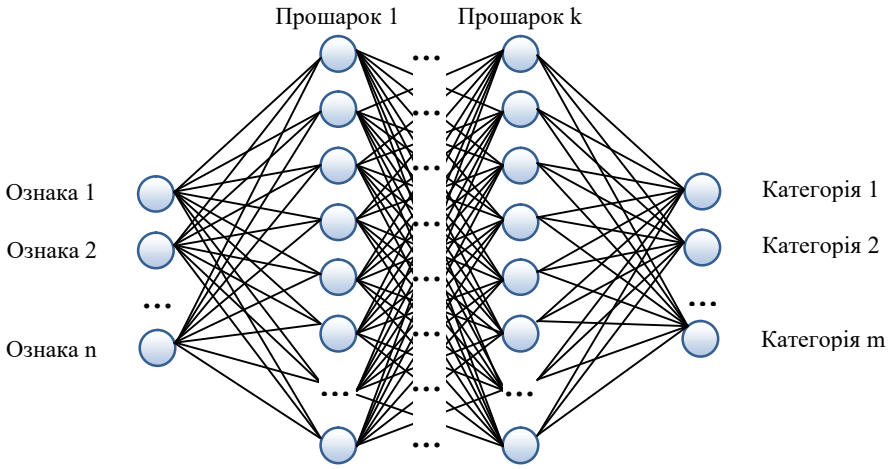


Рис. 1.4. Загальна схема нейронної мережі для класифікації об’єктів у рекомендаційній системі [209]

Зазвичай при побудові нейронної мережі всі вузли створюються заздалегідь. У контентній фільтрації для рекомендаційних систем в [209] пропонується новий прихований вузол створювати тоді, коли зустрічається нова комбінація ознак та створювати для нього зв'язки з вагами за замовчуванням.

Нейронні мережі здатні справлятися зі складними нелінійними функціями та знаходити залежності між різними вхідними даними. Вони допускають адаптивне навчання в процесі використання. Основні недоліки нейронних мереж – вони працюють як чорний ящик, відсутні тверді правила по вибору структури та розміру мережі, швидкості навчання.

Класифікатори на основі дерев рішень. Дерево рішень – це модель, яка являє собою сукупність правил для прийняття рішення [80, 116, 209], в даному разі – рішення, до якої категорії віднести об’єкт. Графічно таку модель можна уявити у вигляді дерева, де вузли – умови переходу, а листки – назви категорій. Якщо для даного об’єкту умова у вузлі істина то здійснюється перехід по лівому ребру, якщо ж ні, то по правому. Залежно від рішення, прийнятого у вузлах, об’єкт відноситься до певної категорії.

Метод дерев рішень реалізує *принцип рекурсивного поділу*. Ця стратегія

також називається «розділяй і володарюй». У вузлах, починаючи з кореневого, вибирається ознака, значення якої використовується для розбиття всіх даних на 2 класи. Процес триває до тих пір, поки не виконається критерій зупинки. Це можливо в таких ситуаціях:

– Всі (або майже всі) дані даного вузла належать одному і тому ж класу.

– Не залишилося ознак, за якими можна побудувати нове розбиття.

– Дерево перевищило заздалегідь заданий «ліміт зростання» (якщо ліміт було встановлено).

Існують різні чисельні алгоритми побудови дерев рішень. Одним з найбільш відомих є алгоритм під назвою C5.0 [16], розроблений програмістом Джоном Квінланом. Фактично алгоритм C5.0 є стандартом процедури побудови дерев рішень. Його програмна реалізація поширюється на комерційній основі, але версія, вбудована в мову програмування для статистичної обробки даних R, доступна безкоштовно.

Дерева рішень корисні не тільки для класифікації, а також і для інтерпретації результатів. На відміну від байєсівського класифікатору вони легко справляються з взаємозалежними ознаками. Але дерева рішень не підтримують адаптивне навчання в процесі використання.

Класифікатори на основі алгоритмів кластеризації. Кластеризація (кластерний аналіз) – це задача розбиття множини об'єктів на групи, які називаються кластерами [112, 115, 116, 209, 214]. Методи кластерного аналізу діляться на: ієрархічні та ітеративні. Ієрархічні в свою чергу поділяються на агломеративні і дивізимні.

Ієрархічні агломеративні методи послідовно об'єднують окремі об'єкти в кластери. Ієрархічні дивізимні методи кластеризації полягають, навпаки, у виділенні в окремий кластер об’єктів, що мають найменші показники схожості, при тому, що спочатку всі об’єкти розглядається як окремий кластер. Перевагами ієрархічних методів кластеризації є їх наочність і можливість отримати детальне уявлення про структуру даних. Ієрархічні алгоритми пов’язані з побудовою дендрограм, які описують близькість

окремих точок і кластерів один до одного та представляєють в графічному вигляді послідовність об’єднання (розділення) кластерів. Недоліки ієрархічних методів кластеризації – обмеження об’єму набору даних; вибір міри близькості; негнучкість отриманих класифікацій. Ієрархічні методи використовуються при невеликих об’ємах наборів даних. При великій кількості даних вони не придатні. У таких випадках використовують ітеративні методи.

Ітеративні методи – методи кластеризації, в яких кластери формуються виходячи з умов розбиття, які можуть бути змінені користувачем для досягнення бажаної цілі. Ці методи можуть призвести до утворення перетину кластерів, коли один об’єкт може одночасно належати декільком кластерам. В процесі поділу нові кластери формуються до тих пір, поки не буде виконано правило зупинки. Існує два підходи до розділення набору даних на певну кількість окремих кластерів. Перший полягає у визначенні меж кластерів як найбільш щільних ділянок в багатовимірному просторі початкових даних, тобто, визначення кластера там, де є велике "згущення" точок. Другий підхід полягає в мінімізації міри відмінності об’єктів. Найбільш поширений метод ітеративної кластеризації – *метод k-середніх* [115, 209, 214]. Ітеративні методи можна використовувати для великих об’ємів даних, також вони виявляють вищу стійкість по відношенню до шумів і викидів, некоректного вибору метрики, включення незначущих змінних в набір, що беруть участь в кластеризації. Недоліком ітеративних методів є те, що треба заздалегідь визначити кількість кластерів. Якщо невідоме число кластерів, треба використовувати ієрархічні алгоритми або декілька разів використати ітеративні методи з різною кількістю кластерів.

Перевагою методів контентної фільтрації є те, що для початку роботи рекомендаційної системи не потрібно великої кількості зареєстрованих користувачів. Головним недоліком даного підходу є неможливість системи рекомендувати нові об’єкти, які не прив’язані до інтересів користувачів. При цьому виникає проблема підтримки зворотного зв’язку з користувачем.

1.1.4. Дослідження методів створення гібридних рекомендаційних систем

Переважна більшість реальних рекомендаційних систем є складними гібридами різних методів фільтрації даних. Існують різні способи об'єднання декількох різних методів в гібридний метод. Виділяють такі основні стратегії гібридизації рекомендаційних систем [15, 18]:

1. Зважена (Weighted). Різні рекомендаційні системи працюють незалежно, а результати їх роботи об'єднуються за допомогою зваженої суми:

$$R=\sum_{i=1}^{n} w_i \cdot A_i \ , \tag{1.51}$$

де w_i – вага алгоритму A_i.

Правильно підібрані ваги та алгоритми дозволяють значно поліпшити якість рекомендацій. Множину ваг бажано визначати за допомогою деякої функції:

$$W=\{w_1, w_2,..., w_n\} = f\left(p_1, p_2,..., p_m\right) \ , \tag{1.52}$$

де $p_1, p_2,..., p_m$ – набір деяких параметрів. Це дозволяє зробити алгоритм більш адаптивним і досягти більшої точності.

2. З перемиканням (Switching). На основі критерію для перемикання обирається яку рекомендаційну систему використати для поточного випадку. Наприклад, для нового об'єкту, у якого немає оцінок, можна обрати контентну фільтрацію, а для об'єкту, якому вже виставлено багато оцінок – колаборативну.

3. Змішана (Mixed). В список рекомендацій потрапляє об'єднання рекомендацій, побудованих різними рекомендаційними системами.

4. З комбінацією ознак (Feature Combination). Дані одержані за допомогою колаборативної фільтрації виступають у якості додаткових ознак у контентній фільтрації.

5. Каскадна (Cascade). Спочатку використовується більш швидка

рекомендаційна система, потім більш повільна, але більш точна. Рекомендації одержані першою рекомендаційною системою ранжуються, та вибираються перші n найкращих результатів, які знову ранжуються (уточнюється) за допомогою другої рекомендаційної системи. В список рекомендацій потрапляють k ($k \leq n$) об'єктів, що мають максимальні прогнози оцінок за результатами роботи другої рекомендаційної системи.

6. Зі збільшенням числа ознак (Feature Augmentation). Вихідні дані від однієї або декількох рекомендаційних систем використовуються як вхідні ознаки для іншої системи. Таким чином можна на перший рівень обчислень додавати різні рекомендаційні системи, що працюють з різним набором ознак. Така система дозволяє гнучко додавати нові вхідні дані.

7. З мета-навчанням (Meta-level). Мета-навчання передбачає, що рекомендаційна система буде навчатися на вибірці, ознаками якої є результати роботи інших рекомендаційних систем.

Підсумовуючи проведене дослідження видів рекомендаційних систем наведемо їх загальну класифікацію на рис. 1.5.

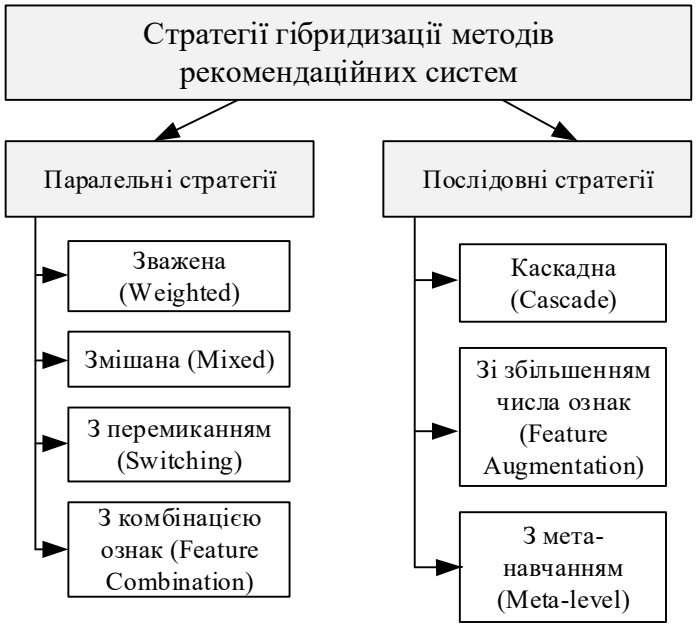


Рис. 1.5. Класифікація стратегій гібридизації методів роботи рекомендаційних систем

Вибір тієї або іншої стратегії гібридизації методів рекомендаційної системи залежить від потреб та вимог конкретного веб-сайту чи додатку, а також особливостей роботи самих методів фільтрації даних. Тому у кожній окремій ситуації стратегія гібридизації рекомендаційної системи має обиратися індивідуально, а правильність її вибору перевірятися експериментальним шляхом.

1.2. Дослідження внутрішніх та зовнішніх факторів, що можуть дестабілізувати роботу рекомендаційних систем

Сучасні рекомендаційні системи мають ряд стандартних проблем та недоліків, пов’язаних з внутрішніми та зовнішніми дестабілізуючими факторами.

До найпоширеніших загальних проблем рекомендаційних систем, пов’язаних з внутрішніми дестабілізуючими факторами у комп’ютерних мережах, можна віднести проблему холодного старту [9, 34, 76] та бульбашки фільтрів [95, 102].

Головними проблемами контентної фільтрації є складність виділення ознак об’єктів та проблема холодного старту для користувачів [9, 34, 76, 171].

Для колаборативної фільтрації актуальними є проблеми холодного старту для користувачів та холодного старту для об’єктів [9, 34, 76, 171].

Одна з головних проблем рекомендаційних систем – проблема холодного старту (Cold-start Problem, CSP). Вона виникає тоді, коли в системі з’являються нові елементи – або нові користувачі (User Cold-Start), історія вподобань яких порожня, або нові об’єкти (Item Cold-Start), у яких ще немає оцінок та/або набору ознак [9, 34, 76, 171].

У багатьох реальних системах проблема холодного старту може набувати характеру циклічної проблеми для вже відомих користувачів або об’єктів. Наприклад, якщо частина користувачів змінює свої інтереси. Дана

проблема отримала назву проблеми постійного холодного старту (Continuous Cold-start Problem, CoCoS) [9].

Як і проблема холодного старту, проблема постійного холодного старту може виникати з користувачами (User Continuous Cold-start Problem) та з об'єктами (Item Continuous Cold-start Problem).

User Continuous Cold-start Problem виникає для користувачів, що змінюють свої вподобання або рідко з’являються у системі та рідко оцінюють нові об’єкти.

Item Continuous Cold-start Problem виникає при наявності об’єктів, властивості яких можуть змінитися з часом.

Для вирішення проблеми холодного старту, як правило, застосовують наступні підходи:

1) гібридизація рекомендаційної системи з поєднанням контентної та колаборативної фільтрації [15, 18].

2) використання контексту, в якому створюються та надаються рекомендації (демографічні дані, час та дата тощо) [76, 171, 186, 203, 229].

Однак всі ці способи не підходять в разі проблеми постійного холодного старту, оскільки припускають, що після того, як користувач став «відомим», він залишається таким необмежену кількість часу, а об’єкти рекомендацій не можуть змінювати свої властивості. Для рішення даної проблеми треба не тільки прогнозувати вподобання користувачів, а й відслідковувати та прогнозувати зміну їх вподобань, а також враховувати можливість зміни властивостей об’єктів рекомендацій. Дану проблему на сьогоднішній день намагаються вирішувати методами машинного навчання, що підвищують адаптивність системи до постійних змін, а також врахуванням часу, як одного з параметрів при формуванні рекомендацій.

Не менш важливою проблемою рекомендаційної системи є проблема бульбашки фільтрів [95]. Класичні рекомендаційні системи пропонують користувачам об’єкти, виходячи лише з їх попередніх вподобань. Проблема бульбашки фільтрів виникає у рекомендаційних системах, коли алгоритм

формування рекомендацій та видачі інформації вибірково підбирає дані, враховуючи тільки те, яку інформацію користувач переглядав і оцінював раніше, і, в результаті, користувачі відділяються від інформації, яка їх раніше не цікавила або невідома їм, або не подобається, фактично ізолюючи їх у власних «бульбашках» [95, 102]. Отже, користувач потрапляє у інформаційне середовище, в якому спостерігає лише обмежену кількість однотипних об'єктів. Це явище виникає, якщо користувач для одержання нової інформації використовує переважно списки рекомендацій. Наслідки, викликані бульбашкою фільтрів [148]:

1. Користувач не одержує альтернативну інформацію, яка може бути йому корисною (наприклад, види об'єктів, про які він зовсім не знає, але які ефективніше вирішать задачі його пошуку).

2. У користувача формується викривлена точка зору на інформаційне середовище, так як він не бачить картини в цілому (наприклад, при рекомендації новин).

3. Користувач може втратити інтерес до списку рекомендацій, так як йому весь час пропонують однотипні об'єкти (наприклад, втратить інтерес до прослуховування онлайн радіо з однотипним набором пісень).

Так як усі рекомендаційні системи як основну метрику якості своєї роботи використовують точність прогнозування вподобань користувачів на основі їх попередніх дій, а формальна постановка задачі прогнозу оцінок виглядає наступним чином [76, 105, 157]:

$$d(R,V) \to \min, \tag{1.53}$$

де $R = (r_1, r_2, \ldots, r_n)$ – вектор, що містить список прогнозованих оцінок користувача, впорядкований по спаданню за величиною оцінок; $V = (v_1, v_2,\ldots, v_n)$ – вектор, що містить справжні оцінки користувача, невідомі системі на етапі формування списку рекомендацій, то для всіх рекомендаційних систем проблема бульбашки фільтрів є актуальною, так як якісна рекомендаційна система повинна створювати рекомендації максимально схожі на попередні вподобання користувача.

Для вирішення проблеми бульбашки фільтрів, як правило, застосовують накладання додаткових вимог до процесу створення рекомендацій, наприклад, забезпечення наступних властивостей, що вимагають додавання випадкових елементів у списки рекомендацій [21, 39, 43, 76, 102]:

1. *Різноманітність* (Diversity) – міра схожості між елементам списку рекомендацій не повинна бути меншою певної заданої величини. Міру схожості між елементами можна визначати на основі коефіцієнтів подоби, наприклад, за допомогою однієї з формул (1.1)-(1.13).

2. *Неочікуваність* (Serendipity) – у список рекомендацій з деякою ймовірністю повинні потрапляти елементи підібрані випадковим чином або на основі методів, не пов'язаних з прогнозуванням вподобань користувача.

3. *Новизна* (Novelty) – у список рекомендацій з деякою ймовірністю повинні потрапляти елементи, які ще ніким не були оцінені або отримали мало оцінок.

При забезпеченні виконання розглянутих вище додаткових вимог до формування списку рекомендацій (різноманітності, неочікуваності та новизни), буде зменшуватися точність прогнозування вподобань, але можна буде подолати проблему бульбашки фільтрів та покращити інші показники якості роботи рекомендаційної системи, наприклад, покриття каталогу.

До зовнішніх дестабілізуючих факторів у роботі рекомендаційної системи можна віднести різні інформаційні загрози [76, 148, 176]. Існує декілька основних загроз інформаційній безпеці користувачів рекомендаційної системи, а саме: загроза одержати рекомендації, сформовані внаслідок накручування рейтингів певних об'єктів в результаті інформаційної атаки; загроза втрати приватності даних користувача, зокрема, приватності його вподобань, коли їх можуть дізнатися та використати треті особи для просування своїх інтересів; загроза одержати неякісні списки рекомендацій, використання яких може нести ризики для життя чи здоров'я користувача або впливати на рішення у соціальних чи політичних сферах у інтересах третіх сторін. Нижче розглянемо дані загрози детальніше.

Атаки накручування рейтингів (атаки ін'єкцією профілів). Некоректні рекомендації можуть виникати при атаках на рекомендаційну систему з метою збільшення (зменшення) рейтингів певних об'єктів. Реалізуються такі атаки шляхом створення множини акаунтів ботів (бот-мережі), які скоординовано виставляють високі (або низькі) оцінки певному об'єкту чи об'єтам [22, 46, 60, 61, 69, 76].

Загрози приватності користувачів. Рекомендаційні системи збирають велику кількість даних про користувачів, значну частину яких користувачі охоче надають самі в обмін на корисні рекомендації. Однак для більшості користувачів, важливо щоб їхні вподобання залишалися приватними, тобто, жодна третя сторона не могла використовувати рекомендаційні системи, щоб дізнатися інформацію про них або їх вподобання. Дана загроза цілком реальна. Як один з прикладів можна навести скандальні ситуації з рекомендаціями друзів у Facebook [32], які виникали через експерименти з використанням даних геолокації [33], подібні рекомендації частково порушували приватні дані людей та давали інформацію третім особам про їх переміщення.

Ризиковані рекомендації. В деяких випадках рекомендації можуть бути пов'язані з ризиком [76]. Наприклад, якщо об'єктами в рекомендаційній системі є акції, кредити, депозити, ліки, медичні послуги, політичні акції тощо. В таких випадках може бути необхідним врахування не тільки вподобань користувача при формуванні рекомендацій, а й інших факторів, врахування яких здатне мінімізувати ризик для користувача, що буде переглядати та обирати рекомендації. Ризиковані рекомендації можна вилучати зі списку рекомендацій або маркувати їх відповідною інформацією для користувачів.

Було проведено дослідження та порівняльний аналіз основних груп методів та моделей синтезу рекомендаційних систем з точки зору наявності/відсутності у них розглянутих проблем. Результати порівняльного аналізу наведені у таблиці 1.1.

Таблиця 1.1. Результати порівняльного аналізу відомих методів побудови рекомендаційних систем

Назва групи методів	Проблема User Cold-Start	Проблема Item Cold-Start	Проблема User Continuous Cold-Start	Проблема Item Continuous Cold-Start	Проблема бульбашки фільтрів	Проблеми при розрідженості матриці рейтингів	Вразливість до інформаційних атак зміни рейтингів	Вразливість до інформаційних атак порушення приватності	Виявлення ознак інформаційної атаки перед пошуком ботів
Методи на основі моделей сусідства									
Методи user-based колаборативної фільтрації [37, 76, 209]	+	+	+	+	+	+	+	+	–
Методи item-based колаборативної фільтрації [76, 78, 205, 209]	+	+	+	+	+	+	+	+	–
Методи на основі матричних факторизаційних моделей									
Метод FunkSVD [27]	+/–	+/–	+	+	+	+/–	+	+	–
Метод SVD++ [19, 35, 76]	+/–	+/–	+	+	+	+/–	+	+	–
Метод Asymmetric SVD [41, 76, 87]	+/–	+/–	+	+	+	+/–	+	+	–
Методи на основі моделей заснованих на знаннях									
Методи контентної фільтрації на основі Байєсівських мереж [76, 209]	+	–	+	+	+	–	–/+	+	–
Методи контентної фільтрації на основі дерев рішень [76, 209]	+	–	+	+	+	–	–/+	+	–
Методи на основі асоціативних правил [1, 73, 79, 137, 224]	+	–	+	+	+	–	+/–	+	–
Методи контекстної фільтрації [76, 186, 203]	–	–	+	+	+	–	+/–	+	–
Методи на основі експертних систем [76]	+/–	–	+	+	+	–	–/+	+	–
Методи на основі моделей класифікації та кластеризації даних									
Методи фільтрації на основі кластеризації даних [36, 81, 209]	+/–	–	+	+	+	–	–/+	+	–
Методи фільтрації на основі нейронних мереж [209]	+/–	–	+	+	+	–	–/+	+	–
Методи на основі моделей заснованих на знаннях про соціальні зв'язки									
Методи соціальної фільтрації [31, 76]	–/+	+	+/–	+/–	+	+/–	+/–	+	–
Методи засновані на репутаційних системах [38, 61, 63, 72, 76]	–/+	+	+/–	+/–	+	+/–	–/+	+	–
Методи, що враховують або створюють зміну вподобань у часі									
Метод TimeSVD++ [42, 76]	+/–	+/–	–	–	–/+	+/–	+	+	–
Методи, що враховують давність дій користувача за допомогою часових коефіцієнтів [24, 51, 76, 94]	+/–	+/–	–/+	–/+	+	+	+	+	–

Назва групи методів	Проблема User Cold-Start	Проблема Item Cold-Start	Проблема User Continuous Cold-Start	Проблема Item Continuous Cold-Start	Проблема бульбашки фільтрів	Проблеми при розрідженості матриці рейтингів	Вразливість до інформаційних атак зміни рейтингів	Вразливість до інформаційних атак порушення приватності	Виявлення ознак інформаційної атаки перед пошуком ботів
Методи, що враховують циклічність дій користувача за допомогою часових коефіцієнтів [76, 100]	+/–	+/–	–/+	–/+	+	+	+	+	–
Рандомізовані методи з використанням деяких або усіх з принципів формування неодноманітного списку рекомендацій: Diversity, Serendipity, Novelty [21, 39, 43, 76, 102]	+/–	+/–	–/+	–/+	–	+/–	+/–	–/+	–
Методи на основі моделей робастних рекомендаційних систем									
Рандомізовані методи з зашумленням списків рекомендацій [76]	+/–	+/–	+/–	+/–	+/–	+/–	+	–/+	–
Методи з підсистемою виявлення профілів ботів [98, 103, 104]	+/–	+/–	+/–	+/–	+/–	+	–/+	–/+	–

Як показали проведені дослідження, переважна більшість моделей та методів синтезу рекомендаційних систем вразливі до дії внутрішніх та зовнішніх дестабілізуючих факторів. Підвищення стійкості існуючих та розробка нових більш стійких до дестабілізуючих факторів моделей та методів дасть змогу підвищити якість роботи рекомендаційних систем.

1.3. Постановка науково-технічної проблеми

Незважаючи на існуючі дослідження в галузі рекомендаційних систем, аналіз відомих моделей та методів показав, що сьогодні не в повній мірі вирішено питання забезпечення стійкості рекомендаційних систем до дестабілізуючих факторів, внаслідок чого вплив таких факторів значно знижує точність формування рекомендацій користувачам (рис. 1.6).

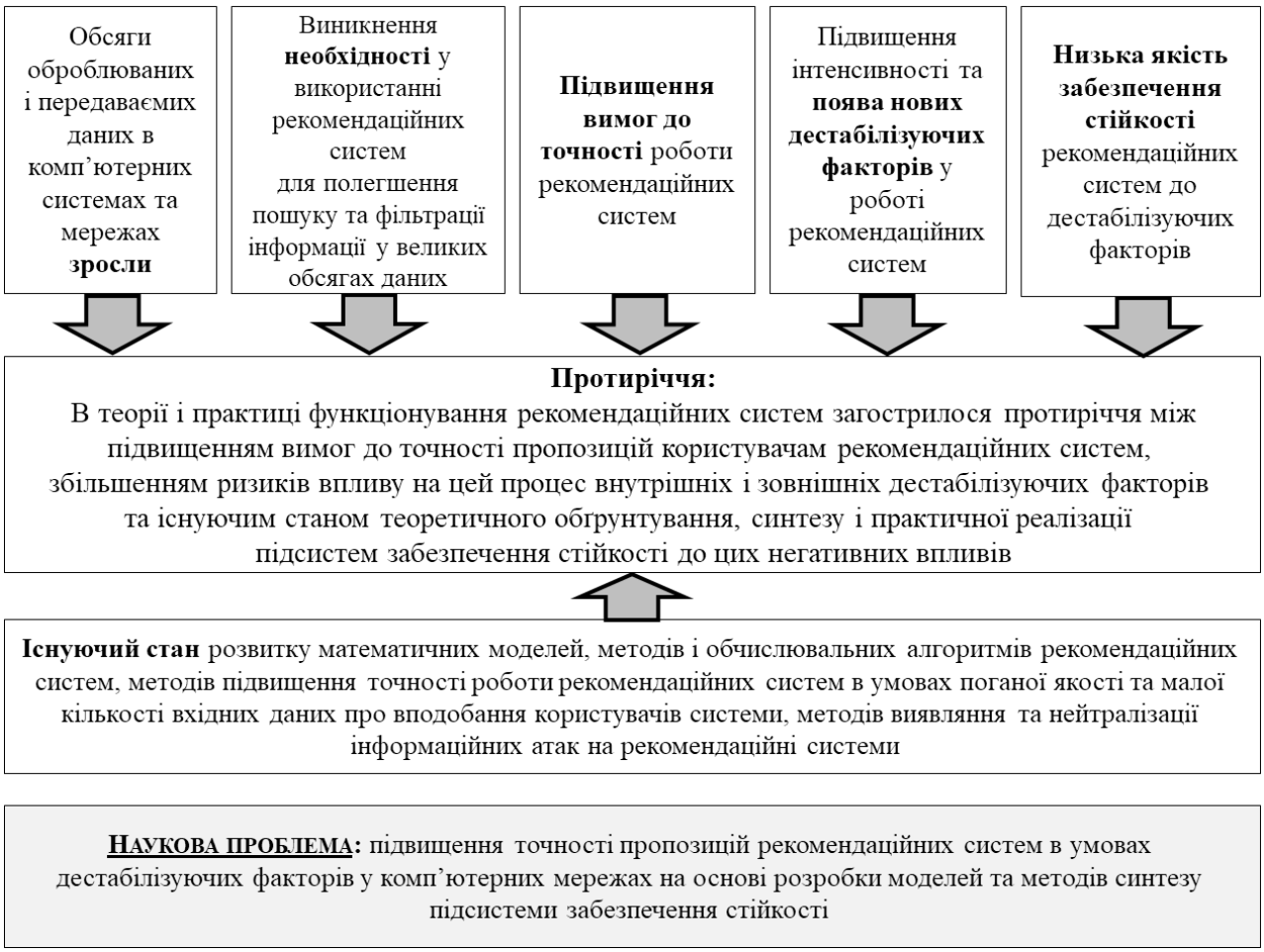


Рис. 1.6. Основні існуючі протиріччя в області розробки і експлуатації рекомендаційних систем. Наукова проблема

У зв'язку з бурхливим розвитком сучасних інформаційних технологій, підвищенням обсягів інформації, що передається та оброблюється у мережі Інтернет, появою нових інформаційних послуг і сервісів, актуальність використання рекомендаційних систем зростає. З одного боку зросли обсяги оброблюваних і передаваємих даних в мережі Інтернет, виникла необхідність у використанні рекомендаційних систем для полегшення пошуку та фільтрації інформації у великих обсягах даних, підвищилися вимоги до точності роботи рекомендаційних систем, підвищилася інтенсивність та поява нових дестабілізуючих факторів у роботі рекомендаційних систем. З іншого боку, існуючий стан теоретичного обґрунтування, синтезу і практичної реалізації підсистем забезпечення стійкості рекомендаційних

систем до зовнішніх та внутрішніх дестабілізуючих факторів не дозволяє реалізувати забезпечення цих підвищених та нових вимог.

Таким чином, на сьогоднішній день в теорії і практиці функціонування рекомендаційних систем загострилося **протиріччя** між підвищенням вимог до точності пропозицій користувачам рекомендаційних систем, збільшенням ризиків впливу на цей процес внутрішніх і зовнішніх дестабілізуючих факторів та існуючим станом теоретичного обґрунтування, синтезу і практичної реалізації підсистем забезпечення стійкості до цих негативних впливів.

Подолати цю суперечність можна шляхом вирішення актуальної **науково-практичної проблеми** підвищення точності пропозицій рекомендаційних систем в умовах дестабілізуючих факторів у комп'ютерних мережах на основі розробки моделей та методів синтезу підсистеми забезпечення стійкості.

Висновки до розділу 1

Перспективним напрямком в систематизації, фільтрації, пошуку та наданні даних користувачам соціальних мереж та контент-орієнтованих веб-сайтів є рекомендаційні системи, основні засоби яких, повинні реалізовувати необхідні послуги надання користувачу релевантних рекомендацій в певному місці, в певний час та через вірний канал комунікації. Дана дисертаційна робота присвячена підвищенню точності пропозицій рекомендаційної системи за рахунок забезпечення її стійкості до внутрішніх та зовнішніх дестабілізуючих факторів.

У даному розділі проведене дослідження та порівняльний аналіз існуючих моделей і методів синтезу рекомендаційних систем, розроблена їх класифікація. Розглянуто основні внутрішні та зовнішні дестабілізуючі фактори у роботі сучасних рекомендаційних систем.

Проведені дослідження та порівняльний аналіз відомих методів синтезу рекомендаційних систем показали, що одним з перспективних напрямків у розвитку даних методів є удосконалення існуючих методів формування рекомендацій та розробка підсистем забезпечення стійкості до проблеми холодного старту, проблеми низької якості і кількості вхідних даних та до інформаційних атак.

У зв'язку з бурхливим розвитком сучасних інформаційних технологій, підвищенням обсягів інформації, що передається та оброблюється у мережі Інтернет, появою нових інформаційних послуг і сервісів, актуальність використання рекомендаційних систем зростає. З одного боку зросли обсяги оброблюваних і передаваємих даних в мережі Інтернет, виникла необхідність у використанні рекомендаційних систем для полегшення пошуку та фільтрації інформації у великих обсягах даних, підвищилися вимоги до точності роботи рекомендаційних систем, підвищилася інтенсивність та поява нових дестабілізуючих факторів у роботі рекомендаційних систем. З іншого боку, існуючий стан теоретичного обґрунтування, синтезу і практичної реалізації підсистем забезпечення стійкості рекомендаційних систем до зовнішніх та внутрішніх дестабілізуючих факторів, не дозволяє реалізувати забезпечення цих підвищених та нових вимог. Перераховані фактори в сукупності складають об'єктивно існуюче науково-технічне протиріччя, на вирішення якого і спрямована мета даної роботи.

Розв’язання важливої науково-практичної проблеми, яка полягає у підвищення точності пропозицій рекомендаційних систем в умовах дестабілізуючих факторів у комп’ютерних мережах на основі розробки моделей та методів синтезу підсистеми забезпечення стійкості, є актуальною. Розв’язання зазначеної проблеми має важливе значення як для розвитку окремого напрямку теорії аналізу даних, так і для вирішення прикладних питань забезпечення ефективного пошуку, класифікації та рекомендації даних у комп’ютерних мережах.

РОЗДІЛ 2.
МЕТОД ВИЗНАЧЕННЯ ДИНАМІКИ ЙМОВІРНОСТЕЙ ПЕРЕБУВАННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ В СВОЇХ МОЖЛИВИХ СТАНАХ ТА ОБҐРУНТУВАННЯ ПІДХОДІВ ДО ЗАБЕЗПЕЧЕННЯ ЇЇ СТІЙКОСТІ

У даному розділі запропоновано метод визначення динаміки ймовірностей перебування стійкої рекомендаційної системи в своїх можливих станах на основі використання математичного апарату марківських та напівмарківських процесів. Даний метод може використовуватися для створення математичних моделей різних рекомендаційних систем та їх підсистем для оптимізації частоти періодичних планових дій у системі.

Також у даному розділі обґрунтовано вибір підходів до забезпечення стійкості рекомендаційної системи в умовах дії зовнішніх та внутрішніх дестабілізуючих факторів у комп'ютерних мережах.

2.1. Розробка методу визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах

Рекомендаційна система під час своєї роботи може перебувати у різних станах. Для розробки методу визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах розглянемо загальний випадок для двох станів:

– **1 стан** (назвемо його **Normal**) – нормальна робота системи, списки рекомендацій, що видаються користувачам відповідають їх потребам та вподобанням. Позначимо цей стан H_0.

– **2 стан** (назвемо його **Problem**) – у цьому стані відбувається помітне зниження точності рекомендацій під впливом зовнішніх чи внутрішніх дестабілізуючих факторів. Під зовнішніми дестабілізуючими факторами

будемо розуміти інформаційні атаки, під внутрішніми – проблему холодного старту, розрідженості та недостатнбої кількості вхідних даних тощо. Позначимо цей стан H_1.

Рекомендаційна система є складною динамічною системою, яку можна розглядати з точки зору систем масового обслуговування. Виникнення зовнішніх чи внутрішніх дестабілізуючих факторів можна розглядати як вплив випадкового процесу.

Для математичного моделювання динаміки ймовірностей станів рекомендаційної системи в процесі перебування під дією внутрішніх та зовнішніх дестабілізуючих факторів використаємо теорію ланцюгів Маркова, марківських та напівмарківських процесів.

Ланцюги Маркова можна застосувати для моделювання процесу зміни станів рекомендаційної системи під час інформаційної атаки так як даний процес є випадковим та задовольняє властивості Маркова (властивості відсутності пам’яті), а також приймає скінченну кількість станів (Normal та Problem).

Значна частина математичних моделей процесів функціонування динамічних систем побудована і описана в термінах загальної теорії графів [109, 123]. При цьому зазвичай передбачається, що система в кожен момент часу може знаходитися в одному з можливих станів і переходить з цього стану в інший під впливом якогось випадкового процесу. Припускається, що задані (або можуть бути отримані в результаті статистичної обробки вихідних даних) закони розподілу тривалості перебування системи в кожному з можливих станів до переходу в інший стан. При цьому в багатьох практичних випадках може бути сформульована і вирішена задача відшукання стаціонарного розподілу ймовірностей станів системи. Однак теоретичний і практичний інтерес представляє рішення більш складного завдання – відшукання розподілу ймовірностей перебування системи в можливих своїх станах в довільний момент часу після початку функціонування з заданого початкового стану.

Вирішення цього завдання при найзагальніших припущеннях щодо характеру впливаючого випадкового процесу практично нездійсненне. Однак це рішення може бути отримане для важливого окремого випадку, коли цей процес є марківским [113, 213]. Вичерпні результати отримані для випадку, коли випадковий процес, що визначає переходи з одного стану в інший, є дискретним в фазовому просторі станів, а закон розподілу довжини інтервалів між переходами – експоненційний [130, 223]. Співвідношення для розрахунку фінальних розподілів ймовірностей станів отримані і для напівмарковських систем, коли щільності розподілу тривалості перебування до переходу в інший стан – інтегровні функції [107, 126]. Однак завдання отримання простих співвідношень для розрахунку ймовірностей перебування в кожному із станів в довільний момент часу вивчена недостатньо. Проблема полягає в наступному. Відсутня методика, що встановлює зв'язок між двома математичними об'єктами. Перший – щільності розподілу тривалостей перебування системи в кожному зі станів до переходу в інший стан. Другий – шукані функції, що описують динаміку ймовірностей перебування системи в можливих своїх станах. Вирішення цієї проблеми – актуальне завдання.

Проблема аналізу напівмарковських систем обговорюється в численних публікаціях. В [20] розглядається задача оцінки ефективності системи, моделлю якої є система масового обслуговування з неоднорідним вхідним потоком. При цьому відшукується фінальний розподіл ймовірностей станів для вкладеного марківського ланцюга. В [23] з використанням напівмарківської моделі досліджується виробнича система. Аналіз завершується розрахунком фінального розподілу ймовірностей станів системи. В [26] розглядається система обслуговування з довільним вхідним потоком. Результат дослідження – стаціонарний розподіл ймовірностей станів. В [48] досліджується можливість застосування напівмарковських моделей для задач аналізу комп'ютерних мереж, транспортних мереж, об'єктів Інтернету речей. При цьому рішення щодо ефективності системи приймається на підставі одержуваного фінального розподілу ймовірностей

станів. Система обслуговування з непуассонівським входом і неекспоненціальним обслуговуванням вивчається в [49] з метою отримання стаціонарних характеристик ефективності. В [70] досліджується система обслуговування з напівмарківським вхідним потоком. Аналіз системи завершується розрахунком фінального розподілу ймовірностей станів. Нарешті, в [77] проведено аналіз системи обслуговування з довільним розподілом випадкової тривалості обслуговування. Для оцінки ефективності системи використовується отриманий стаціонарний розподіл ймовірностей станів.

В результаті дослідження відомих публікацій з проблеми аналізу напівмарковських систем можна зробити наступний висновок. Відомі теоретичні результати дослідження напівмарковських систем обмежуються розрахунком фінального розподілу ймовірностей станів системи. При вирішенні деяких практичних завдань цього досить. Однак, у багатьох випадках, наприклад, при вирішенні задач оцінки ефективності відновлюваних систем, принципово важливо знати динаміку ймовірності перебування системи на множині працездатних станів. Ця ж проблема важлива для багатоканальних систем обслуговування критичного призначення. Ступінь готовності таких систем визначається значенням ймовірності того, що число нормально функціонуючих каналів не нижче заданого.

Таким чином, у важливому для теорії і практики напрямку дослідження великого класу складних систем, наприклад, інформаційних систем віртуальних соціальних мереж, контент-орієнтованих веб-сайтів та рекомендаційних систем, модель функціонування яких можна описати в термінах теорії напівмарковських процесів, є суттєва прогалина, пов'язана з вивченням динаміки ймовірностей станів таких систем. Ця обставина формує необхідність створення методу визначення динаміки ймовірностей перебування складної системи, зокрема, рекомендаційної системи, в своїх можливих станах.

Введемо наступні позначення:

$f_{01}(t)$ – щільність розподілу тривалості перебування системи в стані H_0 до переходу в стан H_1;

$f_{10}(t)$ – щільність розподілу тривалості перебування системи в стані H_1 до переходу в стан H_0;

$G_{00}(t)$ – умовна ймовірність опинитися в стані H_0 в момент часу t, якщо в початковий момент часу об'єкт знаходиться в стані H_0;

$G_{01}(t)$ – умовна ймовірність опинитися в стані H_1 в момент часу t, якщо в початковий момент часу об'єкт знаходиться в стані H_0;

$G_{10}(t)$ – умовна ймовірність опинитися в стані H_0 в момент часу t, якщо в початковий момент часу об'єкт знаходиться в стані H_1;

$G_{11}(t)$ – умовна ймовірність опинитися в стані H_1 в момент часу t, якщо в початковий момент часу об'єкт знаходиться в стані H_1.

На рис. 2.1 зображений ланцюг Маркова для динаміки зміни станів рекомендаційної системи під час інформаційної атаки.

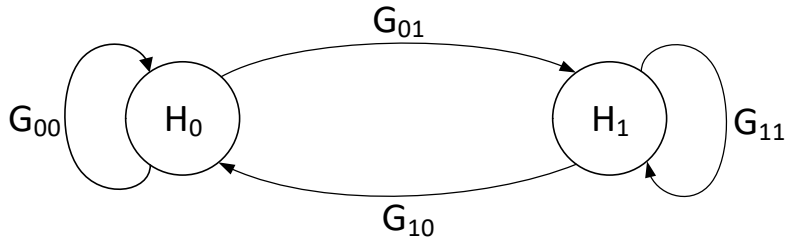


Рис. 2.1. Ланцюг Маркова для динаміки зміни двох станів рекомендаційної системи

Рекомендаційна система із стану H_0 (Normal) може перейти в стан H_1 (Problem), якщо на неї почнуть діяти зовнішні або внутрішні дестабілізуючі фактори та своєю дією знижувати точність рекомендацій.

Рекомендаційна система із стану H_1 (Problem) може перейти в стан H_0 (Normal), якщо буде нейтралізована дія дестабілізуючих факторів.

Для моделювання динаміки зміни станів рекомендаційної системи під час дії дестабілізуючих факторів важливо знати не тільки стаціонарний розподіл ймовірностей станів системи, а й значення цих ймовірностей в будь-який момент часу. Ця інформація дає можливість вирішувати задачі управління станом системи. Для досягнення поставленої мети необхідно вирішити такі завдання:

– розробити математичну модель, яка встановлює зв'язок між заданим набором щільності розподілу випадкових тривалостей перебування системи в своїх можливих станах і функціями, що описують динаміку ймовірностей станів.

– розробити метод отримання аналітичних співвідношень для безпосереднього розрахунку ймовірностей перебування системи в можливих станах в довільний момент часу.

– розглянути на конкретному прикладі технологію реалізації розробленого методу розрахунку співвідношень, що описують ймовірнісну динаміку станів напівмарківської системи.

Етапи запропонованого методу будуть наступними:

Етап 1. Побудова ланцюга Маркова для можливих станів конкретної рекомендаційної системи або її підсистеми.

Етап 2. Побудова системи інтегральних рівнянь, що пов'язує відомі щільності розподілу тривалостей перебування рекомендаційної системи в можливих станах і шукані функції, що описують ймовірнісну динаміку системи.

Етап 3. Розв’язання отриманої системи інтегральних рівнянь з використанням перетворення Лапласа. Результатом будуть співвідношення для розрахунку умовних ймовірностей знаходження системи в можливих станах в довільний момент часу t, якщо в початковий момент часу вона знаходилася в стані H_0.

Розглянемо другий та третій етапи запропонованого методу визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах детальніше.

2.1.1. Розробка математичної моделі динаміки ймовірностей станів рекомендаційної системи

Введемо математичну модель ймовірнісної динаміки станів рекомендаційної системи наступним чином. Нехай напівмарківська система може перебувати в одному з n можливих станів $(H_1,...,H_n)$. Система функціонує у зовнішньому середовищі, під впливом якого вона переходить з одного стану в інший. Формальний опис механізму взаємодії середовища і системи задамо наступним набором щільностей розподілу випадкових величин:

$f_{ij}(t)$ – щільність розподілу тривалості перебування системи в стані H_i до переходу в стан H_j; $i=1,2,...,n$, $j=1,2,...,n$.

Випадкову динаміку станів рекомендаційної системи опишемо набором функцій:

$G_{ij}(t)$ – умовна ймовірність опинитися в стані H_j в момент часу *t*, якщо в початковий момент система знаходилася у стані H_i, $i=1,2,...,n$, $j=1,2,...,n$.

Для знаходження невідомих функцій $G_{ij}(t)$ введемо систему інтегральних рівнянь:

$$G_{ij}(t)=\int_0^t f_{ik}(\tau)G_{kj}(t-\tau)d\tau,\ i=1,2,...,n,\ j=1,2,...,n. \qquad (2.1)$$

Технологію реалізації методу розглянемо на простому прикладі системи з двома можливими станами H_0 та H_1.

Запишемо систему рівнянь відносно невідомих функцій $G_{00}(t)$, $G_{01}(t)$, $G_{10}(t)$, $G_{11}(t)$.

Об'єкт, що знаходиться в початковий момент часу в стані H_0, може виявитися в цьому стані H_0 при реалізації одного з двох можливих незалежних варіантів. По-перше, об'єкт може залишатися в H_0, не покидаючи цей стан протягом всього інтервалу $[0,t]$. По-друге, об'єкт може залишити

стан H_0 в якийсь момент часу $\tau \in [0,t]$, повернувшись далі до моменту часу t в стан H_0. Звідси:

$$G_{00}(t) = \left(1 - \int_0^t f_{01}(\tau) d\tau\right) + \int_0^t f_{01}(\tau) \cdot G_{10}(t-\tau) d\tau . \qquad (2.2)$$

Об'єкт, що знаходиться в початковий момент часу в стані H_0, може виявитися в стані H_1, перейшовши в цей стан в момент часу $\tau \in [0,t)$, зробивши після цього на залишеному до моменту часу t інтервалі $(\tau,t]$ деяке число переходів зі стану H_1 з поверненням в нього на момент часу t. При цьому:

$$G_{01}(t) = \int_0^t f_{01}(\tau) \cdot G_{11}(t-\tau) d\tau . \qquad (2.3)$$

Об'єкт, що знаходиться в початковий момент часу в стані H_1, може виявитися в стані H_0, перейшовши в цей стан в момент часу $\tau \in [0,t)$, зробивши після цього на залишеному до моменту часу t інтервалі $(\tau,t]$ деяке число переходів зі стану H_0 з поверненням в нього на момент часу t. При цьому:

$$G_{10}(t) = \int_0^t f_{10}(\tau) \cdot G_{00}(t-\tau) d\tau . \qquad (2.4)$$

Нарешті об'єкт, що знаходиться в початковий момент часу в стані H_1, може залишитися в цьому стані до моменту часу t або, вийшовши з цього стану в момент часу $\tau \in [0,t)$, повернутися в нього на момент часу t. При цьому:

$$G_{11}(t) = \left(1 - \int_0^t f_{10}(\tau) d\tau\right) + \int_0^t f_{10}(\tau) \cdot G_{01}(t-\tau) d\tau . \qquad (2.5)$$

Система інтегральних рівнянь (2.2)-(2.5) утворює математичну модель, що пов’язує відомі щільності розподілу тривалостей перебування рекомендаційної системи в можливих станах і шукані функції, що описують

ймовірнісну динаміку системи. Використовуємо цю модель.

Відзначимо, що при побудові цієї математичної моделі ніякі обмеження на характер щільності не накладалися. Таким чином, ця математична модель може бути використана для ймовірнісного аналізу будь-якої напівмарківської системи.

2.1.2. Розробка методики отримання аналітичних співвідношень для розрахунку ймовірностей перебування рекомендаційної системи в можливих станах в довільний момент часу

Отриману систему рівнянь (2.2)-(2.5) вирішуємо з використанням перетворення Лапласа [128, 134, 208].

Перетворенням Лапласа функції $u(t)$ є функція:

$$L(u(t)) = \int_0^\infty u(t)e^{-st}dt. \tag{2.6}$$

Для спрощення запису зручно ввести $L(u(t)) = u^*(s)$.

З врахуванням властивостей перетворення Лапласа запишемо рівняння (2.2)-(2.5) в операторній формі.

Якщо інтегрувати (5) по частинах, то:

$$L\left(\int_0^t u(\tau)d\tau\right) = \int_0^\infty e^{-st}\left(\int_0^t u(\tau)d\tau\right) = -\frac{1}{s}\left[\left(e^{-st}\int_0^t u(\tau)d\tau\right)\Big|_0^\infty - \int_0^\infty e^{-st}u(\tau)d\tau\right] =$$

$$= \frac{1}{s}L\big(u(\tau)\big) = \frac{1}{s}u^*(s).$$

При цьому лапласові зображення співвідношень (2.2)-(2.5) матимуть вигляд:

$$G_{00}^*(s) = \frac{1}{s}\big(1 - f_{01}^*(s)\big) + f_{01}^*(s)\cdot G_{10}^*(s), \tag{2.7}$$

$$G_{01}^*(s) = f_{01}^*(s)\cdot G_{11}^*(s), \tag{2.8}$$

$$G_{10}^*(s) = f_{10}^*(s) \cdot G_{00}^*(s), \tag{2.9}$$

$$G_{11}^*(s) = \frac{1}{s}\left(1 - f_{10}^*(s)\right) + f_{10}^*(s) \cdot G_{01}^*(s), \tag{2.10}$$

Отримана система рівнянь розпадається на дві пари {(2.7), (2.9)} і {(2.8), (2.10)}, кожна з яких містить по дві невідомі функції. Маємо:

$$G_{00}^*(s) = \frac{1}{s}\left(1 - f_{01}^*(s)\right) + f_{01}^*(s) \cdot G_{10}^*(s),$$

$$G_{10}^*(s) = f_{10}^*(s) \cdot G_{00}^*(s).$$

Підставляючи друге з цих рівнянь на початок, отримаємо:

$$G_{00}^*(s) = \frac{1}{s}\left(1 - f_{01}^*(s)\right) + f_{01}^*(s) \cdot f_{10}^*(s) \cdot G_{00}^*(s).$$

Звідси:

$$G_{00}^*(s) \cdot \left(1 - f_{01}^*(s) \cdot f_{10}^*(s)\right) = \frac{1}{s}\left(1 - f_{01}^*(s)\right),$$

$$G_{00}^*(s) = \frac{1}{s} \cdot \frac{1 - f_{01}^*(s)}{1 - f_{01}^*(s) \cdot f_{10}^*(s)}, \tag{2.11}$$

Підставляючи (2.11) в (2.9), отримаємо:

$$G_{10}^*(s) = \frac{1}{s} \cdot \frac{\left(1 - f_{01}^*(s)\right) \cdot f_{10}^*(s)}{1 - f_{01}^*(s) \cdot f_{10}^*(s)}, \tag{2.12}$$

Аналогічно цьому:

$$G_{01}^*(s) = f_{01}^*(s) \cdot G_{11}^*(s),$$

$$G_{11}^*(s) = \frac{1}{s}\left(1 - f_{10}^*(s)\right) + f_{10}^*(s) \cdot f_{01}^*(s) \cdot G_{11}^*(s).$$

Звідси:

$$G_{11}^*(s) \cdot \left(1 - f_{10}^*(s) \cdot f_{01}^*(s)\right) = \frac{1}{s}\left(1 - f_{10}^*(s)\right),$$

$$G_{11}^*(s) = \frac{1}{s} \cdot \frac{1 - f_{10}^*(s)}{1 - f_{10}^*(s) \cdot f_{01}^*(s)}. \qquad (2.13)$$

Підставляємо тепер (2.13) в (2.8):

$$G_{01}^*(s) = \frac{1}{s} \cdot \frac{\left(1 - f_{10}^*(s)\right) \cdot f_{01}^*(s)}{1 - f_{10}^*(s) \cdot f_{01}^*(s)}. \qquad (2.14)$$

Використовуємо отримані загальні співвідношення, що описують лапласові зображення шуканих функцій, для вирішення конкретного завдання. Нехай відновлювана система може перебувати в одному з двох станів:

– H_0 – система нормально функціонує;

– H_1 – система після відмови (або помилок у роботі) відновлюється.

Виконаємо спочатку розрахунки для найпростішого випадку, коли система є марковською. Задамо щільності розподілу тривалостей перебування в кожному із станів до переходу в інший стан наступним чином:

$$f_{01}(t) = \lambda e^{-\lambda t},\ f_{10}(t) = \mu e^{-\mu t}.$$

При цьому:

$$f_{01}^*(s) = \frac{\lambda}{s + \lambda},\ f_{10}^*(s) = \frac{\mu}{s + \mu}. \qquad (2.15)$$

Підставляючи (2.15) в (2.11) – (2.14), отримаємо аналітичні описи зображень $G_{00}^*(s)$, $G_{01}^*(s)$, $G_{10}^*(s)$, $G_{11}^*(s)$, що відповідають заданим початковим даним.

При цьому:

$$G_{00}^*(s) = \frac{1}{s} \cdot \frac{1 - \frac{\lambda}{s + \lambda}}{1 - \frac{\lambda}{s + \lambda} \cdot \frac{\mu}{s + \mu}} = \frac{1}{s} \cdot \frac{s(s + \mu)}{s^2 + s(\lambda + \mu)} = \frac{s + \mu}{s \cdot (s + \lambda + \mu)}. \qquad (2.16)$$

Зворотне перетворення Лапласа виконаємо, розкладаючи (2.16) на елементарні дроби. Знайдемо корені полінома, що стоїть в знаменнику, вирішуючи рівняння $s(s+\lambda+\mu)=0$, звідки:

$$s_1=0,\ s_2=-(\lambda+\mu).$$

Перепишемо тепер (2.16) наступним чином:

$$\frac{s+\mu}{s(s+\lambda+\mu)}=\frac{\alpha}{s-s_1}+\frac{\beta}{s-s_2}=\frac{\alpha}{s}+\frac{\beta}{s+\lambda+\mu}. \qquad (2.17)$$

Після приведення до спільного знаменника, одержимо:

$$\frac{s+\mu}{s(s+\lambda+\mu)}=\frac{(s+\lambda+\mu)\alpha+s\beta}{s(s+\lambda+\mu)}=\frac{s(\alpha+\beta)+\alpha(\lambda+\mu)}{s(s+\lambda+\mu)}. \qquad (2.18)$$

Невідомі коефіцієнти α та β в (2.17) знайдемо в результаті рішення системи рівнянь, які утворюються після прирівнювання коефіцієнтів при однакових ступенях в чисельнику дробів зліва і справа в (2.18). Маємо:

$$\begin{cases}\alpha+\beta=1,\\ \alpha\lambda+\alpha\mu=\mu.\end{cases}$$

Звідси:

$$\alpha=\frac{\mu}{\lambda+\mu},\ \beta=\frac{\lambda}{\lambda+\mu}. \qquad (2.19)$$

Підставляючи (2.19) в (2.17), отримаємо:

$$G^*_{00}(s)=\frac{s+\mu}{s(s+\lambda+\mu)}=\frac{\mu}{\lambda+\mu}\cdot\frac{1}{s}+\frac{\lambda}{\lambda+\mu}\cdot\frac{1}{s+\lambda+\mu}. \qquad (2.20)$$

Використовуючи таблицю відповідності функцій і їх перетворень Лапласа, запишемо:

$$G_{00}(t)=\frac{\mu}{\lambda+\mu}+\frac{\lambda}{\lambda+\mu}\cdot e^{-(\lambda+\mu)\cdot t}. \qquad (2.21)$$

Зворотні перетворення для зображень інших функцій, що описують закони розподілу ймовірностей, наведемо без пояснень.

$$G_{10}^*(s)=\frac{1}{s}\cdot\frac{\left(1-f_{01}^*(s)\right)\cdot f_{10}^*(s)}{1-f_{01}^*(s)\cdot f_{10}^*(s)}=\frac{1}{s}\cdot\frac{\left(1-\dfrac{\lambda}{s+\lambda}\right)\cdot\dfrac{\mu}{s+\mu}}{1-\dfrac{\lambda}{s+\lambda}\cdot\dfrac{\mu}{s+\mu}}=\frac{1}{s}\cdot\frac{s\mu}{s^2+s(\lambda+\mu)}=$$

$$=\frac{\mu}{s(s+\lambda+\mu)}=\frac{\alpha}{s}+\frac{\beta}{s+\lambda+\mu}=\frac{\alpha s+\alpha(\lambda+\mu)+\beta s}{s(s+\lambda+\mu)}=\frac{s(\alpha+\beta)+\alpha(\lambda+\mu)}{s(s+\lambda+\mu)},$$

$$\begin{cases}\alpha+\beta=0,\\ \alpha(\lambda+\mu)=\mu.\end{cases}$$

Звідси:

$$\alpha=\frac{\mu}{\lambda+\mu},\ \beta=\frac{\mu}{\lambda+\mu}.$$

$$G_{10}^*(s)=\frac{\mu}{\lambda+\mu}\cdot\frac{1}{s}-\frac{\mu}{\lambda+\mu}\cdot\frac{1}{s+\lambda+\mu}. \qquad (2.22)$$

$$G_{10}(t)=\frac{\mu}{\lambda+\mu}-\frac{\mu}{\lambda+\mu}\cdot e^{-(\lambda+\mu)\cdot t}. \qquad (2.23)$$

Далі:

$$G_{11}^*(s)=\frac{1}{s}\cdot\frac{1-f_{10}^*(s)}{1-f_{10}^*(s)\cdot f_{01}^*(s)}=\frac{1}{s}\cdot\frac{1-\dfrac{\mu}{s+\mu}}{1-\dfrac{\mu}{s+\mu}\cdot\dfrac{\lambda}{s+\lambda}}=\frac{s+\lambda}{s(s+\lambda+\mu)}=$$

$$=\frac{\alpha}{s}+\frac{\beta}{s+\lambda+\mu}=\frac{\alpha(s+\lambda+\mu)+\beta s}{s(s+\lambda+\mu)}=\frac{s(\alpha+\beta)+\alpha(\lambda+\mu)}{s(s+\lambda+\mu)}.$$

$$\begin{cases}\alpha+\beta=1,\\ \alpha(\lambda+\mu)=\lambda.\end{cases}$$

Звідси:

$$\alpha=\frac{\lambda}{\lambda+\mu},\ \beta=\frac{\mu}{\lambda+\mu}.$$

$$G_{11}^*(s)=\frac{\lambda}{\lambda+\mu}\cdot\frac{1}{s}+\frac{\mu}{\lambda+\mu}\cdot\frac{1}{s+\lambda+\mu}.$$

$$G_{11}(t) = \frac{\lambda}{\lambda+\mu} + \frac{\mu}{\lambda+\mu} \cdot e^{-(\lambda+\mu)\cdot t}. \qquad (2.24)$$

Нарешті,

$$G_{01}^*(s) = \frac{1}{s} \cdot \frac{\left(1 - f_{10}^*(s)\right) \cdot f_{01}^*(s)}{1 - f_{10}^*(s) \cdot f_{01}^*(s)} = \frac{1}{s} \cdot \frac{\left(1 - \frac{\mu}{s+\mu}\right) \cdot \frac{\lambda}{s+\lambda}}{1 - \frac{\mu}{s+\mu} \cdot \frac{\lambda}{s+\lambda}} = \frac{1}{s} \cdot \frac{s\lambda}{s^2 + s(\lambda+\mu)} =$$

$$= \frac{\lambda}{s(s+\lambda+\mu)} = \frac{\alpha}{s} + \frac{\beta}{s+\lambda+\mu} = \frac{\alpha(s+\lambda+\mu)+\beta s}{s(s+\lambda+\mu)} = \frac{s(\alpha+\beta)+\alpha(\lambda+\mu)}{s(s+\lambda+\mu)},$$

$$\begin{cases} \alpha + \beta = 0, \\ \alpha(\lambda+\mu) = \lambda. \end{cases}$$

Звідси:

$$\alpha = \frac{\lambda}{\lambda+\mu},\ \beta = -\frac{\lambda}{\lambda+\mu}.$$

$$G_{01}^*(s) = \frac{\lambda}{\lambda+\mu} \cdot \frac{1}{s} - \frac{\lambda}{\lambda+\mu} \cdot \frac{1}{s+\lambda+\mu}.$$

$$G_{01}(t) = \frac{\lambda}{\lambda+\mu} - \frac{\lambda}{\lambda+\mu} \cdot e^{-(\lambda+\mu)\cdot t}. \qquad (2.25)$$

Якщо початковий стан системи – нормальне функціонування, то результатом рішення задачі є наступні функції, що описують динаміку зміни станів:

$$G_{00}(t) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} \cdot e^{-(\lambda+\mu)\cdot t},$$

$$G_{01}(t) = \frac{\lambda}{\lambda+\mu} - \frac{\lambda}{\lambda+\mu} \cdot e^{-(\lambda+\mu)\cdot t}.$$

При цьому, природно, $G_{00}(t) + G_{01}(t) = 1$.

Отримані співвідношення визначають значення ймовірностей перебування системи в станах H_0 та H_1 в довільний момент часу t. З них,

зокрема, випливає, що ці значення асимптотично наближаються до своїх стаціонарних значень:

$$P(H_0)=\frac{\mu}{\lambda+\mu},\ P(H_1)=\frac{\lambda}{\lambda+\mu}. \tag{2.26}$$

Розглянемо тепер більш складну ситуацію, коли процеси функціонування системи є напівмарківськими.

Наведемо щільності розподілу тривалості перебування в кожному із станів системи до переходу в інший стан розподілами Ерланга другого порядку:

$$f_{01}(t)=t\lambda^2 e^{-\lambda t},$$

$$f_{10}(t)=t\mu^2 e^{-\mu t}.$$

Лапласові зображення цих функцій мають вигляд:

$$f_{01}^*(s)=\frac{\lambda^2}{(s+\lambda)^2},\ f_{10}^*(s)=\frac{\mu^2}{(s+\mu)^2}. \tag{2.27}$$

Зрозуміло, що з четвірки функцій $G_{00}(t)$, $G_{01}(t)$, $G_{10}(t)$, $G_{11}(t)$ практичний інтерес представляє тільки перша з них – $G_{00}(t)$. Відповідно до цього підставимо (2.27) в (2.11). При цьому отримаємо:

$$G_{00}^*(s)=\frac{1}{s}\cdot\frac{1-f_{01}^*(s)}{1-f_{01}^*(s)\cdot f_{10}^*(s)}=\frac{1}{s}\cdot\frac{1-\dfrac{\lambda^2}{(s+\lambda)^2}}{1-\dfrac{\lambda^2}{(s+\lambda)^2}\cdot\dfrac{\mu^2}{(s+\mu)^2}}=$$

$$=\frac{1}{s}\cdot\frac{\left[(s+\lambda)^2-\lambda^2\right]\cdot(s+\lambda)^2\cdot(s+\mu)^2}{(s+\lambda)^2\cdot\left[(s+\lambda)^2\cdot(s+\mu)^2-\lambda^2\cdot\mu^2\right]}= \tag{2.28}$$

$$=\frac{1}{s}\cdot\frac{(s^2+2s\lambda)\cdot(s+\mu)^2}{(s+\lambda)^2\cdot(s+\mu)^2-\lambda^2\cdot\mu^2}=$$

$$= \frac{(s+2\lambda)\cdot(s+\mu)^2}{\left[(s+\lambda)\cdot(s+\mu)-\lambda\mu\right]\cdot\left[(s+\lambda)\cdot(s+\mu)+\lambda\mu\right]} =$$

$$= \frac{(s+2\lambda)\cdot(s^2+2s\mu+\mu^2)}{s\cdot(s+\lambda+\mu)\cdot(s^2+s\cdot(\lambda+\mu)+2\lambda\mu)} =$$

$$= \frac{s^3+2s^2\cdot(\lambda+\mu)+s\cdot(4\lambda\mu+\mu^2)+2\lambda\mu^2}{s\cdot(s+\lambda+\mu)\cdot(s^2+s\cdot(\lambda+\mu)+2\lambda\mu)}.$$

Структура і аналітичне подання рішення задачі залежить від характеру коренів знаменника в (2.28). При цьому перші два кореня визначаються безпосередньо:

$$s_0 = 0,\ s_1 = -(\lambda+\mu).$$

Два останніх кореня отримаємо, вирішуючи рівняння:

$$s^2 + s(\lambda+\mu) + 2\lambda\mu = 0, \tag{2.29}$$

При цьому:

$$s_{2,3} = -\frac{\lambda+\mu}{2} \pm \sqrt{\frac{(\lambda+\mu)^2}{4} - 2\lambda\mu} = -\frac{\lambda+\mu}{2} \pm \sqrt{D}\,. \tag{2.30}$$

Якщо дискримінант $D > 0$, то рівняння (2.29) має два різних дійсних корені:

$$s_2 = -\frac{\lambda+\mu}{2} + \sqrt{D},\ s_3 = -\frac{\lambda+\mu}{2} - \sqrt{D}. \tag{2.31}$$

Якщо дискримінант $D = 0$, то корені $s_2 = s_3 = -\frac{\lambda+\mu}{2}$.

Якщо, нарешті, $D < 0$, то корні s_2 та s_3 – комплексні.

У всіх цих випадках вираз для зворотного перетворення Лапласа відшукується з використанням розкладання (2.28) на елементарні дроби.

Якщо всі корені дійсні і різні, то це розкладання має вигляд:

$$G_{00}^{*}(s)=\frac{\alpha_0}{s}+\frac{\alpha_1}{s-s_1}+\frac{\alpha_2}{s-s_2}+\frac{\alpha_3}{s-s_3}=$$

$$=\frac{\alpha_0(s-s_1)(s-s_2)(s-s_3)+\alpha_1 s(s-s_2)(s-s_3)}{s(s-s_1)(s-s_2)(s-s_3)}+$$

$$+\frac{\alpha_2 s(s-s_1)(s-s_3)+\alpha_3 s(s-s_1)(s-s_2)}{s(s-s_1)(s-s_2)(s-s_3)}= \tag{2.32}$$

$$=\frac{A(s)}{s(s-s_1)(s-s_2)(s-s_3)}.$$

Після перемноження і приведення подібних членів, це розкладання набуде вигляду:

$$A(s)=s^3(\alpha_0+\alpha_1+\alpha_2+\alpha_3)+$$

$$+s^2\left[-\alpha_0(s_1+s_2+s_3)+\alpha_1(s_2+s_3)+\alpha_2(s_1+s_3)+\right.$$

$$\left.+\alpha_3(s_1+s_2)\right]+ \tag{2.33}$$

$$+s\left[\alpha_0(s_1s_2+s_1s_3+s_2s_3)+\alpha_1s_2s_3+\alpha_2s_1s_3+\alpha_3s_1s_2\right]-\alpha_0s_1s_2s_3.$$

Тепер, прирівнюючи коефіцієнти при однакових степенях s в (2.28) та (2.33), отримаємо систему рівнянь щодо α_0, α_1, α_2, α_3:

$$\begin{cases}\alpha_0+\alpha_1+\alpha_2+\alpha_3=1,\\ -\alpha_0(s_1+s_2+s_3)+\alpha_1(s_2+s_3)+\alpha_2(s_1+s_3)+\alpha_3(s_1+s_2)=2(\lambda+\mu),\\ \alpha_0(s_1s_2+s_1s_3+s_2s_3)+\alpha_1s_2s_3+\alpha_2s_1s_3+\alpha_3s_1s_2=4\lambda\mu+\mu^2,\\ -\alpha_0s_1s_2s_3=2\lambda\mu^2.\end{cases} \tag{2.34}$$

Так як $D = \frac{(\lambda+\mu)^2}{4} - 2\lambda\mu$, то:

$$s_2 + s_3 = -\frac{\lambda+\mu}{2} + \sqrt{D} - \frac{\lambda+\mu}{2} - \sqrt{D} = -(\lambda+\mu),$$

$$s_1 + s_3 = -(\lambda+\mu) - \frac{\lambda+\mu}{2} + \sqrt{D} = -\frac{3}{2}(\lambda+\mu) + \sqrt{D},$$

$$s_1 + s_2 = -(\lambda+\mu) - \frac{\lambda+\mu}{2} - \sqrt{D} = -\frac{3}{2}(\lambda+\mu) - \sqrt{D},$$

$$s_1 + s_2 + s_3 = -2(\lambda+\mu), \tag{2.35}$$

$$s_2 s_3 = \left(\frac{\lambda+\mu}{2}\right)^2 - \left[\left(\frac{\lambda+\mu}{2}\right)^2 - 2\lambda\mu\right] = 2\lambda\mu,$$

$$s_1 s_2 = -(\lambda+\mu)\cdot\left(-\frac{\lambda+\mu}{2} - \sqrt{D}\right) = \frac{(\lambda+\mu)^2}{2} + (\lambda+\mu)\sqrt{D},$$

$$s_1 s_3 = -(\lambda+\mu)\cdot\left(-\frac{\lambda+\mu}{2} + \sqrt{D}\right) = \frac{(\lambda+\mu)^2}{2} - (\lambda+\mu)\sqrt{D},$$

$$s_1 s_2 s_3 = -(\lambda+\mu)\cdot 2\lambda\mu.$$

З урахуванням (2.35) розв'язок системи рівнянь (2.34) має вигляд:

$$\alpha_0 = \frac{\mu}{\lambda+\mu},$$

$$\alpha_1 = -\frac{\lambda(\lambda-\mu)}{2\mu(\lambda+\mu)}, \tag{2.36}$$

$$\alpha_2 = \alpha_3 = \frac{\lambda}{4\mu} \mp \frac{\lambda(\lambda-3\mu)}{4\mu\sqrt{\lambda^2 - 6\lambda\mu + \mu^2}}.$$

Тепер, використовуючи (2.32) та (2.36), запишемо результат зворотного перетворення $G_{00}^*(s)$:

$$G_{00}(t) = \alpha_0 + \alpha_1 e^{-(\lambda+\mu)t} + \alpha_2 e^{-\left(\frac{\lambda+\mu}{2}+\sqrt{D}\right)t} + \alpha_3 e^{-\left(\frac{\lambda+\mu}{2}-\sqrt{D}\right)t}. \tag{2.37}$$

Так як у випадку, що розглядається, $\left(\frac{\lambda+\mu}{4}\right)^2 > 2\lambda\mu$, то шукана (2.37) ймовірність перебування системи в стані H_0 в момент часу t має стаціонарне значення, рівне, як і в марківському випадку, $P(H_0) = \frac{\mu}{\lambda+\mu}$.

Нехай тепер $D < 0$. При цьому корені рівняння (2.29) будуть рівні:

$$s_1 = -(\lambda+\mu),$$

$$s_2 = -\frac{(\lambda+\mu)}{2} + i\sqrt{|D|},$$

$$s_3 = -\frac{(\lambda+\mu)}{2} - i\sqrt{|D|},$$

а розкладання (2.28) на елементарні множники буде мати вигляд:

$$G^*_{00}(s) = \frac{\alpha_0}{s} + \frac{\alpha_1}{s-s_1} + \frac{\alpha_2 s + \alpha_3}{s^2 - s(s_2+s_3) + s_2 s_3} =$$

$$= \frac{\alpha_0(s-s_1)(s^2 - s(s_2+s_3) + s_2 s_3)}{s(s-s_1)(s^2 - s(s_2+s_3) + s_2 s_3)} +$$

$$+ \frac{\alpha_1 s(s^2 - s(s_2+s_3) + s_2 s_3) + \alpha_2 s^2(s-s_1) + \alpha_3 s(s-s_1)}{s(s-s_1)(s^2 - s(s_2+s_3) + s_2 s_3)} = \quad (2.38)$$

$$= \frac{1}{B}\left[s^3(\alpha_0+\alpha_1+\alpha_2) - s^2(\alpha_0(s_1+s_2+s_3) + \alpha_1 s_2 + \alpha_1 s_3 + \alpha_2 s_1 - \alpha_3) +\right.$$

$$\left. + s(\alpha_0(s_1 s_2 + s_1 s_3 + s_2 s_3) + \alpha_1 s_2 s_3 - \alpha_3 s_1) - \alpha_0 s_1 s_2 s_3\right],$$

$$B = s(s-s_1)(s^2 - s(s_2+s_3) + s_2 s_3).$$

Після прирівнювання коефіцієнтів при однакових степенях в (2.28) і (2.38), отримаємо систему рівнянь:

$$\begin{cases} \alpha_0 + \alpha_1 + \alpha_2 = 1, \\ -\alpha_0(s_1 + s_2 + s_3) - \alpha_1 s_2 - \alpha_1 s_3 - \alpha_2 s_1 + \alpha_3 = 2(\lambda + \mu), \\ \alpha_0(s_1 s_2 + s_1 s_3 + s_2 s_3) + \alpha_1 s_2 s_3 - \alpha_3 s_1 = 4\lambda\mu + \mu^2, \\ -\alpha_0 s_1 s_2 s_3 = 2\lambda\mu^2. \end{cases},$$

Вирішення цієї системи:

$$\alpha_0 = \frac{\mu}{\lambda + \mu},\ \alpha_1 = -\frac{\lambda(\lambda - \mu)}{2\mu(\lambda + \mu)}, \alpha_2 = \frac{\lambda}{2\mu},\ \alpha_3 = \lambda.$$

Таким чином:

$$G_{00}^*(s) = \frac{\mu}{\lambda + \mu} \cdot \frac{1}{s} - \frac{\lambda(\lambda - \mu)}{2\mu(\lambda + \mu)} \frac{1}{s - s_1} + \frac{\frac{\lambda}{2\mu} \cdot s + \lambda}{s^2 - s(s_2 + s_3) + s_2 s_3}. \qquad (2.39)$$

Приведемо третій доданок в (2.39) до виду, зручного для виконання зворотного перетворення Лапласа. При цьому:

$$s^2 - s(s_2 + s_3) + s_2 s_3 =$$

$$= s^2 - 2s\frac{s_2 + s_3}{2} + \frac{(s_2 + s_3)^2}{4} + s_2 s_3 - \frac{(s_2 + s_3)^2}{4} =$$

$$= \left(s - \frac{s_2 + s_3}{2}\right)^2 - \left(\frac{s_2 + s_3}{2}\right)^2 + 2\lambda\mu = \qquad (2.40)$$

$$= \left(s + \frac{\lambda + \mu}{2}\right)^2 - \frac{(\lambda + \mu)^2}{4} + 2\lambda\mu = (s + b)^2 + a^2, \quad a > 0.$$

Далі:

$$\alpha_2 s + \alpha_3 = \alpha_2(s + b - b) + \alpha_3 = \alpha_2(s + b) + (\alpha_3 - \alpha_2 b). \qquad (2.41)$$

З врахуванням (2.40) та (2.41) співвідношення (2.39) набуває вигляду:

$$G_{00}^{*}(s)=\frac{\mu}{\lambda+\mu}\cdot\frac{1}{s}-\frac{\lambda(\lambda-\mu)}{2\mu(\lambda+\mu)}\cdot\frac{1}{s-s_1}+\frac{\lambda}{2\mu}\cdot\frac{s+\frac{\lambda+\mu}{2}}{(s+b)^2+a^2}+$$
$$+(\lambda-b\cdot\frac{\lambda}{2\mu})\cdot\frac{1}{(s+b)^2+a^2}. \qquad (2.42)$$

Отримаємо результат зворотного перетворення $G_{00}^{*}(s)$:

$$G_{00}(t)=\alpha_0+\alpha_1\cdot e^{-(\lambda+\mu)t}+\alpha_2\cdot\cos at\cdot e^{-\frac{(\lambda+\mu)}{2}t}+\frac{\alpha_3-\alpha_2\cdot b}{a}\cdot\sin at\cdot e^{-\frac{(\lambda+\mu)}{2}t}. \qquad (2.43)$$

Рішення задачі завершене. Отримане співвідношення визначає ймовірність перебування системи в стані H_0 в будь-який момент часу *t*. Якщо, як в розглянутому окремому випадку, проводиться надійнісний аналіз відновлюваної системи, то це співвідношення дозволяє обґрунтовано сформулювати і вирішити оптимізаційну задачу підвищення ефективності системи з використанням стандартних технологій управління параметрами.

2.2. Обґрунтування вибору шляхів забезпечення стійкості рекомендаційних систем до дестабілізуючих факторів

Стійкість (робастність) системи в комп'ютерній інженерії в загальному випадку означає міру здатності обчислювальної системи відновлюватися при виникненні помилкових ситуацій як зовнішнього, так і внутрішнього походження [124].

Під стійкістю (робастністю) рекомендаційної системи часто розуміють [61, 76, 101] збереження точності рекомендацій за наявності підробленої інформації, яка зазвичай додається до системи навмисне з метою впливу на рекомендації користувачам. Для просування свого контенту на певному веб-ресурсі, треті особи можуть створити мережу ботів, що будуть викривляти рейтинги об'єктів системи, а отже, і ймовірність та кількість потраплянь їх у рекомендації звичайним користувачам системи. Такі дії, як правило, називають атакою на рекомендаційну систему [61, 62, 69, 76, 98, 103, 104].

У даній роботі будемо виходити з більш широкого визначення для поняття стійкість та сформулюємо стійкість рекомендаційних систем наступним чином.

Стійкість рекомендаційної системи – це міра здатності рекомендаційної системи створювати релевантні та точні рекомендації користувачам системи не зважаючи на зовнішні впливи у вигляді інформаційних атак, та внутрішні проблеми, такі, наприклад, як проблема холодного старту та постійного холодного старту, розрідженості даних, появи некоректної інформації тощо.

Таким чином задачу створення стійкої рекомендаційної системи слід розділити на дві:

1. Забезпечити стійкість рекомендаційної системи до інформаційних атак.

2. Забезпечити стійкість рекомендаційної системи до внутрішніх проблем системи, таких як проблема холодного страту, розрідженості даних тощо.

Оцінити стійкість рекомендаційної системи можна вимірюючи показники точності її роботи до та після виникнення дестабілізуючих факторів. Чим менша різниця між цими двома одержаними значеннями точності, тим вища стійкість системи до тих дестабілізуючих факторів, які використовувалися у вимірюваннях.

При вимірюванні стійкості рекомендаційної системи можуть виникнути наступні складнощі. У реальних умовах роботи веб-ресурсу з рекомендаційною системою не завжди можна спостерігати ситуації породжені дестабілізуючими факторами. Наприклад, важко передбачити здійснення справжнього нападу на реальну систему і скористатися даним фактом для проведення випробувань системи, тому зазвичай такі стани системи моделюють за допомогою програмних симуляцій [40, 76]. Так, наприклад, у роботах [40, 76] експерименти проводяться на основі реальних даних з відкритих датасетів з додаванням у них даних створених бот-

мережами, отриманих за допомогою програмного моделювання атак на рекомендаційну систему, що дозволяє визначити стійкість системи до різних видів атак та емпірично вимірювати середню вартість успішної атаки.

При оцінюванні стійкості системи слід враховувати, що вона буде різною по відношенню до різних дестабілізуючих факторів, зокрема, стійкість системи може значно відрізнятися для різних видів інформаційних атак.

Якщо необхідно розробити стійку до дестабілізуючих факторів рекомендаційну систему, то задачу оптимізації можна сформулювати наступним чином:

$$d(R_b, R_a) \to \min, \tag{2.44}$$

де вектор R_b містить список прогнозованих рекомендацій (оцінок) для користувачів, сформований у нормальному режимі роботи системи; а вектор R_a містить список прогнозованих рекомендацій (оцінок) для користувачів, сформований під час дії дестабілізуючих факторів.

Взагалі, створити рекомендаційну систему, стійку до всіх видів інформаційних атак та внутрішніх помилок, складно та затратно. Тому більш доцільно оцінити вартість збитків від виникнення дестабілізуючих факторів та вартість усунення даних факторів, щоб на основі даної інформації приймати рішення про доцільність внесення тих чи інших змін у систему, направлених на підвищення її стійкості.

2.2.1. Дослідження методів оцінювання стійкості рекомендаційних систем

Існуючі методи оцінки стійкості (робастності) рекомендаційних систем направлені на визначення стійкості системи до інформаційних атак [76, 101, 178]. Хоча дані методи, після деякої адаптації, можна застосовувати і до визначення стійкості й до внутрішніх дестабілізуючих факторів. Отже,

розглянемо існуючі методи оцінювання стійкості рекомендаційних систем до інформаційних атак.

Оскільки мета інформаційних атак на рекомендаційні системи – це зміна рейтингу цільового об'єкту, то для вимірювання стійкості рекомендаційної системи, потрібно оцінити, наскільки змінилися рейтинги об'єктів системи після атаки.

Також показники стійкості повинні фіксувати відмінності в рекомендованому статусі цільового об'єкту до та після атаки, тобто, визначати наскільки змінилася частота потрапляння цільового об'єкту у списки рекомендацій користувачам та чи покращилися його позиції у даних списках. Ці дані важливіші за дані про зміни у рейтингах, оскільки показують ефективність атаки. Але показники, спрямовані на виявлення змін у рекомендованому статусі об'єкту, можуть не зафіксувати невдалу атаку на відміну від показників орієнтованих на виявлення змін у рейтингах, так як незначна зміна рейтингу об'єкту системи може не вплинути на його рекомендований статус, але може бути зафіксована.

Багато дослідників використовують [69, 76] середній зсув прогнозування для оцінювання змін у прогнозованих рейтингах.

Нехай U_T та I_T – це набори користувачів та об'єктів системи. Для кожної пари user-item (u, i) зсув прогнозування може бути виміряний як:

$$\Delta_{u,i} = p'_{u,i} - p_{u,i}, \tag{2.45}$$

де p і p' є прогнози до- та після атаки відповідно.

Позитивне значення $\Delta_{u,i}$ означає, що атака зуміла збільшити прогнозовані рейтинги об'єкту, а негативне – зменшити їх. Середній зсув прогнозування рейтингів об'єкта i для всіх користувачів можна обчислити наступним чином:

$$\Delta_i = \sum_{u \in U_T} \frac{\Delta_{i,u}}{|U_T|}, \tag{2.46}$$

де $|U_T|$ – кількість елементів у наборі користувачів U_T.

Аналогічно середній зсув прогнозування рейтингів для всіх об'єктів у тестовій виборці може бути обчислений як:

$$\overline{\Delta} = \sum_{i \in I_T} \frac{\Delta_i}{|I_T|}, \tag{2.47}$$

де $|I_T|$ – кількість елементів у наборі об'єктів I_T.

Зсув прогнозування дозволяє дослідити як атаки впливають на рейтинги цільових об'єктів. Однак навіть дуже сильні зміни у рейтингу об'єкту можуть не змінити його рекомендований статус. Така ситуація може виникнути, наприклад, якщо його початкова середня оцінка дуже низька, що навіть сильний її приріст недостатній для потрапляння у списки рекомендацій.

Для оцінювання впливу атаки на списки рекомендацій існує наступний показник – коефіцієнт звернень [46, 69, 76]. Нехай R_u – це набір найпопулярніших N рекомендацій для користувача u. Якщо цільовий об'єкт потрапляє в R_u, для користувача u, функція оцінювання результату атаки $H_{u,i}$ має значення 1; інакше – 0. Коефіцієнт звернень для елемента i визначається як:

$$HitRatio_i = \sum_{u \in U_T} \frac{H_{i,u}}{|U_T|}. \tag{2.48}$$

Середнє значення коефіцієнту звернень може бути обчислене як:

$$\overline{HitRatio} = \sum_{i \in I_T} \frac{HitRatio_i}{|I_T|}. \tag{2.49}$$

Для оцінювання стійкості різних методів колаборативної фільтрації формується два набори тестових даних, одні без додавання профілів, які моделюють атаку (профілів ботів), інші з додаванням таких профілів. Для кожного набору даних створюються списки рекомендацій, а потім обчислюються вищерозглянуті показники стійкості. Чим менша різниця між прогнозуванням вподобань, яку можна оцінити значеннями вищерозглянутих показників стійкості, для першого та другого наборів даних, тим більш стійка до дестабілізуючих факторів розглядувана рекомендаційна система.

2.2.2. Дослідження шляхів забезпечення стійкості рекомендаційних систем

Вибір шляхів забезпечення стійкості рекомендаційних систем залежить від того, до яких дестабілізуючих факторів слід її забезпечити.

Стійкість системи до холодного старту та низької якості вхідних даних. Якщо до системи додається новий користувач або об'єкт, про якого ще немає даних або їх дуже мало, для формування рекомендацій можна використовувати контекстну інформацію, інформацію про найрейтинговіші об'єкти системи тощо. В будь-якому випадку треба використовувати та поєднувати різні алгоритми фільтрації даних, серед яких є такі, що використовують контекстну та групову інформацію.

Стійкість системи до атак накручування рейтингів. Побудова стійких до інформаційних атак рекомендаційних систем базується на двох принципах: 1) виявлення спаму серед дій користувачів; 2) невраховування при побудові рекомендацій даних користувачів, що поширюють спам. На сьогоднішній день існують різні методи, які дозволяють виявити атаку на рекомендаційну систему, вони базуються на пошуку профілів спам-користувачів (ботів). Так як при атаці створюють не один фейковий профіль, а багато схожих, такі спам-користувачі будуть мати незвично високу схожість між собою та аномальну поведінку, у порівнянні зі звичайними користувачами, що часто дає можливість їх виявити. Однак, надійні методи виявлення атак на рекомендаційні системи та протидії ним все ще залишаються активною областю досліджень.

Стійкість до атак на приватність користувачів. Для забезпечення приватності користувача рекомендаційної системи можна застосовувати наступні методи:

– Інформування користувачів про те, яку інформацію про них збирає рекомендаційна система, гнучкі налаштування параметрів конфіденційності.

– Анонімізація – інформація про користувача може частково видалятися

або піддаватися обфускації (маскуванню) користувачем або власником рекомендаційної системи.

– Рандомізація – дані користувача (наприклад, виставлені об’єктам оцінки) можуть бути частково зашумлені випадковими значеннями. Необхідний рівень шуму залежить від того, як часто дані будуть використовуватися, і передбачає балансування між точністю прогнозування та конфіденційністю користувача.

– Шифрування даних користувача, що зберігаються в базі даних рекомендаційної системи.

Підвищувати стійкість рекомендаційної системи до зовнішніх дестабілізуючих факторів можна за допомогою наступних підходів [22, 28, 46, 61, 69, 98, 101]:

1. Заміняти одні методи створення списків рекомендацій на інші більш стійкі до дестабілізуючих факторів аналоги.

2. Використовувати методи виявлення профілів ботів та вилучати їх дані з розрахунків для формування рекомендацій користувачам системи.

3. Впровадити та враховувати у рекомендаційній системі параметр репутація та/або експертність для користувачів таким чином, щоб користувачі з низькою репутацією (експертністю), визначеною на основі їх попередніх дій, слабо впливали або не впливали на формування списків рекомендацій.

Висновки до розділу 2

У даному розділі запропоновано метод визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах. Основу методу становить модель динаміки ймовірностей станів системи. Модель містить набір інтегральних рівнянь щодо невідомих функцій, що описують ймовірнісну динаміку системи (2.2)-(2.5). Розв’язання цих інтегральних рівнянь отримано з використанням перетворення Лапласа

(2.6)-(2.14). В результаті рішення інтегральних рівнянь отримано шукане співвідношення (2.43) для розрахунку умовної ймовірності знаходження рекомендаційної системи в стані H_0 в довільний момент часу t, якщо в початковий момент часу об'єкт знаходився в стані H_0. Таким чином, запропонований метод, на відміну від відомих, дозволяє не тільки розрахувати фінальний розподіл ймовірностей системи, але і значення ймовірності будь-якого стану в довільний момент часу t. Отримані співвідношення, по-перше, дають можливість вирішувати задачі оцінки ефективності системи в залежності від значень задаваємого набору її параметрів. По-друге, вони можуть бути використані для оптимізації управління розподілом обмеженого ресурсу з метою підвищення ефективності системи.

Відзначимо, що запропонований метод узагальнюється за наступними напрямками.

По-перше, цей метод може бути застосований, якщо аналізована система має $m > 2$ станів. При цьому необхідно ввести і аналітично описати m^2 умовних ймовірностей знаходження системи в кожному із станів в момент часу t, за умови, що в початковий момент часу система перебувала в будь-якому іншому стані. Важливо, що складність вирішення цієї задачі не залежить від числа можливих станів системи.

По-друге, при вирішенні задачі статистичної обробки вихідних даних про тривалість перетворення системи в кожному з можливих станів доцільно використовувати більш адекватну, ніж наведену вище, модель – розподіл Ерланга довільного порядку n, тобто, $f(t) = \frac{t^{n-1}\lambda^n}{(n-1)!}e^{-\lambda t}$. Точність апроксимації гістограм, природно, зростає, але складність вирішення задачі практично залишається незмінною завдяки тому, що перетворення Лапласа розподілу Ерланга порядку n має вигляд: $L\left(f(t)\right) = \frac{\lambda^n}{(s-\lambda)^n}$. При цьому виникає потреба у відшуканні коренів алгебраїчного рівняння степені $2n$,

$n > 2$. Зрозуміло, аналітичне рішення цієї задачі неможливе, проте чисельне здійсненне завжди, що істотно підвищує практичну застосовність запропонованого методу.

Також у даному розділі досліджено поняття стійкості рекомендаційних систем та способи її оцінювання. Обґрунтовано шляхи забезпечення стійкості рекомендаційної системи до внутрішніх та зовнішніх дестабілізуючих факторів.

РОЗДІЛ 3.
МОДЕЛЬ ТА МЕТОДИ ЗАБЕЗПЕЧЕННЯ СТІЙКОСТІ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДО ВНУТРІШНІХ ДЕСТАБІЛІЗУЮЧИХ ФАКТОРІВ

У даному розділі запропоновано математичну модель стійкої рекомендаційної системи для оптимізації загальних витрат на обслуговування системи та стійкі методи колаборативної фільтрації в умовах внутрішніх дестабілізуючих факторів. Наведено результати експериментів для визначення точності та стійкості розроблених методів та порівняння їх з відомими методами колаборативної фільтрації.

3.1. Розробка математичної моделі стійкої рекомендаційної системи в умовах внутрішніх дестабілізуючих факторів

Рекомендаційні системи, що використовують колаборативну фільтрацію, повинні час від часу оновлювати дані, на основі яких вони здійснюють прогнози вподобань користувачів для формування рекомендацій, так як ці дані обчислюються на основі матриці рейтингів, яка весь час поповнюється новими даними.

Для різних моделей колаборативної фільтрації ці дані будуть різними:

– для моделей сусідства – коефіцієнти подоби;

– для факторизаційних моделей – приховані фактори.

Ці додаткові дані можна обчислювати за наступними схемами:

1) тільки один раз – для систем, що майже не оновлюються, а таких практично немає.

2) кожного разу при формуванні нових рекомендацій – дуже збитково, може бути доцільним тільки, якщо враховувати не всі дані при перерахунках (використовувати часове вікно або окіл схожих користувачів біля кожного користувача), але це може по різному впливати на точність рекомендацій.

3) з певною періодичністю – якщо підібрати оптимальну частоту, можна досягти достатньої точності при допустимих збитках.

Швидке зростання кількості елементів рекомендаційної системи і часта зміна їх характеристик робить неактуальною раніше зібрані та обчислені вхідні дані для формування рекомендацій. Така проблема швидкої втрати актуальності вхідних даних може бути віднесена до внутрішніх дестабілізуючих факторів.

Проведені дослідження показали, що існуючі роботи або взагалі не розглядають дану проблему [35-37, 78, 82, 92, 209], або використовують підходи з пункту (2), інколи поєднуючи їх з розпаралелюванням обчислень, кешуванням даних та зменшенням розмірності вхідних даних [76, 194].

Було розроблено математичну модель, що дасть можливість використовувати підхід з пункту (3).

При розробці моделі стійкої рекомендаційної системи в умовах внутрішніх дестабілізуючих факторів було використано раніше запропонований у даній роботі метод визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах [58].

Було побудовано ланцюг Маркова для можливих станів стійкої рекомендаційної системи в умовах внутрішніх дестабілізуючих факторів, представлених проблемою швидкої втрати актуальності вхідних даних (рис. 3.1). Запропоновано наступний набір станів рекомендаційної системи з точки зору зміни вподобань користувачів:

1) **Вхідні дані актуальні (H_1)** – коефіцієнти подоби (чи приховані фактори) відповідають дійсності. В цьому режимі витрачаються ресурси $v_1 t_1 L_1$ на залишення системи в стані H_1 пропорційно часу роботи системі в цьому стані та інтенсивності перерахунку вхідних даних.

2) **Вхідні дані не актуальні (H_2)** – коефіцієнти подоби (чи приховані фактори) не достатньо точно відображають реальні дані. В цьому режимі роботи накопичуються збитки $t_2 L_2$ від низької точності рекомендацій пропорційно часу. Також в цьому режимі додатково витрачаються ресурси

$v_3 t_3 L_3$ на організацію повернення роботи системи в нормальний H_1 пропорційно часу роботи системі в поточному стані та інтенсивності перерахунку даних (коефіцієнтів подоби/прихованих факторів).

Ланцюг Маркова для даної стійкої рекомендаційної системи в умовах внутрішніх дестабілізуючих факторів показано на рис. 3.1:

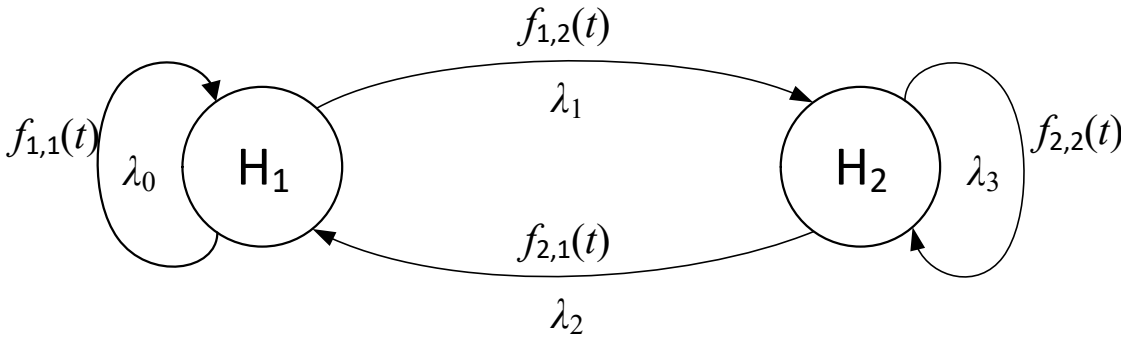


Рис. 3.1. Ланцюг Маркова для стійкої рекомендаційної системи в умовах внутрішніх дестабілізуючих факторів

Було знайдено співвідношення для розрахунку умовних ймовірностей знаходження системи в можливих станах в довільний момент часу *t*, якщо в початковий момент часу об'єкт знаходився в стані H_0.

Виходячи з запропонованого методу визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах, реалізацію якого у попередньому розділі було розглянуто саме на системі з двома станами, якщо система є марківською, ймовірності перебування її у своїх станах будуть наступними:

$$G_{11}(t) = \frac{\lambda_0}{\lambda_1 + \lambda_0} + \frac{\lambda_1}{\lambda_1 + \lambda_0} \cdot e^{-(\lambda_1 + \lambda_0) \cdot t}, \tag{3.1}$$

$$G_{12}(t) = \frac{\lambda_1}{\lambda_1 + \lambda_0} - \frac{\lambda_1}{\lambda_1 + \lambda_0} \cdot e^{-(\lambda_1 + \lambda_0) \cdot t}, \tag{3.2}$$

$$G_{21}(t) = \frac{\lambda_3}{\lambda_2 + \lambda_3} - \frac{\lambda_3}{\lambda_2 + \lambda_3} \cdot e^{-(\lambda_2 + \lambda_3) \cdot t}, \tag{3.3}$$

$$G_{22}(t) = \frac{\lambda_2}{\lambda_2 + \lambda_3} + \frac{\lambda_3}{\lambda_2 + \lambda_3} \cdot e^{-(\lambda_2 + \lambda_3) \cdot t}. \tag{3.4}$$

А якщо система є напівмарківською, ймовірності станів можна визначити за формулами, що наведені нижче.

Ймовірність G_{11} буде визначатися однією з формул (3.6)-(3.8) в залежності від значень дискримінанта (3.5).

$$D = \frac{(\lambda_0 + \lambda_1)^2}{4} - 2\lambda_0\lambda_1 . \tag{3.5}$$

Якщо дискримінант $D > 0$, то:

$$G_{11}(t) = \alpha_0 + \alpha_1 e^{-(\lambda_1+\lambda_0)t} + \alpha_2 e^{-\left(\frac{\lambda_1+\lambda_0}{2}+\sqrt{D}\right)t} + \alpha_3 e^{-\left(\frac{\lambda_1+\lambda_0}{2}-\sqrt{D}\right)t}, \tag{3.6}$$

де $\alpha_0 = \frac{\lambda_0}{\lambda_1+\lambda_0}$, $\alpha_1 = -\frac{\lambda_1(\lambda_1-\lambda_0)}{2\lambda_0(\lambda_1+\lambda_0)}$, $\alpha_2 = \alpha_3 = \frac{\lambda_1}{4\lambda_0} \mp \frac{\lambda_1(\lambda_1-3\lambda_0)}{4\lambda_0\sqrt{\lambda_1{}^2 - 6\lambda_1\lambda_0 + \lambda_0{}^2}}$.

Якщо дискримінант $D = 0$, то:

$$G_{11}(t) = \alpha_0 + \alpha_1 \cdot e^{-\frac{(\lambda_1+\lambda_0)}{2}t} + \left(\left(-\alpha_2 \cdot \frac{(\lambda_1+\lambda_0)}{2} + \alpha_3\right)\cdot t + \alpha_2\right)\cdot e^{-\frac{(\lambda_1+\lambda_0)}{2}t}, \tag{3.7}$$

де $\alpha_0 = \frac{1}{2} - \frac{\sqrt{2}}{4}$, $\alpha_1 = -\left(1 + \frac{3\sqrt{2}}{4}\right)$, $\alpha_2 = \frac{2}{3} + \sqrt{2}$, $\alpha_3 = \left(3 + 2\sqrt{2}\right)\lambda_0$.

Якщо, $D < 0$, то:

$$G_{11}(t) = \alpha_0 + \alpha_1 \cdot e^{-(\lambda_1+\lambda_0)t} + \alpha_2 \cdot \cos at \cdot e^{-\frac{(\lambda_1+\lambda_0)}{2}t} + \frac{\alpha_3 - \alpha_2 \cdot b}{a} \cdot \sin at \cdot e^{-\frac{(\lambda_1+\lambda_0)}{2}t}, \tag{3.8}$$

де $\alpha_0 = \frac{\lambda_0}{\lambda_1+\lambda_0}$, $\alpha_1 = -\frac{\lambda_1(\lambda_1-\lambda_0)}{2\lambda_0(\lambda_1+\lambda_0)}$, $\alpha_2 = \frac{\lambda_1}{2\lambda_0}$, $\alpha_3 = \lambda_1$.

Ймовірність G_{22} буде визначатися однією з формул (3.10)-(3.12) в залежності від значень дискримінанта (3.9).

$$D = \frac{(\lambda_2 + \lambda_3)^2}{4} - 2\lambda_2\lambda_3 . \tag{3.9}$$

Якщо $D > 0$, то:

$$G_{22}(t) = \alpha_0 + \alpha_1 e^{-(\lambda_2+\lambda_3)t} + \alpha_2 e^{-\left(\frac{\lambda_2+\lambda_3}{2}+\sqrt{D}\right)t} + \alpha_3 e^{-\left(\frac{\lambda_2+\lambda_3}{2}-\sqrt{D}\right)t}, \quad (3.10)$$

де $\alpha_0 = \frac{\lambda_3}{\lambda_2+\lambda_3}$, $\alpha_1 = -\frac{\lambda_2(\lambda_2-\lambda_3)}{2\lambda_3(\lambda_2+\lambda_3)}$, $\alpha_2 = \alpha_3 = \frac{\lambda_2}{4\lambda_3} \mp \frac{\lambda_2(\lambda_2-3\lambda_3)}{4\lambda_3\sqrt{\lambda_2{}^2 - 6\lambda_2\lambda_3 + \lambda_3{}^2}}$.

Якщо $D = 0$, то:

$$G_{22}(t) = \alpha_0 + \alpha_1 \cdot e^{-\frac{(\lambda_2+\lambda_3)}{2}t} + \left(\left(-\alpha_2 \cdot \frac{(\lambda_2+\lambda_3)}{2} + \alpha_3\right)\cdot t + \alpha_2\right)\cdot e^{-\frac{(\lambda_2+\lambda_3)}{2}t}, \quad (3.11)$$

де $\alpha_0 = \frac{1}{2} - \frac{\sqrt{2}}{4}$, $\alpha_1 = -\left(1 + \frac{3\sqrt{2}}{4}\right)$, $\alpha_2 = \frac{2}{3} + \sqrt{2}$, $\alpha_3 = \left(3 + 2\sqrt{2}\right)\lambda_3$.

Якщо, $D < 0$, то:

$$G_{22}(t) = \alpha_0 + \alpha_1 \cdot e^{-(\lambda_2+\lambda_3)\cdot t} + \alpha_2 \cdot \cos at \cdot e^{-\frac{(\lambda_2+\lambda_3)}{2}t} + \\ + \frac{\alpha_3 - \alpha_2 \cdot b}{a} \cdot \sin at \cdot e^{-\frac{(\lambda_2+\lambda_3)}{2}t}, \quad (3.12)$$

де $\alpha_0 = \frac{\lambda_3}{\lambda_2+\lambda_3}$, $\alpha_1 = -\frac{\lambda_2(\lambda_2-\lambda_3)}{2\lambda_3(\lambda_2+\lambda_3)}$, $\alpha_2 = \frac{\lambda_2}{2\lambda_3}$, $\alpha_3 = \lambda_2$.

Ймовірність G_{12} буде визначатися за формулою (3.13):

$$G_{12}(t) = 1 - G_{11}. \quad (3.13)$$

А ймовірність G_{21} – за формулою (3.14):

$$G_{21}(t) = 1 - G_{22}. \quad (3.14)$$

Було розроблено спосіб визначення повних витрат системи. У стані H_1 система зазнає витрат $З_1 = tK_1\lambda_0$. Нехай, в ситуації, коли система перебуває у стані H_2, завдяки неправильно створеним рекомендаціям, втрачаються прибутки пропорційно часу перебування в цьому стані $З_2 = tK_2$. Якщо проводити перерахунок коефіцієнтів подоби (прихованих факторів) частіше, то потрібно використати додаткові обчислювальні ресурси, що можна виразити витратами $З_3 = tK_3\lambda_2$, які пропорційні не лише часу перебування

системи у стані H_2, а й інтенсивності перерахунку даних. Таким чином:

$$t_1 = G_{1,1} \ , \ t_2 = G_{2,2} \ , \ t_3 = G_{2,2} + G_{2,1} \tag{3.15}$$

Відповідно, повні витрати системи складатимуть:

$$L = G_{1,1} \cdot K_1 \cdot \lambda_1 + G_{2,2} \cdot K_2 + \left(G_{2,2} + G_{2,1}\right) K_3 \lambda_2 . \tag{3.16}$$

Запропоновано спосіб оптимізації частоти перерахунку вхідних даних. В рекомендаційній системі можна керувати частотою перерахунку вхідних даних v, що відповідає за значення параметру λ_2. Для того, щоб визначити оптимальну частоту перерахунку v_{opt}, треба знайти таке λ_2, при якому загальні збитки L будуть мінімальними. Тому система має мінімальну збитковість при:

$$v_{opt} = \arg\min_{\lambda_2} L(\lambda_1, \lambda_2) = \arg\min_{\lambda_2} \left[G_{1,1} \cdot K_1 \cdot \lambda_1 + G_{2,2} \cdot K_2 + \left(G_{2,2} + G_{2,1}\right) K_3 \lambda_2 \right]. \tag{3.16}$$

На жаль, система є нелінійною, тому її розв'язання в загальному випадку можливе лише за допомогою чисельних методів.

Для прикладу, зафіксуємо значення K_1=1, K_2=5, K_3=3; а після проведемо табуляцію функції витрат для:

$$\lambda_1 = 0.05 \ \lambda_2 = 0.05;\ 0.1;\ 0.15;\ 0.2;\ 0.25;\ 0.3;\ 0.35.$$

Для марківської моделі одержимо наступні дані, наведені у таблиці 3.1.

Таблиця 3.1. Приклад розрахунків для визначення оптимальної частоти перерахунку вхідних даних λ_2 для стійкої рекомендаційної системи

$v=\lambda_2$	0.05000	0.10000	0.15000	0.20000	0.25000	0.30000	0.35000	0.40000	0.45000
$G_{1,1}$	0.41770	0.41770	0.41770	0.41770	0.41770	0.41770	0.41770	0.41770	0.41770
$G_{2,2}$	1.00000	0.44440	0.34781	0.28571	0.24242	0.21052	0.18604	0.16666	0.15094
$G_{2,1}$	0.00000	0.55559	0.65218	0.71428	0.75757	0.78947	0.81395	0.83333	0.84905
L	5.02088	2.54293	2.20997	2.04945	1.98300	1.97351	2.00111	2.05421	2.12560

Графічне відображення побудованих точок з табл. 3.1, які поєднані плавною кривою показано на наступному графіку (рис. 3.2).

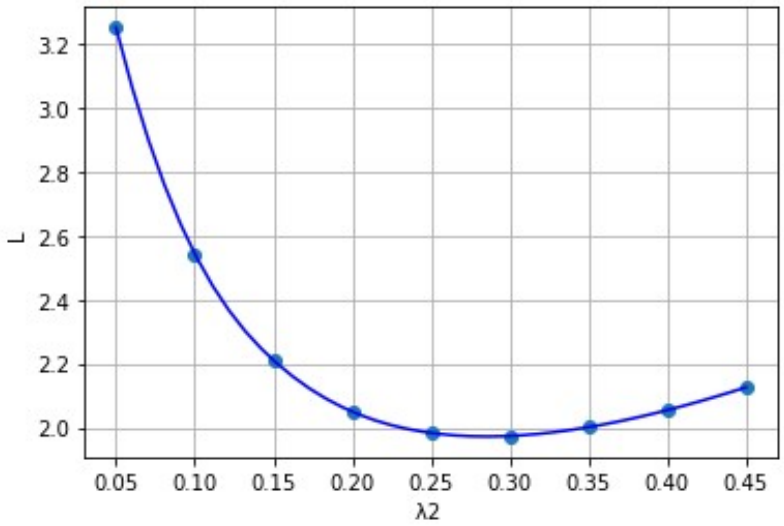


Рис. 3.2. Залежність повних витрат L від інтенсивності потоку λ_2

З графіку можна зробити висновок, що мінімальні витрати на повернення системи до стану H_1 зі стану H_2, а отже, забезпечення необхідної точності роботи системи при заданих параметрах, будуть при інтенсивності перерахунку даних $\lambda_2 = 0.3$, і витрати складатимуть $L = 1.97351$ умовних грошових одиниць за одиницю часу.

3.2. Розробка гібридного методу колаборативної фільтрації з підвищеною стійкістю до внутрішніх дестабілізуючих факторів

Перш ніж почати розробку методів колаборативної фільтрації, стійких до внутрішніх дестабілізуючих факторів, розглянемо способи оцінки якості, і головним чином точності, роботи рекомендаційних систем, які знадобляться в подальшому для тестування розроблених методів та порівняння їх з уже існуючими.

3.2.1. Дослідження способів оцінки та показників якості роботи рекомендаційних систем

Перевірку якості роботи рекомендаційної системи можна здійснити, використовуючи наступні дані:

1. За допомогою тестової вибірки, підготованої заздалегідь.

2. За допомогою моніторингу роботи рекомендаційної системи та збору даних у реальному часі.

У першому варіанті перевірки усі наявні дані слід розділити на навчаючу та тестову вибірку. Тестову вибірку не можна використовувати при навчанні рекомендаційної системи та при генерації списку рекомендацій. Тестова вибірка буде використана для порівняння даних з неї з даними зі списку рекомендацій. Для перевірки точності прогнозування вподобань при знаходженні однакових об'єктів у тестовій вибірці та у списку рекомендацій, їх прогнозовані та реально поставлені користувачем оцінки порівнюються. Таку перевірку легко організувати, однак її результати не досить інформативні, оскільки збігів у цих двох наборах даних буде досить мало.

Більш інформативним буде другий варіант, оскільки при моніторингу у реальному часі одразу буде видно як користувач реагує на кожну рекомендацію. Можна буде перевірити кожен з прогнозів в тій чи іншій формі – виявити факт обрання/необрання об'єктів зі списку рекомендацій користувачем, а при виставленні оцінки, порівняти її з прогнозованою. Але даний спосіб більш складно реалізувати, необхідно створити систему моніторингу веб-ресурсу, бажано мати доступ до нього на рівні адміністратора.

Було досліджено існуючі показники якості роботи рекомендаційних систем та способи їх оцінки [153-155].

Найголовнішим показником якості роботи рекомендаційної системи є *точність прогнозування вподобань (Prediction Accuracy)*. Для перевірки точності прогнозування вподобань користувачів порівнюють два вектори [76, 105]:

1) вектор $\hat{R} = (\hat{r}_1, \hat{r}_2, \dots, \hat{r}_n)$, що містить список прогнозованих оцінок користувача, впорядкований по спаданню за величиною оцінок;

2) вектор $R = (r_1, r_2, \dots, r_n)$, що містить справжні оцінки користувача, невідомі системі на етапі формування списку рекомендацій.

Якщо користувачі виставляють оцінки об’єктам, точність прогнозування можна визначити за допомогою середньоквадратичної помилки (1) або середньої абсолютної помилки (2):

$$RMSE = \sqrt{\frac{1}{|\tau|}\sum_{(u,i)\in\tau}(\hat{r}_{ui} - r_{ui})^2}\,, \tag{3.17}$$

$$MAE = \frac{1}{|\tau|}\sum_{(u,i)\in\tau}|\hat{r}_{ui} - r_{ui}|\,, \tag{3.18}$$

де $\hat{r}_{ui}$ – прогнозовані рейтинги для тестового набору даних τ пар користувач-об’єкт (u, i), r_{ui} – справжні рейтинги.

Щоб здійснити вимірювання точності система генерує прогнозовані рейтинги $\hat{r}_{ui}$ для тестового набору даних τ пар користувач-об’єкт (u, i), для яких відомі справжні рейтинги r_{ui}. Реальні та прогнозовані рейтинги порівнюються.

Існує також багато інших показників якості роботи рекомендаційних систем [12, 21, 39, 76, 99, 105], основні з них наведені нижче.

Прогнозування використання. Оцінювати точність роботи рекомендаційної системи можна завдяки реакції користувача на об’єкти наведені у списку рекомендацій. В такому разі інформація для оцінки роботи рекомендаційної системи з’являється в процесі її використання.

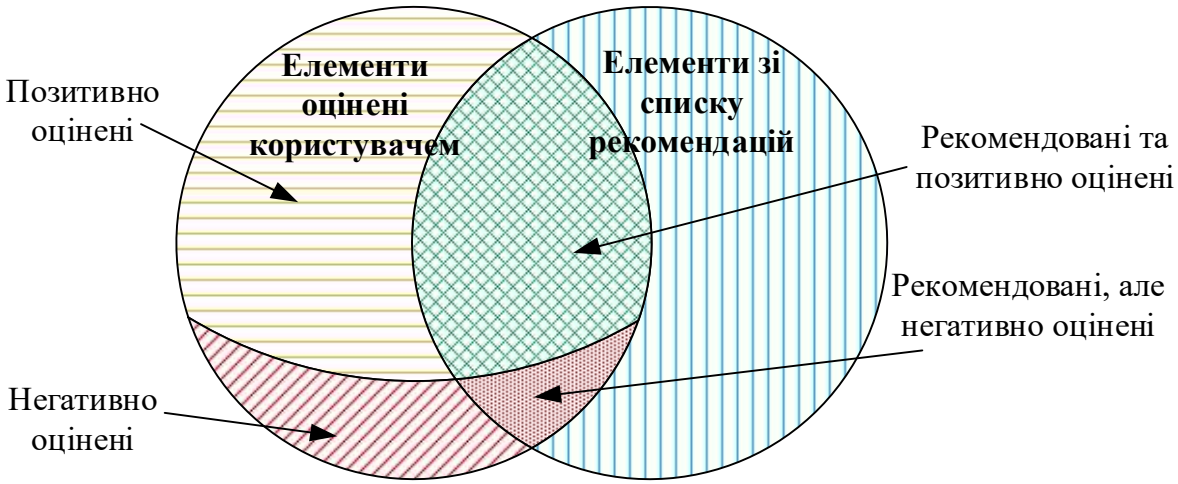


Рис. 3.3. Можливі варіанти реакції користувача на об’єкти у списку рекомендацій

Наведемо можливі результати прогнозу роботи рекомендаційної системи (табл. 3.2).

Таблиця 3.2. Класифікація можливих результатів рекомендації

	Рекомендували	**Не рекомендували**
Позитивно оцінено	True-Positive (*tp*)	False-Negative (*fn*)
Негативно оцінено	False-Positive (*fp*)	True-Negative (*tn*)

Як видно з таблиці, можливі чотири результати роботи рекомендаційної системи:

- *tp* – результати, в яких позитивний прогноз виявився вірним;
- *tn* – результати, в яких негативний прогноз виявився вірним;
- *fp* – результати, в яких позитивний прогноз виявився помилковим;
- *fn* – результати, в яких негативний прогноз виявився помилковим.

Можна підрахувати кількість подій, що відповідають кожній з комірок, та на основі одержаних значень оцінити точність роботи рекомендаційної системи, наприклад, на основі наступних показників [76]:

$$Precision = \frac{tp}{tp + fp}, \tag{3.19}$$

$$Recall\ (True\ Positive\ Rate) = \frac{tp}{tp + fn}, \tag{3.20}$$

$$False\ Positive\ Rate = \frac{fp}{fp + tn}, \tag{3.21}$$

Міри ранжирування. Важливою задачею є ранжування об’єктів у списку рекомендацій. За допомогою даної міри можна визначити наскільки вірно рекомендаційна система впорядкувала елементи у списку рекомендацій.

Міри ранжирування з використанням заздалегідь відомих рейтингів. Як міру ранжування можна використати Normalized Distance-based Performance Measure (NDPM) [99]. Якщо у нас є справжні рейтинги об’єктів r_{ui}, одержані в результаті дій користувача, та рейтинги, згенеровані рекомендаційною системою $\hat{r}_{ui}$ для n_u об’єктів i користувачу u, то можна одержати:

$$C^{+} = \sum_{ij} \mathrm{sgn}(r_{ui} - r_{uj}) \, \mathrm{sgn}(\hat{r}_{ui} - \hat{r}_{uj}), \tag{3.22}$$

$$C^{-} = \sum_{ij} \mathrm{sgn}(r_{ui} - r_{uj}) \, \mathrm{sgn}(\hat{r}_{uj} - \hat{r}_{ui}), \tag{3.23}$$

$$C^{u} = \sum_{ij} \mathrm{sgn}^2 (r_{ui} - r_{uj}), \tag{3.24}$$

$$C^{s} = \sum_{ij} \mathrm{sgn}^2 (\hat{r}_{ui} - \hat{r}_{uj}), \tag{3.25}$$

$$C^{u0} = C^{u} - (C^{+} + C^{-}), \tag{3.26}$$

де суми перевищують $\frac{1}{2} n_u (n_u - 1)$ пар об’єктів. Таким чином, C^u – це кількість пар об’єктів, для яких справжні рейтинги встановлюють впорядкування один відносно одного, тоді як C^+ та C^- – це кількість пар об’єктів, для яких рекомендаційна система встановила правильний порядок і неправильний порядок впрорядкування відповідно. C^{u0} – це кількість пар, для яких немає впорядкування в справжньому рейтингу, але рекомендаційна система встановлює для них деяке впорядкування. NDPM одержується наступним чином:

$$NDPM = \frac{C^{-} + 0.5 C^{u0}}{C^{u}}, \tag{3.27}$$

Таким чином, міра NDPM дає найкращу оцінку 0 – для систем, які правильно передбачають кожне впорядкування пар об’єктів. А найгірша оцінка 1 – призначена системам, що суперечать усім вірним впорядкуванням пар об’єктів.

Міри ранжирування на основі корисності. Популярною альтернативою попереднього методу є припущення, що корисність переліку рекомендацій є сумарною, що визначається сумою корисності окремих рекомендацій. Корисність кожної рекомендації – це корисність рекомендованого об’єкту помножена на коефіцієнт зменшення, який залежить від позиції об’єкту в переліку рекомендацій. Одним з прикладів такої корисності є ймовірність того, що користувач буде дотримуватися рекомендації в даній позиції у

списку. Зазвичай передбачається, що користувачі проглядають списки рекомендацій від початку до кінця, з урахуванням того, що переваги рекомендацій значно знижуються до кінця списку.

В багатьох додатках список рекомендацій далеко не самий основний спосіб пошуку об'єктів, він містить невелику кількість елементів, а в крайніх випадках може містити тільки один елемент. В таких системах користувач зазвичай переглядає тільки невелику частину списку рекомендацій – тільки декілька пунктів на початку. В таких випадках цінність рекомендацій знижується дуже швидко відносно позицій елементів у списку. Для таких додатків можна використати *метрику R-Score* [12, 76].

Метрика R-Score передбачає, що корисність рекомендацій знижується експоненціально вниз по впорядкованому списку рекомендацій, щоб отримати наступний бал для кожного користувача *u*:

$$R_u = \sum_j \frac{\max(r_{ui_j} - d, 0)}{2^{\frac{j-1}{\alpha-1}}}, \tag{3.28}$$

де i_j – це об'єкт в *j*-й позиції, r_{ui} – рейтинг (оцінка) користувача *i*, *d* – значення максимальної негативної оцінки, а α – період напіврозпаду, який контролює експоненціальне зниження значення позицій у ранжовому списку. У випадку задач прогнозування рейтингів r_{ui} – це рейтинг (оцінка), наданий користувачем для кожного елемента, а *d* – це негативна оцінка (наприклад, три зірки з п'яти), і алгоритм отримує кредити лише за позиції рейтингу вище негативних оцінок, тобто, вище значення *d* (наприклад, 4 або 5 зірок). У задачі передбачення використання r_{ui} звичайно дорівнює 1, якщо *u* вибирає *i,* та 0 в іншому випадку, якщо $d = 0$.

Отримані для кожного користувача результати агрегуються за допомогою:

$$R = 100 \frac{\sum_u R_u}{\sum_u R_u^*}, \tag{3.29}$$

де R_u^* – оцінка найкращого рейтингу для користувача *u*.

Крім точності формування та ранжування списку рекомендацій до рекомендаційної системи може висуватися багато інших вимог, наприклад: покриття, різноманітність, новизна, приватність, робастність до атак, адаптивність, масштабованість, пропускна здатність тощо. Покращення даних показників рекомендаційної системи може знизити її точність прогнозування оцінок, але підвищити загальну якість роботи. Розглянемо основні з них.

Важливим показником якості роботи рекомендаційної системи є *покриття (Coverage)* [76], що характеризує відсоток охоплення елементів системи у процесі формування рекомендацій, існує декілька його видів:

– *Покриття каталогу* (покриття простору об'єктів) – може визначатися як відсоток усіх елементів, які можуть бути рекомендовані, даний показник дозволяє виявити об'єкти, які нікому не рекомендуються.

– *Покриття простору користувачів* – може характеризуватися часткою користувачів або взаємодій користувачів, для яких система може рекомендувати об'єкти (у багатьох системах рекомендації можуть не надаватися для користувачів, про яких зібрано мало даних, через низьку впевненість у точності прогнозів), якщо рекомендації слід надавати всім користувачам у системі, то необхідно йти на компроміс між покриттям та точністю.

– *Різноманітність збуту* – міра неоднорідності вибору різних об'єктів користувачами зі списку рекомендацій, для її визначення можна використовувати різні індекси, зокрема, коефіцієнт Джині (3.30) або ентропію Шеннона (3.31):

$$G = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j), \tag{3.30}$$

де $i_1, \ldots, i_n$ – це список об'єктів, впорядкованих за збільшенням частоти їх вибору користувачем або частоти їх появи у списку рекомендацій $p(i)$. Індекс $G = 0$, коли всі елементи вибираються однаково часто, а $G = 1$ – коли завжди вибирається один елемент.

$$H = -\sum_{i=1}^{n} p(i) \log p(i), \tag{3.31}$$

де ентропія дорівнює 0, коли завжди обирається або рекомендується один об'єкт, та дорівнює $\log n$, коли n об'єктів вибирається чи рекомендується однаково часто.

Точність прогнозування вподобань і покриття є настільки важливими показниками якості роботи рекомендаційних систем, що їх слід віднести до основних.

Розглянемо й інші показники, за допомогою яких можна оцінити якість роботи рекомендаційної системи за різними критеріями.

Усі сучасні рекомендаційні системи схильні до проблеми бульбашки фільтрів, що виникає, коли алгоритм формування списку рекомендацій підбирає інформацію, яку користувач хотів би бачити, і, в результаті, користувачі відділяються від інформації, яка їх не цікавить, тому що зовсім невідома їм або не подобається, фактично ізолюючи їх у власних «бульбашках». Для вирішення даної проблеми до рекомендаційних систем висуваються наступні вимоги – список рекомендацій повинен володіти наступними властивостями [21, 39, 43, 76, 102]:

– *Різноманітність рекомендацій (Diversity).* Це міра схожості елементів списку. Елементи у рекомендацій системі не повинні бути майже однаковими, вони повинні містити різнотипні об'єкти (наприклад, фільми різних жанрів, а не тільки одного жанру, чи однієї трилогії). Для визначення схожості елементів можна застосовувати різні коефіцієнти подоби (коефіцієнт кореляції Пірсона, косинусну міру, евклідову відстань, відстать Хеммінга тощо), за допомогою яких попарно порівнювати елементи списку, після визначення рівня схожості між окремими елементами можна буде оцінити різноманітність списку рекомендацій вцілому.

– *Новизна рекомендацій (Novelty).* Нові об'єкти у системі можуть ще не мати оцінок і не бути популярними, але вони можуть бути цікавими користувачам через свою новизну. В той же час нові об'єкти необхідно

комусь рекомендувати, щоб вони не залишилися без уваги. Якщо користувачу рекомендувати лише популярні об'єкти, скоріше за все він їх і так знає та обере без рекомендаційної системи, такі рекомендації не будуть містити для нього нової інформації. По суті, новизна – це характеристика елемента у списку рекомендацій протилежна його популярності, і в найпростішому випадку може визначатися за формулою [21]:

$$novelty(i) = -\log_2 p(i), \tag{3.32}$$

де $p(i)$ – ймовірність того, що об'єкт i потрапить у список рекомендацій (буде обрано).

Неочікуваність рекомендацій (Serendipity). Неочікуваність представляє собою деякий сюрприз у списку рекомендацій, несхожість на історію дій користувача. Не існує консенсусу у визначенні неочікуваності, однак більшість авторів вказує, що елемент, який має властивість неочікуваність, повинен бути важливим, новим та непрогнозованим для користувача [39]. Важливість для користувача виражається в його реакції на даний елемент після рекомендацій, новизна виражається в тому, наскільки користувач знайомий з даним елементом. Елемент може бути незнайомим для користувача, якщо: 1) користувач ніколи не чув про даний елемент, 2) користувач чув про даний елемент, але ніколи не використовував, 3) користувач використовував даний елемент, але забув про це. Елемент може бути непрогнозованим для користувача, якщо: 1) користувач не очікує, що цей елемент буде для нього актуальним, 2) користувач не очікує, що цей елемент буде рекомендований йому, 3) користувач не знайшов би цього елементу самостійно, 4) цей елемент значно відрізняється від елементів, які як правило, обирає користувач, 5) користувач не очікує даного елементу у списку рекомендацій, оскільки він переглядав інші види елементів.

Збільшення різноманітності, неочікуваності та новизни рекомендацій може знизити точність прогнозування та точність ранжування, в той же час може підвищитися покриття каталогу об'єктів, різноманітність збуту та частково вирішитися проблема бульбашки фільтрів.

На даний час не існує загальноприйнятих мір та методів оцінки різноманітності, новизни та неочікуваності списків рекомендацій.

Серед показників якості рекомендаційних систем, що є важливими з погляду інформаційної безпеки можна виділити наступні [22, 32, 47, 61, 76]:

1. *Приватність користувача (Privacy).* Міра того наскільки приватна інформація користувачів, яку вони надають рекомендаційній системі для одержання списків пропозицій, захищена від потрапляння її до третіх сторін.

2. *Ризик для користувача (Risk).* В деяких випадках рекомендації можуть бути пов'язані з ризиком. Наприклад, якщо об'єктами в рекомендаційній системі є акції, кредити, депозити, ліки, медичні послуги, політичні акції тощо. В таких випадках може бути необхідним врахування не тільки вподобань користувача при формуванні рекомендацій, а й інших факторів, врахування яких здатне мінімізувати ризик для користувача, що буде переглядати та обирати рекомендації.

3. *Робастність системи до атак (Robustness).* Здатність системи надавати адекватні рекомендації при появі некоректної інформації. Некоректна інформація може виникати при іформаційних атаках бот-мереж на систему для (збільшення або зменшення) рейтингів певних об'єктів з метою зміни частоти потрапляння їх у списки рекомендацій.

Також, в залежності від потреб певного веб-ресурсу чи додатку, можна виділити й інші показники якості роботи рекомендаційних систем, наприклад [63, 76, 209]:

– *Впевненість (Confidence)*. Рекомендаційна система може додавати до своїх рекомендацій процент впевненості у них. Так, наприклад, система може вказати для першої рекомендації прогнозовану оцінку користувача 5 балів з впевненістю на 95%, а для другої рекомендації – 5 балів з впевненістю на 89%. Користувач може враховувати даний параметр та обирати в першу чергу об'єкти з більшим значенням впевненості системи.

– *Довіра (Trust)*. Дана властивість характеризує наскільки користувач може довіряти даній рекомендаційній системі. Якщо рекомендаційна система

запропонувала користувачу лише невідомі йому об’єкти, то користувач може засумніватися у правильності роботи системи, а якщо серед рекомендацій є певна кількість об’єктів, про які він знає і які йому подобаються, то рівень довіри до системи буде вищим. Тобто, деяка кількість об’єктів, про які відомо користувачу, може збільшити довіру до системи. Ще одним варіантом збільшення довіри до системи є пояснення до рекомендації. Наприклад, коли елемент зі списку рекомендацій подається у вигляді «Якщо Вам подобається X, то спробуйте Y». Також довіра користувача зростає, якщо рекомендації були йому корисні і відповідали його вподобанням, та знижується в протилежному випадку.

– *Адаптивність (Adaptivity)*. Реальна система рекомендацій може працювати у ситуації, коли колекції об’єктів, і/або інтереси користувачів швидко змінюються. Прикладом таких систем можуть бути рекомендаційні системи новинних сайтів, коли об’єкти (новини) цікаві тільки на певному проміжку часу. В таких системах з одного боку цікавість до певного об’єкту існує тільки деякий час, з іншого боку якісь старі об’єкти можуть знову ставати дуже цікавими, якщо нові об’єкти у системі поновлюють інтерес до старих. В таких системах слід вибирати швидкі алгоритми, навіть якщо доводиться в деякій мірі жертвувати точністю. Інший тип адаптивності – це те, з якою швидкістю система адаптується до вподобань користувача або до змін в його профілі. Якщо дії користувача на сайті занадто повільно змінюють рекомендації, він може відмовитися оцінювати об’єкти, не отримавши зворотного зв’язку. Адаптивність алгоритмів можна оцінювати шляхом визначення різниці між списками рекомендацій до та після додавання нової інформації з профіля користувача, наприклад, за допомогою міри ентропії Шеннона.

– *Масштабованість (Scalability)*. Зі збільшенням об’ємів даних багато алгоритмів працюють значно повільніше або вимагають додаткових ресурсів, таких як обчислювальні потужності або пам’ять. Тому важливо враховувати просторову та часову складність алгоритмів. В реальних системах може

виникнути необхідність згодитися на меншу точність рекомендацій для одержання більш масштабованої системи.

– *Пропускна здатність (Throughput)*. Це кількість рекомендацій, які система може створити та надати користувачам в одиницю часу.

– *Корисність (Utility)*. Значення, що характеризує, яку вигоду від рекомендацій отримує власник системи та/або користувач.

При побудові рекомендаційної системи для контент-орієнтованого веб-сайту чи додатку досить логічним кроком буде визначення списку показників якості, яким повинна задовольняти розроблювана рекомендаційна система, та вибір/розробка алгоритмів і методів її побудови на основі визначення у процесі їх тестування, наскільки вони задовольняють висунутим критеріям.

На основі проведеного дослідження показників якості роботи рекомендаційних систем було обрано найбільш важливі з них для оцінювання точності роботи системи в умовах дії дестабілізуючих факторів. А саме, для тестування розроблюваних методів роботи рекомендаційних систем вирішено використовувати наступні показники: Precision, Recall, RMSE, покриття каталогу та покриття простору користувачів.

3.2.2. Розробка методу колаборативної фільтрації з врахуванням показників активності користувачів для підвищення стійкості рекомендаційної системи в умовах холодного старту

Однією з важливих проблем колаборативної фільтрації є те, що не для всіх користувачів вдається створити списки рекомендацій через проблеми холодного старту [34, 76, 157], постійного холодного старту [9] або недостатньої кількості користувачів, схожих на певного користувача. Робота сучасних рекомендаційних систем на основі колаборативної фільтрації з застосуванням моделі сусідства заснована на визначенні коефіцієнтів подоби між користувачами та/або об’єктами системи [76, 78, 84, 171, 209]. Подоба користувачів між собою визначається на основі їх дій з одними й тими

самими об’єктами, якщо ці дії можна порівняти, наприклад, порівнюються оцінки, які вони виставляли однаковим об’єктам. Якщо у користувачів немає однакових дій для порівняння, то визначити їх коефіцієнти подоби стандартними методами фільтрації неможливо. А якщо для певного користувача не вдалося визначити схожих на нього користувачів, то і не можна сформувати для нього список рекомендацій на основі колаборативної фільтрації. В таких випадках застосовують додаткові дані, наприклад, врахування контекстної інформації або інформації про соціальні зв’язки, або перехід до контентної фільтрації [76, 171, 186, 203].

Отже, дану проблему можна частково вирішувати за допомогою гібриду колаборативної фільтрації з іншими методами, зокрема, контентною, контекстною або соціальною фільтрацією. Оскільки не завжди можна створити такий гібрид, так як потрібно зібрати досить багато інформації про об’єкти або користувачів, а також більше часу витратити на обчислення, було вирішено дослідити альтернативні способи рішення проблеми відсутності рекомендацій для деяких користувачів. Зокрема, спробувати враховувати при генерації рекомендацій показники активності користувачів системи, та оцінити наскільки врахування інформації про активність користувачів покращить якість формування списків рекомендацій.

Було запропоновано для формування списку рекомендацій використовувати дані про показники активності користувачів. Тобто, у список рекомендацій користувачів, для яких не вдалося створити пропозиції іншими методами, будуть потрапляти найпопулярніші об’єкти серед найактивніших користувачів. Такий підхід пропонується назвати експертно-орієнтованим, оскільки кожний користувач рекомендаційної системи розглядається як експерт з контенту цієї системи та має вагу (коефіцієнт експерта), що залежить від його активності. Чим більша вага експерта, тим більший внесок у формування списків рекомендацій іншим користувачам вносять дані про його вподобання. Пропонується визначати вагу експерта на основі рівня його активності, хоча в майбутньому можлива розробка й інших

методів визначення ваги експерта (наприклад, на основі його репутації у системі, кількості користувачів подібних до нього за інтересами тощо).

На основі даного підходу був розроблений метод експертно-орієнтованої фільтрації з застосуванням показників рівня активності користувачів, а також гібрид даного методу з відомим методом колаборативної фільтрації [55].

Також було розроблено програмне забезпечення та проведена серія експериментів для перевірки ефективності розробленого методу та розробленого гібриду даного методу з відомим методом колаборативної фільтрації, а також порівняння їх зі стандартним методом колаборативної фільтрації [55, 161].

У розробленому методі пропонується вважати найактивніших користувачів системи, які виставили найбільшу кількість оцінок об’єктам системи, експертами у контенті системи. Також пропонується для кожного користувача системи обчислювати значення коефіцієнту експерта та використовувати такі коефіцієнти в алгоритмі формування списків рекомендацій.

Було вирішено розраховувати коефіцієнт експерта для кожного користувача наступним чином:

$$k_{\text{exp}}(u) = \frac{m_u}{\max(m_i)}, \tag{3.33}$$

де m_u – кількість об’єктів, які оцінив користувач u; $\max(m_i)$ – кількість об’єктів, які оцінив користувач, що оцінив найбільше об’єктів системи з-поміж усіх користувачів.

Чим більше користувач оцінив об’єктів, тим більший коефіцієнт експерта у нього буде.

Значення даного коефіцієнту може змінюватися у діапазоні від 0 до 1. Він буде дорівнювати 0, якщо користувач не оцінив жодного об’єкту, та дорівнювати 1, якщо користувач оцінив найбільшу кількість об’єктів порівняно з іншими.

Етапи розробленого методу колаборативної фільтрації з врахуванням показників активності користувачів:

Етап 1. Обчислюємо коефіцієнти експертів (3.33) для всіх користувачів системи. Для формування списоку рекомендацій окремому користувачу u_i виконуємо наступні етапи методу.

Етап 2. Беремо усіх користувачів, крім користувача u_i.

Етап 3. Вибираємо всі об'єкти, яким поставили оцінки дані користувачі та розраховуємо для кожного об'єкту коефіцієнт його цікавості для користувача u_i:

$$k_{\text{int}}(o, u_i) = \sum_{j \geq 1, j \neq i}^{n} r_j \cdot k_{\text{exp}}(u_j), \tag{3.34}$$

де o – об'єкт, для якого розраховуємо коефіцієнт цікавості для користувача u_i; n – кількість користувачів у системі; k_{exp} – коефіцієнт експерта для користувача u_j; r_j – оцінка, яку поставив користувач u_j об'єкту o.

Етап 4. Формуємо список всіх вибраних об'єктів, за необхідності, відкидаємо зі списку об'єкти, які вже оцінював користувач u_i.

Етап 5. Сортуємо список всіх об'єктів за спаданням коефіцієнту цікавості для користувача u_i.

Етап 6. За необхідності, вибираємо TopN об'єктів з одержаного списку.

Необхідність виконання пунктів 4 та 6 залежить від того, для якого додатку чи веб-ресурсу створюється рекомендаційна система. Можливі випадки, коли користувачу можна рекомендувати об'єкти, які він уже переглядав раніше (наприклад, у випадку, Інтернет-радіо). Можливі випадки, коли треба показувати весь список рекомендацій, а не тільки TopN (наприклад, у випадку пошукових систем).

Також в рамках даної роботи було розроблено гібрид запропонованого методу з відомим методом колаборативної фільтрації на основі сусідства.

Оскільки всі дані у системі зручно записувати у вигляді графу [161], де користувачі є вершинами графу, а коефіцієнти подоби та коефіцієнти експертів записуються як значення властивостей відповідних ребер графу,

для того, щоб пояснити принцип роботи розробленого гібриду, введемо наступні поняття:

– двонаправлені зв’язки типу «Схожість» між користувачами (рис. 3.4);

– орієнтовані зв’язки типу «Експерт» між користувачами (рис. 3.5).

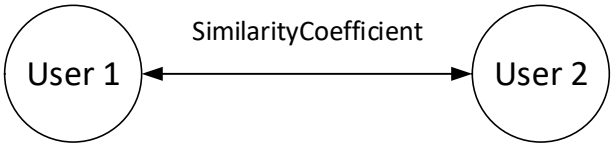


Рис. 3.4. Зв’язок типу «Схожість» між користувачами

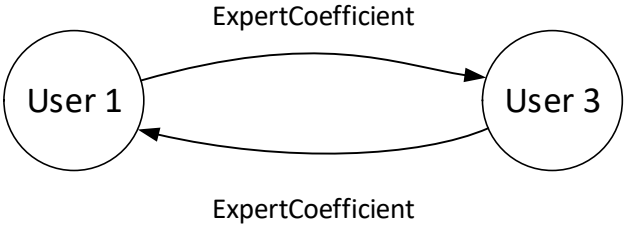


Рис. 3.5. Зв’язки типу «Експерт» між користувачами

Розроблений гібридний метод працює наступним чином:

Етап 1. Обчислюються коефіцієнти подоби між усіма користувачами системи (рис. 3.4), наприклад, на основі кореляції Пірсона (1.6) або косинусної подоби (1.7).

Етап 2. Створюються зв’язки типу «Схожість» між користувачами, для яких знайдено коефіцієнти подоби.

Етап 3. Для всіх користувачів визначаються коефіцієнти експертів (3.33).

Етап 4. Для тих користувачів, у яких немає спільних дій с іншими користувачами (рис. 3.5-3.6, User3), а отже, неможливо обчислити коефіцієнти подоби з ними та створити зв’язки типу «Схожість», застосовується розроблений метод експертно-орієнтованої фільтрації, заснованої на врахуванні показників активності користувачів. В такому випадку від усіх користувачів до даного користувача, а також від даного користувача до усіх інших користувачів, будуть направлені зв’язки типу

«Експерт» (рис. 3.6, User3). Для формування списоку рекомендацій окремому користувачу u_i виконуємо наступні етапи методу.

Етап 5. Якщо у користувача u_i є зв'язки типу «Схожість» прогнозуємо для нього оцінки відомим методом колаборативної фільтрації на основі моделі сусідства. Якщо у користувача є зв'язки типу «Експерт», вибираємо всі об'єкти, яким поставили оцінки інші користувачі та розраховуємо для кожного об'єкту коефіцієнт його цікавості (3.34) для користувача u_i.

Етап 6. Формуємо список всіх об'єктів для яких обчислено прогнозовану оцінку або коефіцієнт цікавості, за необхідності, відкидаємо зі списку об'єкти, які вже оцінював користувач u_i.

Етап 7. Сортуємо список всіх об'єктів за спаданням прогнозованої оцінки або коефіцієнту цікавості для користувача u_i.

Етап 8. За необхідності, вибираємо TopN об'єктів з одержаного списку.

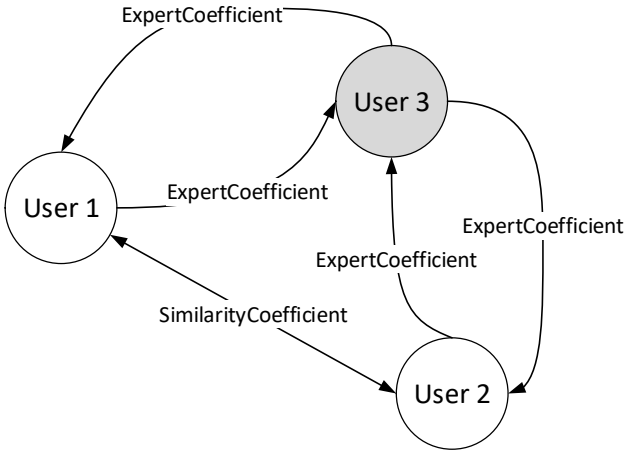


Рис. 3.6. Приклад побудови зв'язків типу «Схожість» та «Експерт» у гібридному методі

Таким чином зникає проблема з користувачами, у яких ні з ким немає зв'язків типу «Схожість», рекомендації для них будуються на основі вподобань найбільш активних користувачів у системі. Також зникає проблема з об'єктами, які оцінили тільки ті користувачі, у яких немає ні з ким зв'язків подоби, а отже невідомо кому їх можна рекомендувати, зв'язки

типу «Експерт» від цих користувачів будуть направлені до всіх інших користувачів, і чим більше об’єктів вони оцінили, тим більший внесок у список рекомендацій іншим користувачам зроблять поставлені ними оцінки. Можна сказати, що «ні на кого не схожі користувачі», є експертами з «рідкісного контенту».

Для проведення експериментів було побудовано рекомендаційну систему з використанням мови програмування Python та графової бази даних Neo4j [66], так як СУБД Neo4j надає широкі можливості для роботи з даними рекомендаційної системи [66, 144, 152, 161, 196]. В розробленій рекомендаційній системі було реалізовано колаборативну фільтрацію, в якій коефіцієнти подоби між користувачами визначалися на основі косинусної подоби (1.7).

Розроблена рекомендаційна система тестувалася на відкритому наборі даних MovieLens Datasets науково-дослідної лабораторії, створеному у відділі комп'ютерних наук та інженерії в Університеті Міннесоти [30]. Дана рекомендаційна система розроблена для веб-сайту трейлерів фільмів. Користувачі, які ставлять оцінки та хеш-теги фільмам, одержують для себе списки рекомендацій фільмів для майбутніх переглядів.

Під час кожного експерименту з набору даних MovieLens Datasets обиралося випадковим чином N_u користувачів. Дані про оцінки, які вони ставили фільмам було розділено на основі часових міток на дві частини для розрахунків рекомендацій («поточні дані») та для тестування системи («майбутні дані»).

З MovieLens Datasets вибиралися наступні записи для кожного з N_u користувачів:

1. Ідентифакатор користувача N_x (ціле число);

2. Ідентифікатор фільму M_y (ціле число);

3. Оцінка, яку користувач N_x поставив фільму M_y;

4. Часова мітка, яка вказує час, коли користувач N_x поставив фільму M_y оцінку.

У кожного користувача в наборі MovieLens Datasets є мінімум один такий запис, загалом у користувача може бути декілька десятків або сотень таких записів.

Обрана інформація з MovieLens Datasets вибиралася та записувалася у графову базу даних neo4j, у наступному форматі (рис. 3.7):

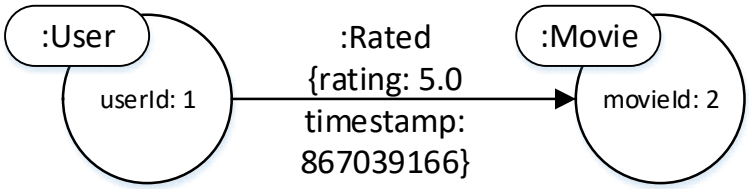


Рис. 3.7. Приклад запису у базі даних розробленої системи

В результаті занесення таких записів у графову базу даних рекомендаційної системи, вона буде мати приблизно такий вигляд (рис. 3.8):

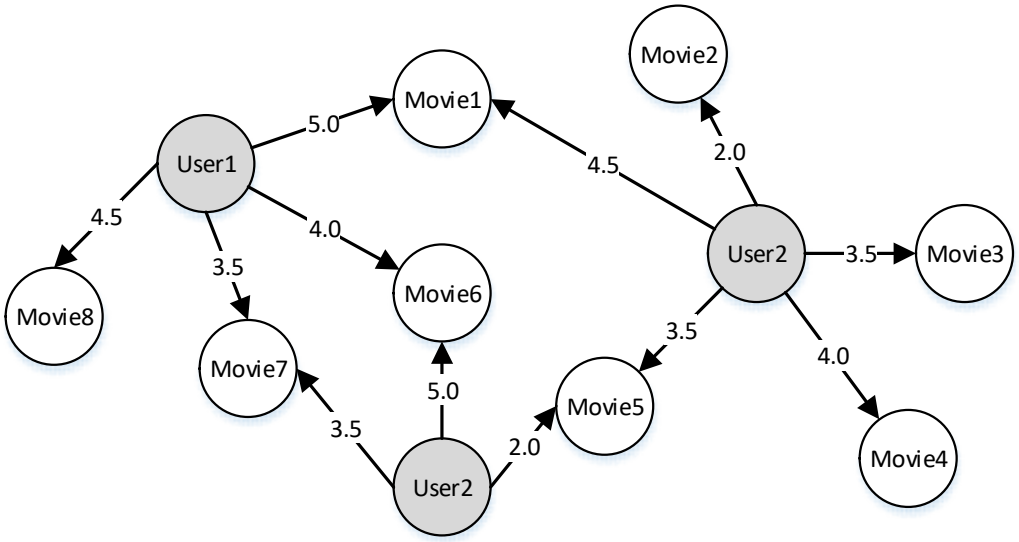


Рис. 3.8. Загальний вигляд вмісту графової бази даних у розробленій рекомендаційній системі

Для кожного набору даних система запускалася у трьох режимах:

– відомий метод колаборативної фільтрації на основі моделі сусідства;

– розроблений експертно-орієнтований метод колаборативної фільтрації на основі даних про показники рівня активності користувачів;

– гібрид перших двох методів.

В наборі даних MovieLens Datasets оцінки користувачів могли приймати наступні значення 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0. Було прийнято рішення розділити їх на позитивні (від 3.5 до 5.0) та негативні (від 0.5 до 3.0) оцінки. Прогнози вподобань користувачів також поділені на позитивні, коли прогнозується позитивна оцінка для пари користувач-об'єкт, та негативні, коли прогнозується негативна оцінка.

Було проведено серію експериментів для користувачів, вибраних випадковим чином з MovieLens Datasets. Для кожного експерименту вибирався інший набір користувачів. Дані експериментів наведені у таблиці 3.3. У ній використовувалися наступні скорочення:

– **ВМ** – відомий метод колаборативної фільтрації на основі моделі сусідства;

– **АҐ** – розроблений метод, що використовує інформацію про активність користувачів;

– **Г** – гібрид цих двох методів.

Таблиця 3.3. Результати тестування розробленого методу та гібриду

№ експерименту	Загальна кількість користувачів	Кількість користувачів без рекомендацій			Точність (Precision)			Повнота (Recall)			Покриття простору користувачів, %			Покриття каталогу, %		
		ВМ	АК	Г	ВМ	АК	Г	ВМ	АК	Г	ВМ	АК	Г	ВМ	АК	Г
1	30	13	0	0	0.8275	0.7281	0.8168	0.8571	0.6881	0.6792	56.66	100	100	33.79	72.81	72.81
2	30	16	0	0	0.9230	0.8155	0.8173	0.7741	0.7966	0.7985	46.66	100	100	22.19	63.08	63.08
3	30	10	0	0	0.5818	0.6745	0.6745	0.5818	0.7544	0.7722	66.66	100	100	27.29	67.45	67.45
4	50	18	0	0	0.7213	0.8308	0.8458	0.8627	0.7729	0.7723	64.00	100	100	33.14	85.57	85.57
5	50	21	0	0	0.4166	0.6655	0.6627	0.8333	0.7131	0.8415	57.99	100	100	28.88	71.31	71.31
6	50	22	0	0	0.6571	0.7424	0.7413	0.6969	0.7222	0.7199	56.00	100	100	21.85	52.65	52.65
7	100	43	0	0	0.7258	0.6524	0.6510	0.7758	0.7292	0.7292	56.99	100	100	26.74	68.43	68.43
8	100	54	0	0	0.7343	0.8172	0.8175	0.7121	0.6853	0.6860	46.00	100	100	20.21	47.80	47.80
9	100	46	0	0	0.8333	0.7552	0.7548	0.6250	0.7771	0.7727	54.00	100	100	37.77	77.71	77.71
Середні значення показників:					**0.7134**	**0.7424**	**0.7535**	**0.7465**	**0.7376**	**0.7523**	**56.10**	**100**	**100**	**27.98**	**67.42**	**67.42**

Як показали результати експерименту, розроблений метод на відміну від методу колаборативної фільтрації дозволяє створювати рекомендації для всіх

користувачів системи. Точність роботи та повнота розробленого методу можуть бути як гіршими, так і кращими, ніж у відомого методу колаборативної фільтрації, що повністю залежить від особливостей набору вхідних даних. Але в середньому розроблений метод та розроблений гібрид мають більшу точність та практично таку ж повноту порівняно з відомим методом колаборативної фільтрації. Покриття простору користувачів в розробленому методі та в розробленому гібриді завжди 100%, а от у відомому методі колаборативної фільтрації в середньому цей показник рівний 56.1%. Також розроблений метод та розроблений гібрид завжди дають кращі показники покриття каталогу об'єктів, в середньому на 39.44% більше, ніж відомий метод колаборативної фільтрації.

Отже, без суттєвих коливань точності розроблений метод та розроблений гібрид збільшують покриття каталогу товарів в 2.5 рази, а покриття простору користувачів роблять 100%, вирішуючи проблему з користувачами, для яких у відомому методі колаборативної фільтрації не було рекомендацій.

Важливо відмітити, що при наявності бот-мереж даний метод може бути не стійким до їх інформаційних атак, так як активність профілів ботів може бути досить високою, що даватиме їм великі значення коефіцієнтів експерта. Але в той же час ботів з високими показниками активності легко виявити. Тому спочатку треба перевірити систему на наявність ботів, провести їх ідентифікацію та нейтралізацію і тільки після цього використовувати даний метод.

3.2.3. Розробка методу колаборативної фільтрації з використанням продукційних правил для підвищення стійкості рекомендаційної системи в умовах малої кількості вхідних даних

Як було зазначено вище, в основі базових методів колаборативної фільтрації лежить визначення коефіцієнтів подоби між користувачами або

об’єктами на основі оцінок, які користувачі ставили об’єктам [76, 78, 84, 171, 209], щоб рекомендувати кожному користувачу контент, який зацікавив максимально схожих на нього користувачів та водночас не переглядався ним у минулому. Для визначення коефіцієнту подоби між двома довільними користувачами у них повинні бути наявні спільні дії для порівняння, наприклад, вони повинні поставити оцінки одним і тим же об’єктам. Порівнюючи виставлені оцінки однаковим об’єктам можна зрозуміти однакові у користувачів вподобання або різні.

На основі коефіцієнтів подоби прогнозуються можливі оцінки користувачів об’єктам, які вони ще не переглядали, та формуються рекомендації. У найпростішому варіанті оцінки прогнозуються за наступними формулами:

$$r_{u,i} = k \sum_{u' \in U} sim(u,u') r_{u',i} \,, \tag{3.35}$$

$$k = \frac{1}{\sum_{u' \in U} \left| sim(u,u') \right|}, \tag{3.36}$$

де *sim* – обраний коефіцієнт подоби користувачів; *U* – множина користувачів; *u* – користувач, для якого прогнозується вподобання; *u’* – усі інші користувачі системи; *i* – об’єкт, для якого прогнозується вподобання користувача *u*; *r* – оцінка, яку поставив користувач *u’* об’єкту *i*; *k* – нормувальний коефіцієнт.

Як правило, коефіцієнти подоби вдається визначити далеко не для кожної пари користувачів. Якщо два користувача не оцінювали однакових об’єктів, то визначити коефіцієнт подоби між ними таким способом не можна. Якщо один або обидва користувачі оцінювали невелику кількість об’єктів, то точність визначеного коефіцієнту подоби буде низькою.

Було вирішено спробувати визначити коефіцієнти подоби між користувачами, які не ставили оцінок однаковим об’єктам. Для цього було запропоновано асоціативні [54, 150, 160] та продукційні правила [64, 149] засновані на принципі транзитивності.

Асоціативні правила – правила, що встановлюють закономірності між пов'язаними подіями, в загальному випадку мають наступну структуру: з події *A* слідує подія *B* [1, 13, 73, 79, 112, 137, 224]. Вперше були використані для знаходження типових шаблонів покупок в супермаркетах [112, 224], наприклад, якщо користувач *A* придбає товар x_1, то придбає й товар x_2 з ймовірністю *p*.

Продукційні правила – це правила, які мають вигляд: **ЯКЩО** <умова>, **ТО** <подія> [112, 116].

Перед тим як запропонувати продукційні правила для визначення відсутніх коефіцієнтів подоби у рекомендаційній системі, було розроблено асоціативне правило, запропоноване в роботі [54]:

Якщо коефіцієнт схожості користувачів *A* та *B* дорівнює 1, тобто, користувачі «повністю» схожі, **та** при цьому коефіцієнт подоби між користувачами *A* та *C* дорівнює *x*, **то** коефіцієнт подоби між користувачами *B* та *C* також дорівнює *x*.

Зобразимо це правило схематично на рис. 3.9.

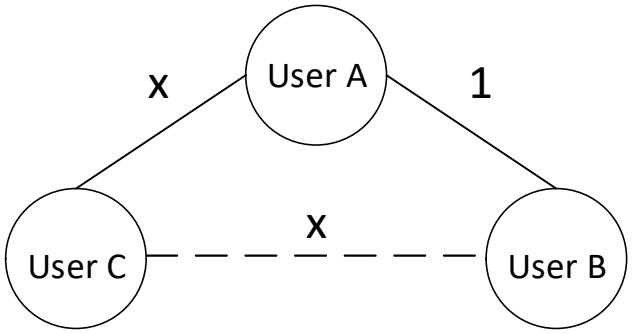


Рис. 3.9. Схематичне зображення асоціативного правила визначення подоби користувачів

Застосування даного правила дозволило збільшити покриття простору користувачів в середньому на 11% та покриття каталогу товарів в середньому на 16% без суттєвих коливань точності роботи рекомендаційної системи [54].

Було запропоновано продукційні правила для визначення невідомих коефіцієнтів подоби користувачів [64]. Додавання продукційних правил дозволить визначати значення коефіцієнту подоби між користувачами *C* та *B* у тому випадку, якщо коефіцієнт подоби між користувачами *A* та *B* дорівнює не одиниці, а якомусь іншому значенню. В такому випадку за допомогою продукційних правил можна спробувати визначити мінімально та максимально можливе значення для коефіцієнту подоби між користувачами *C* та *B*, а також для використання у розрахунках певним чином обрати конкретне значення з цього діапазону.

Запропоновані наступні продукційні правила:

1. Якщо коефіцієнт подоби користувачів *A* та *B* дорівнює *y* **та** коефіцієнт подоби між користувачами *A* та *C* дорівнює *x*, **то** коефіцієнт подоби між користувачами *B* та *C* дорівнює $z = [min(x, y), 1 - |x - y|]$ (рис. 3.10).

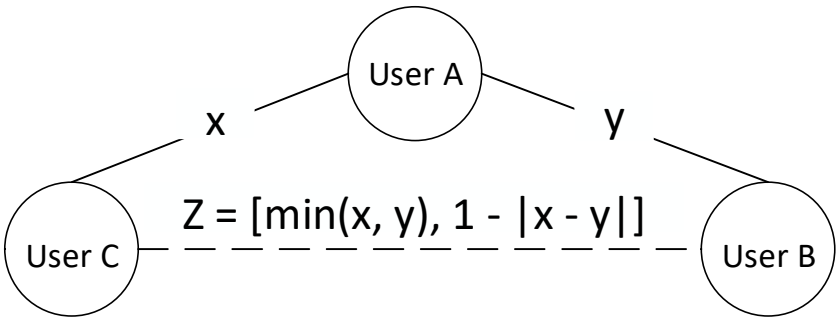


Рис. 3.10. Схематичне зображення продукційного правила 1 визначення подоби користувачів

2. **Якщо** для користувачів *C* та *B* є множина користувачів $\{A_1, A_1, \ldots, A_n\}$, для яких відомі коефіцієнти подоби з користувачами *B* та *C*, **то** коефіцієнт подоби між *B* та *C* дорівнює $z = z_1 \cap z_2 \cap \ldots \cap z_n$ (рис. 3.11).

На основі запропонованих продукційних правил було розроблено та реалізовано відповідні алгоритми та проведено експерименти для перевірки ефективності та якості їх роботи.

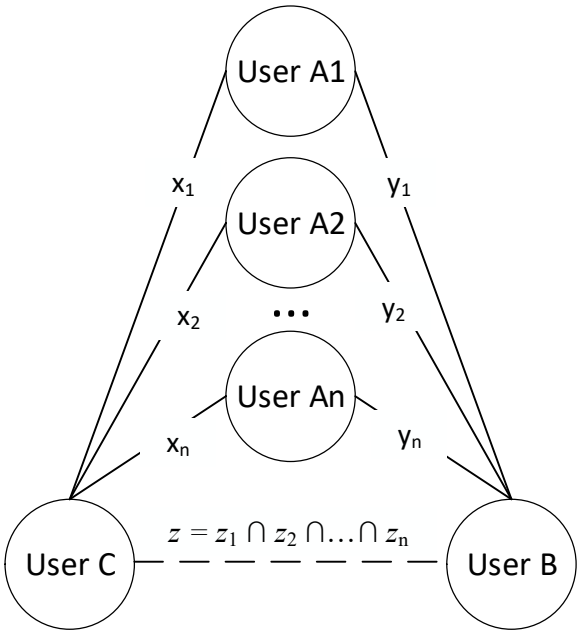


Рис. 3.11. Схематичне зображення продукційного правила 2 визначення подоби користувачів

Розроблювана рекомендаційна система тестувалася на відкритому наборі даних MovieLens Datasets науково-дослідної лабораторії створеному у відділі комп'ютерних наук та інженерії в Університеті Міннесоти [30]. Для розробки рекомендаційної системи було використано мову програмування Python та графову базу даних Neo4j. Під час кожного експерименту з набору даних MovieLens Datasets обиралося випадковим чином N_u користувачів. Для обчислення коефіцієнтів подоби користувачів використовувалася косинусна подоба (1.7). Було проведено 12 серій експериментів.

Дані кожного користувача ділилися на робочі та тестові на основі часових міток. Прогнози вподобань користувачів поділені на позитивні, коли прогнозується позитивна оцінка для пари користувач-об’єкт, та негативні, коли прогнозується негативна оцінка. Позитивним прогнозом вважалися прогнози оцінок від 3.5 до 5.0. Негативним прогнозом вважалися прогнози оцінок від 0.5 до 3.0.

Спочатку будувалися прогнози на основі робочих даних, потім їх якість перевірялася на основі тестових даних. Вірним вважався прогноз, який

співпадав з оцінкою користувача у тестових даних. Помилковим вважався прогноз, який не співпадав з оцінкою у тестових даних.

Результати проведених експериментів наведені у таблиці 3.4, у ній використані наступні скорочення:

– **ВМ** – відомий метод колаборативної фільтрації на основі моделі сусідства;

– **ПП** – розроблений метод з використанням продукційних правил.

На основі результатів експериментів можна зробити висновок, що розроблений метод дозволяє збільшити загальну кількість рекомендацій в середньому для одного користувача у 1.8 разів та підвищити покриття каталогу об’єктів в 2 рази без суттєвих коливань точності та повноти роботи системи. Точність та повнота можуть як збільшуватися так і зменшуватися, що залежить від характеристик вхідних даних. Отже, користувачі зможуть отримувати списки рекомендацій більшої довжини, а їх наповнення буде більш різноманітним за рахунок врахування більшої кількості вподобань різних користувачів при розрахунках.

Таблиця 3.4. Результати тестування розробленого методу

№ експерименту	Кількість користувачів	Загальна кількість спрогнозованих рейтингів		Точність (Precision)		Повнота (Recall)		Покриття простору користувачів, %		Покриття каталогу, %	
		ВМ	ПП	ВМ	ПП	ВМ	ВВ	ВМ	ПП	ВМ	ПП
1	50	10358	22550	0.7155	0.7450	0.7904	0.7760	50.00	52.00	15.06	40.11
2	50	70211	108648	0.7570	0.7526	0.5751	0.5577	61.00	61.00	39.39	73.72
3	50	14456	34880	0.7241	0.8205	0.8400	0.7804	54.00	54.00	19.40	62.72
4	50	8527	25286	0.8461	0.7857	0.9166	0.8684	46.00	46.00	22.68	22.68
5	50	18285	41838	0.7778	0.7401	0.7001	0.6667	74.00	74.00	34.84	87.53
6	50	14150	32984	0.8401	0.8387	0.8401	0.8666	50.00	50.00	32.56	79.35
7	100	39602	68929	0.6935	0.7176	0.6417	0.6703	52.00	52.00	23.89	48.39
8	100	71496	120303	0.7777	0.8000	0.9800	0.9800	46.00	46.00	42.99	77.89
9	100	46614	88136	0.8571	0.8787	0.8888	0.9062	49.00	49.00	32.44	68.68
10	100	78858	112284	0.7761	0.7794	0.6933	0.6883	61.00	61.00	42.06	69.51
11	100	47228	70700	0.6630	0.5799	0.8133	0.8260	57.99	57.99	31.70	67.24
12	100	38751	83261	0.7254	0.7460	0.8043	0.8034	56.00	56.00	27.19	59.11
Середні значення показників:				**0.7627**	**0.7653**	**0.7903**	**0.7825**	**54.74**	**54.91**	**30.35**	**63.07**

Отже, запропоновано продукційні правила для визначення відсутніх коефіцієнтів подоби користувачів у рекомендаційних системах для підвищення якості їх роботи.

Розроблено метод колаборативної фільтрації з використанням запропонованих продукційних правил.

Проведено серію експериментів, які показали, що розроблений метод підвищує загальну кількість рекомендацій та покриття каталогу об’єктів без суттєвих коливань точності та повноти роботи системи. При використанні розробленого методу значно збільшується загальна кількість рекомендацій, що дозволяє формувати більш довгі списки рекомендацій користувачам.

Слід зазначити, що при наявності бот-мереж у системі даний метод буде втрачати точність, адже при наявності профілів ботів розроблені продукційні правила даватимуть викривлені результати, тому даний метод слід використовувати тільки після ідентифікації та нейтралізації профілів ботів.

3.2.4. Розробка гібридного методу колаборативної фільтрації з використанням продукційних правил та врахуванням показників активності користувачів

На основі розроблених методів колаборативної фільтрації було створено гібридний метод, що поєднав в собі:

1) відомий метод колаборативної фільтрації на основі моделі сусідства;

2) запропонований метод колаборативної фільтрації з використанням продукційних правил для визначення відсутніх коефіцієнтів подоби між користувачами;

3) запропонований експертно-орієнтований метод колаборативної фільтрації з врахуванням показників активності користувачів.

Була застосована послідовна стратегія гібридизації, алгоритми на основі даних методів запускалися по черзі у порядку наведеному вище.

Було розроблено та реалізовано відповідні алгоритми для реалізації

запропонованого гібридного методу.

Були проведені експерименти для перевірки якості та точності роботи розробленого гібридного методу синтезу рекомендаційної системи та порівняння його показників з відомими методами колаборативної фільтрації на основі моделі сусідства та на основі моделі матричної факторизації. Результати тестувань наведені у таблиці 3.5.

У таблиці були використані наступні скорочення:

– МС – відома колаборативна фільтрація на основі моделі сусідства;

– ФМ – відома колаборативна фільтрація на основі факторизації матриць;

– Г+ – розроблений гібридний метод колаборативної фільтрації.

Таблиця 3.5. Результати експериментів з тестування точності роботи гібриду запропонованих методів та його порівняння з відомими методами колаборативної фільтрації

№ експерименту	Кількість користувачів	Точність (Precision)			Повнота (Recall)			Покриття простору користувачів, %			Покриття каталогу, %			RMSE		
		МС	ФМ	Г+	МС	ФМ	Г+	МС	ФМ	Г+	МС	ФМ	Г+	МС	ФМ	Г+
1	100	0.8731	0.8420	0.8873	0.8058	0.6398	0.8221	77.89	100.0	100.0	96.50	100.0	99.00	0.954	1.796	0.952
2	100	0.8932	0.8456	0.8970	0.7243	0.6506	0.7316	77.89	100.0	100.0	97.50	100.0	99.50	0.826	1.691	0.841
3	100	0.9178	0.8557	0.9173	0.8265	0.6452	0.8200	91.57	100.0	100.0	97.50	100.0	98.50	0.919	1.765	0.911
4	100	0.9326	0.9031	0.9290	0.8779	0.6242	0.8727	86.31	100.0	100.0	98.00	100.0	99.50	0.972	1.902	1.000
5	100	0.8726	0.7826	0.8808	0.7982	0.6455	0.7954	94.73	100.0	100.0	100.0	100.0	100.0	0.988	1.902	0.993
6	100	0.8586	0.8224	0.8837	0.8066	0.6424	0.7934	75.78	100.0	100.0	96.50	100.0	99.00	0.877	1.777	0.878
7	100	0.8600	0.8433	0.8816	0.8376	0.6543	0.8495	81.05	100.0	100.0	98.50	100.0	99.50	0.967	1.767	0.951
8	100	0.9151	0.8713	0.8930	0.7475	0.6526	0.7529	78.94	100.0	100.0	97.00	100.0	98.50	1.076	1.789	1.068
9	100	0.8946	0.8530	0.8724	0.8109	0.6478	0.8275	85.26	100.0	100.0	97.00	100.0	99.00	0.918	1.761	0.927
10	100	0.8903	0.8703	0.9123	0.8102	0.6932	0.8445	86.31	100.0	100.0	98.50	100.0	99.50	0.932	1.716	0.927
11	100	0.8495	0.8033	0.8514	0.8109	0.6064	0.8100	91.57	100.0	100.0	99.50	100.0	99.50	0.888	1.797	0.883
12	100	0.8858	0.8484	0.8768	0.8644	0.6267	0.8260	75.78	100.0	100.0	94.00	100.0	100.0	0.870	1.743	0.907
13	100	0.8722	0.8197	0.8904	0.7904	0.5901	0.8036	89.47	100.0	100.0	100.0	100.0	100.0	0.896	1.772	0.896
14	100	0.8934	0.8565	0.8918	0.8178	0.6718	0.8187	94.73	100.0	100.0	99.50	100.0	100.0	0.797	1.686	0.803
15	100	0.8754	0.8237	0.8911	0.8690	0.5959	0.8554	72.63	100.0	100.0	93.50	100.0	98.50	0.930	1.849	0.926
16	100	0.8752	0.8476	0.9022	0.8070	0.6545	0.7877	86.31	100.0	100.0	99.00	100.0	100.0	1.015	1.735	1.003
17	100	0.8613	0.8053	0.8723	0.8299	0.6003	0.8306	87.36	100.0	100.0	99.00	100.0	99.00	0.880	1.782	0.873
18	100	0.8881	0.8531	0.9015	0.7541	0.6594	0.7552	69.47	100.0	100.0	92.50	100.0	98.50	0.940	1.793	0.978
19	100	0.9051	0.8418	0.8855	0.8378	0.6565	0.8286	85.26	100.0	100.0	98.50	100.0	99.50	0.985	1.729	0.947
20	100	0.9419	0.8998	0.9365	0.8326	0.6696	0.8400	84.21	100.0	100.0	98.50	100.0	99.50	0.848	1.736	0.859
Сер. знач.:		**0.8877**	**0.8444**	**0.8926**	**0.8129**	**0.6413**	**0.8132**	**83.62**	**100.0**	**100.0**	**97.55**	**100.0**	**99.32**	**0.923**	**1.774**	**0.926**

Розроблений метод, на відміну від методу колаборативної фільтрації на основі моделі сусідства (МС), дозволяє забезпечити 100% покриття користувачів та 99% покриття каталогу товарів без зменшення точності формування рекомендацій, а на відміну від методу колаборативної фільтрації на основі матричної факторизації (ФМ), дозволяє отримати вищі значення точності і повноти на 5% і 17% відповідно та менше в 1.9 разів значення помилки прогнозування рекомендацій (RMSE).

3.2.5. Дослідження показників стійкості розробленого гібридного методу колаборативної фільтрації до внутрішніх дестабілізуючих факторів

Було проведено серію експериментів для перевірки стійкості розробленого гібридного методу колаборативної фільтрації до внутрішніх дестабілізуючих факторів на прикладі проблеми холодного старту.

Результати експериментів наведені у таблиці 3.6.

В проведених експериментах здійснювалося визначення наступних показників стійкості:

– Середній зсув прогнозувань оцінок для користувачів.

– Середній зсув кількості елементів у списках рекомендацій користувача.

– Середній зсув прогнозувань оцінок для об’єктів.

– Середній зсув кількості потраплянь у списки рекомендацій для об’єктів.

Було проведено порівняння ідеального режиму роботи рекомендаційної системи, коли повністю відсутня проблема холодного старту, з ситуацією, коли 30% користувачів знаходяться у холодному старті для тих самих наборів даних.

Показники стійкості обчислювалися наступним чином.

Нехай U_T та I_T – це набори користувачів та елементів відповідно у тестових даних. Для кожної пари user-item (u, i) зсув прогнозування визначався як:

$$\Delta_{u,i} = p_{u,i} - p'_{u,i}, \tag{3.37}$$

де p – прогнози, коли у системі немає користувачів у холодному старті, p' – прогнози, коли у системі 30% користувачів знаходяться у холодному старті.

Середній зсув прогнозування оцінок для користувачів визначався наступним чином:

$$\overline{\Delta}_i = \left| \frac{\sum_{i=1}^{n} \frac{\Delta_{i,u}}{m_i}}{n} \right|, \tag{3.38}$$

де i – індекс об'єкта, u – індекс користувача, m_i – кількість користувачів, що оцінили i-тий об'єкт, n – кількість об'єктів у системі.

В даному випадку неважливо у яку сторону відбувається зсув – у бік збільшення чи зменшення рейтингів, а важливий лише його розмір, тому бралося абсолютне значення показника для зручності порівняння результатів для різних наборів даних. Ділення на кількість об'єктів у системі здійснюється для нормалізації результатів.

Аналогічно визначається середній зсув прогнозування оцінок для всіх об'єктів у тестовій виборці:

$$\overline{\Delta}_u = \left| \frac{\sum_{u=1}^{m} \frac{\Delta_{i,u}}{n_u}}{m} \right|, \tag{3.39}$$

де i – індекс об'єкта, u – індекс користувача, n_u – кількість об'єктів оцінених u-тим користувачем, m – кількість користувачів у системі.

Ділення на кількість користувачів у системі здійснюється для нормалізації результатів.

Середній зсув кількості елементів у списках рекомендацій користувача визначався як:

$$HitRatio_i = \frac{\sum_{i=1}^{n} \frac{H_{i,u}}{m_i}}{n} \tag{3.40}$$

де $H_{i,u}$ – потрапляння об'єкта i у список рекомендацій користувачу u.

Середній зсув кількості потраплянь у списки рекомендацій для об’єктів визначався як:

$$HitRatio_u = \frac{\sum_{u=1}^{m} \frac{H_{i,u}}{n_u}}{m} \tag{3.41}$$

Таблиця 3.6. Результати тестування стійкості гібриду розроблених методів та порівняння з існуючими методами колаборативної фільтрації

№ експерименту	Середній зсув прогнозувань оцінок для користувачів			Середній зсув кількості елементів у списках рекомендацій користувача			Середній зсув прогнозувань оцінок для об’єктів			Середній зсув кількості потраплянь у списки рекомендацій для об’єктів		
	ВМ	**ФМ**	**Г+**	**ВМ**	**ФМ**	**Г+**	**ВМ**	**ФМ**	**Г+**	**ВМ**	**ФМ**	**Г+**
1	0.0300	0.0150	0.0183	0.0592	0.1062	0.0310	0.0152	0.0074	0.0531	0.0296	0.0096	0.0155
2	0.0251	0.0147	0.0162	0.0268	0.0976	0.0088	0.0152	0.0074	0.0096	0.0296	0.0531	0.0155
3	0.0188	0.0147	0.0133	0.0328	0.1078	0.0146	0.0126	0.0073	0.0081	0.0134	0.0488	0.0044
4	0.0298	0.0154	0.0183	0.0549	0.1010	0.0333	0.0094	0.0072	0.0066	0.0164	0.0539	0.0073
5	0.0224	0.0147	0.0138	0.0438	0.1148	0.0092	0.0149	0.0076	0.0093	0.0274	0.0505	0.0166
6	0.0280	0.0151	0.0178	0.0469	0.1184	0.0220	0.0112	0.0073	0.0069	0.0219	0.0574	0.0046
7	0.0198	0.0155	0.0131	0.0483	0.1244	0.0153	0.0143	0.0075	0.0090	0.0235	0.0592	0.0110
8	0.0209	0.0149	0.0142	0.0324	0.1107	0.0255	0.0099	0.0077	0.0622	0.0241	0.0066	0.0077
9	0.0277	0.0154	0.0177	0.0358	0.1200	0.0216	0.0103	0.0074	0.0071	0.0162	0.0554	0.0127
10	0.0224	0.0147	0.0136	0.0311	0.1028	0.0167	0.0140	0.0077	0.0091	0.0179	0.0600	0.0108
11	0.0249	0.0146	0.0156	0.0443	0.0891	0.0172	0.0113	0.0073	0.0068	0.0155	0.0514	0.0083
12	0.0203	0.0147	0.0150	0.0401	0.1132	0.0360	0.0126	0.0073	0.0078	0.0221	0.0446	0.0086
13	0.0191	0.0142	0.0131	0.0337	0.0988	0.0228	0.0103	0.0073	0.0075	0.0200	0.0566	0.0180
14	0.0243	0.0151	0.0163	0.0135	0.1069	0.0044	0.0096	0.0070	0.0066	0.0169	0.0494	0.0114
15	0.0287	0.0150	0.0165	0.0562	0.1076	0.0169	0.0123	0.0075	0.0082	0.0067	0.0534	0.0022
16	0.0258	0.0150	0.0159	0.0115	0.1061	0.0000	0.0144	0.0074	0.0083	0.0281	0.0538	0.0084
17	0.0169	0.0143	0.0133	0.0219	0.1053	0.0154	0.0130	0.0075	0.0080	0.0057	0.0530	0.0000
18	0.0286	0.0154	0.0173	0.0684	0.1113	0.0314	0.0084	0.0070	0.0067	0.0109	0.0526	0.0077
19	0.0234	0.0152	0.0149	0.0304	0.1166	0.0138	0.0144	0.0076	0.0089	0.0342	0.0556	0.0157
20	0.0219	0.0151	0.0132	0.0236	0.1164	0.0022	0.0117	0.0076	0.0075	0.0152	0.0583	0.0069
С. з.:	**0.0239**	**0.0149**	**0.0153**	**0.0377**	**0.1087**	**0.0179**	**0.0122**	**0.0074**	**0.0128**	**0.0197**	**0.0491**	**0.0096**

Також було запропоновано та використано консолідований показник стійкості рекомендаційної системи:

$$persistence = \sum k_i x_i \,, \tag{3.42}$$

де k_i – коефіцієнт важливості i-того показника стійкості для розроблюваної

рекомендаційної системи; x_i – i-тий показник стійкості рекомендаційної системи з набору обраних для розрахунків показників.

Для розроблюваної рекомендаційної системи формула (3.42) має наступний вигляд:

$$persistence = \overline{\Delta}_u + HitRatio_u + \overline{\Delta}_i + HitRatio_i, \quad (3.43)$$

де усі коефіцієнти k_i мають однакову важливість та рівні 1.

Обчислені консолідовані показники стійкості запропонованого та існуючих методів колаборативної фільтрації наведені у таблиці 3.7.

Таблиця 3.7. Порівняння стійкості відомих та розробленого методів колаборативної фільтрації

№ експ.	Стійкість. Консолідований показник		
	ВМ	**ФМ**	**Г+**
1	0.1340	0.1382	0.1179
2	0.0967	0.1728	0.0501
3	0.0776	0.1786	0.0404
4	0.1105	0.1775	0.0655
5	0.1085	0.1876	0.0489
6	0.1080	0.1982	0.0513
7	0.1059	0.2066	0.0484
8	0.0873	0.1399	0.1096
9	0.0900	0.1982	0.0591
10	0.0854	0.1852	0.0502
11	0.0960	0.1624	0.0479
12	0.0951	0.1798	0.0674
13	0.0831	0.1769	0.0614
14	0.0643	0.1784	0.0387
15	0.1039	0.1835	0.0438
16	0.0798	0.1823	0.0326
17	0.0575	0.1801	0.0367
18	0.1163	0.1863	0.0631
19	0.1024	0.195	0.0533
20	0.0724	0.1974	0.0298
Середні значення:	**0.0937**	**0.1802**	**0.0558**

Розроблений гібридний метод (Г+), на відміну від існуючих методів колаборативної фільтрації, більш стійкий до внутрішнього дестабілізуючого фактора холодного старту користувачів. А саме, в 1.6 разів підвищена

стійкість в порівнянні з методом на основі моделі сусідства (ВМ) та в 3.2 разів в порівнянні з методом на основі факторизації матриць (ФМ).

Висновки до розділу 3

У даному розділі запропоновано математичну модель стійкої рекомендаційної системи для оптимізації загальних витрат на її обслуговування, розроблену на основі методу визначення динаміки ймовірностей перебування системи в своїх можливих станах. На основі розробленої математичної моделі запропоновано спосіб визначення повних витрат підсистеми збору та перерахунку вхідних даних, а також спосіб визначення оптимальної частоти перерахунку вхідних даних, при яких система має мінімальну збитковість.

Також у даному розділі запропоновано гібридний метод колаборативної фільтрації, стійкий до внутрішніх дестабілізуючих факторів. Даний метод є гібридом наступних методів: запропонованого експертно-орієнтованого методу колаборативної фільтрації з врахуванням показників активності користувачів для підвищення стійкості системи в умовах холодного старту, запропонованого методу колаборативної фільтрації з використанням продукційних правил для підвищення стійкості рекомендаційної системи в умовах малої кількості вхідних даних та відомого методу колаборативної фільтрації на основі моделі сусідства.

Було проведено експерименти для порівняння точності та стійкості розробленого гібридного методу з відомими методами колаборативної фільтрації. Експерименти показали, що розроблений гібридний метод, на відміну від методу колаборативної фільтрації на основі моделі сусідства, дозволяє забезпечити 100% покриття простору користувачів та 99% покриття каталогу товарів без зменшення точності формування рекомендацій, а на відміну від методу колаборативної фільтрації на основі матричної факторизації дозволяє отримати вищі значення точності і повноти на 5% і

17% відповідно та менше в 1.9 разів значення помилки прогнозування рекомендацій (RMSE). Також розроблений гібридний метод, на відміну від існуючих методів колаборативної фільтрації, більш стійкий до дестабілізуючого фактору холодного старту користувачів. А саме, в 1.6 разів підвищується стійкість в порівнянні з методом на основі моделі сусідства та в 3.2 разів в порівнянні з методом на основі факторизації матриць.

РОЗДІЛ 4.
МЕТОД ПРОГРАМНОГО ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ КОРИСТУВАЧІВ ТА ОБ'ЄКТІВ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ СОЦІАЛЬНОЇ МЕРЕЖІ АБО КОНТЕНТ-ОРІЄНТОВАНОГО ВЕБ-РЕСУРСУ

У даному розділі запропоновані математичні та програмні імітаційні моделі користувачів, об'єктів та інформаційних процесів рекомендаційної системи, що дозволяють одержувати набори даних для тестування алгоритмів рекомендаційних систем. Крім поведінки звичайних користувачів системи було здійснено моделювання також поведінки ботів на основі різних відомих моделей інформаційних атак на рекомендаційні системи з метою формування наборів даних, що можна використати для тестування стійкості системи до зовнішніх дестабілізуючих факторів.

Для проведення наукових досліджень в області рекомендаційних систем треба мати у розпорядженні набори даних про дії користувачів та об'єкти деякої системи для створення та тестування якості списків рекомендацій. Такі набори даних можна одержати наступними способами: 1) мати доступ до бази даних реального веб-ресурсу чи додатку; 2) скористатися одним з існуючих відкритих наборів даних; 3) створити додаток для автоматичного збору відкритих даних з веб-ресурсів; 4) створити програмну модель рекомендаційної системи для генерації даних про її користувачів та об'єкти.

Найпростіший спосіб з перерахованих – використати відкритий набір даних. В той же час, найбільш повна інформація для створення та тестування якості списків рекомендацій знаходиться у адміністраторів та власників веб-сайтів. Також вони мають можливість відслідковувати реакцію користувачів на сформовані рекомендації у режимі реального часу. Але у переважної більшості дослідників немає можливості одержати доступ до таких даних. В такому разі, якщо інформації з відкритих наборів даних недостатньо, залишається два шляхи – парсинг відкритих даних з веб-сайтів або створення

програмної імітаційної моделі рекомендаційної системи на основі відомої інформації про деяку систему (веб-сайт або додаток).

Збір відкритих даних з веб-ресурсів може відбуватися двома способами – або з використанням API-функцій [209, 222], або шляхом розробки та використання власного веб-кроулера [169, 180, 197, 222]. API-функції легкі у використанні, але не всі сайти надають можливість зібрати необхідну інформацію завдяки API-функціям – такі функції можуть бути повністю відсутніми або мати обрізаний функціонал та ряд обмежень і умов для використання. Також час від часу інтерфейс API-функцій змінюється, а їх нові версії можуть містити менше можливостей [171].

При використанні власного web-кроулера можна зібрати будь-яку відкриту у вільний доступ інформацію з веб-сайту. Але написання веб-кроулера – значно складніша задача, необхідно розібратися в html-верстці чужого веб-сайту, власноруч розробити функції для збору даних з html-коду, вносити зміни або повністю переписувати веб-кроулер кожного разу, коли на веб-сайті змінюється html-верстка. Також великі великі веб-сайти блокують автоматичний збір даних, якщо помічають його. Одним з перспективних способів парсингу веб-сайтів є застосування бібліотеки для автоматизації дій веб-браузера та його тестування Selenium (для мов Java, C#, Ruby, Python, Javascript) [96, 198]. Дана бібліотека створює часові затримки між своїми діями характерні для поведінки людини, чим обходить захист веб-сайтів від автоматичного збору даних, але це ж робить її роботу досить повільною і малопридатною для збору великих об’ємів даних.

Застосування відкритих наборів даних, а також збір даних за допомогою API-функцій або web-кроулера з веб-сайтів не дозволяють досліджувати реакцію користувачів на запропоновані рекомендації та вплив бот-мереж.

Виходячи з наведеної інформації у даній роботі було вирішено для дослідження рекомендаційних систем поєднати використання відкритих наборів даних та власної програмної імітаційної моделі користувачів і об’єктів та інформаційних процесів рекомендаційної системи.

4.1. Розробка методу програмного імітаційного моделювання користувачів та об'єктів рекомендаційної системи

Дамо наступне формальне визначення загальній рекомендаційній системі:

Рекомендаційна система – це система, яка працює з множиною елементів $E = U \cup O$, що складається з підмножини об'єктів O та підмножини користувачів U, а саме, збирає дані про ці елементи, прогнозує вподобання (оцінки, інтерес тощо) користувачів, і на основі них формує списки рекомендацій $R = \left(r_1, r_2, ..., r_n \middle| n \le N\right)$ з одних елементів системи E_1 для інших елементів системи E_2 довжиною n, де $E_1, E_2 \subset E$. Також рекомендаційна система має множину ознак $A = \{a_1, a_2, ..., a_m\}$, які можуть бути наявними у елементів системи. Кожен елемент рекомендаційної системи має множину коефіцієнтів $K = \{k_{i,1}, k_{i,2}, ..., k_{i,q}\}$, що характеризують ступінь його приналежності до кожної з можливих ознак. А також кожен користувач системи має інтереси, представлені множиною коефіцієнтів $K = \{k_{u,1}, k_{u,2}, ..., k_{u,q}\}$ – що характеризують ступінь його інтересу до кожної з ознак елементів системи. Елементи списків рекомендацій r_i можуть бути впорядковані за різними принципами, але найчастіше вони впорядковані за значеннями прогнозованих оцінок (або ступенем можливого інтересу) для них одержувача рекомендацій.

Використовуючи дане формальне визначення опишемо можливі способи застосування рекомендаційних систем.

1. Створення списку рекомендацій для користувача, що містить об'єкти (наприклад, рекомендація відвідувачу Інтернет-магазину товарів):

$$R = \{(u_i, O_r) | u_i \in U, O_r \subset O\}, \tag{4.1}$$

де u_i – користувач, для якого формуються рекомендації; O_r – множина рекомендованих об'єктів.

2. Створення списку рекомендацій для групи користувачів, що містить

об’єкти (наприклад, рекомендація відеороликів користувачам, що переглядають одночасно одне й те саме потокове відео):

$$R = \{(U_c, O_r) | U_c \subset U, O_r \subset O\}, \quad (4.2)$$

де U_c – група користувачів, для якої формуються рекомендації; O_r – рекомендовані об’єкти.

3. Створення списку рекомендацій для користувача, що містить інших користувачів (наприклад, рекомендація друзів у соціальній мережі):

$$R = \{(u_i, U_r) | u_i \in U, U_r \subset U\}, \quad (4.3)$$

де u_i – користувач, для якого формуються рекомендації; U_r – рекомендовані користувачі.

4. Створення списку рекомендацій для групи користувачів, що містить інших користувачів (наприклад, рекомендація співтовариству у соціальній мережі підписку на інші співтовариства):

$$R = \{(U_c, U_r) | U_c, U_r \subset U\}, \quad (4.4)$$

де U_c – група користувачів, для яких формуються рекомендації; U_r – рекомендовані групи користувачів.

А також запропонуємо ще декілька можливих способів використання рекомендаційних систем, якщо використовувати їх для пошуку та фільтрації даних:

5. Пошук користувачів, що можуть зацікавитися певним об’єктом, відранжовані по рівню можливої зацікавленості (наприклад, пошук можливих покупців для товару, аудиторії для перегляду таргетованої реклами):

$$R = \{(o_j, U_r) | o_j \in O, U_r \subset U\}, \quad (4.5)$$

де o_j – об’єкт, для якого здійснюється пошук; U_r – рекомендовані користувачі.

6. Пошук користувачів, що можуть зацікавитися певною множиною об’єктів, відранжовані по рівню можливої зацікавленості (наприклад, пошук користувачів, що зацікавляться певним блогом з деяким набором статей):

$$R = \{(O_s, U_r) | O_s \subset O, U_r \subset U\}, \quad (4.6)$$

де O_s – група об'єктів, для яких здійснюється пошук; U_r – рекомендовані користувачі.

7. Пошук об'єктів схожих на заданий (наприклад, інформаційний пошук по шаблону, відранжований по ступеню інтересу користувача до них):

$$R = \{(u_i, o_j, O_r) | u_i \in U; o_j \in O; O_r \subset O\}, \quad (4.7)$$

де u_i – користувач, який здійснює пошук; o_j – об'єкт, який являється шаблоном пошуку; O_r – рекомендовані об'єкти.

8. Формування списків об'єктів, схожих на певну групу об'єктів, на основі оцінок (відгуків) різних користувачів для них (наприклад, формування списку контенту, що відповідає певним ознакам – (не)спам, (не)деструктивний контент, тематичний контент тощо):

$$R = \{(U_c, O_s, O_r) | U_c \subset U; O_s, O_r \subset O\}, \quad (4.8)$$

де U_c – група користувачів, на основі відгуків яких формуються списки; O_s – група об'єктів, на основі яких здійснюється пошук; O_r – рекомендовані об'єкти.

Найчастіше рекомендаційні системи застосовуються для створень рекомендацій користувачам або групам користувачів за схемами описаними у перших трьох випадках (4.1)-(4.3).

Рекомендації R користувачам системи формуються на основі даних D, зібраних про елементи системи за допомогою деякого алгоритму фільтрації даних:

$$R(u_i, D, N) = (r_1, r_2, ..., r_n | n \le N), \quad (4.9)$$

$$D = \{D_1, D_2, D_3, D_4, D_5\}, \quad (4.10)$$

де D – види доступних рекомендаційній системі даних, які можна використати для формування рекомендацій: D_1 – дані, що прямо показують відношення користувачів до переглянутих елементів системи: оцінки або лайки/дизлайки, які користувачі виставляють елементам системи; D_2 – дані, які опосередковано показують відношення користувачів до переглянутих

елементів системи: перегляди сторінок елементів системи, здійснені покупки, написані коментарі тощо; D_3 – дані, які дозволяють віднести користувача до певної групи користувачів з відомими інтересами: дані з профілю, зокрема, демографічні дані, контекстні дані; D_4 – властивості та опис об'єктів системи; D_5 – результати опитувань, якщо система використовує опитування.

Дані типів D_1, D_2 та D_5 по суті являються зворотним зв'язком від користувачів для рекомендаційної системи. D_3 та D_4 є описовими даними.

Довжина списку рекомендацій залежить від вимог конкретного додатку або веб-ресурсу.

Базова вимога до якості роботи рекомендаційної системи, це забезпечення мінімальної різниці між прогнозованими та реальними вподобаннями користувачів:

$$d(P,R) \to \min, \tag{4.11}$$

де вектор $R = (r_1, r_2, ..., r_n)$ містить список прогнозованих рекомендацій (оцінок) користувача, впорядкований по спаданню за величиною оцінок; вектор $P = (p_1, p_2, ..., p_n)$ містить справжні вподобання (оцінки) користувача, невідомі системі на етапі формування списку рекомендацій.

Дані рекомендаційної системи зручно представляти у вигляді графу, де користувачі та об'єкти – вершини графу, а дії користувачів (перегляди, оцінки тощо) та результати роботи рекомендаційної системи (рекомендації, коефіцієнти подоби тощо) – ребра. Тож моделювати дані рекомендаційної системи та поведінку користувачів зручно за допомогою складних графів. Для якісного моделювання елементів та процесів рекомендаційної системи необхідно знати властивості, якими вони володіють.

Рекомендаційна система представляє собою мережу зв'язків між користувачами та об'єктами веб-ресурсу. Її можна розглядати як різновид соціального графу. *Соціальний граф* – граф, вузлами якого є соціальні об'єкти (наприклад, користувачі, співтовариства користувачів, об'єкти контенту тощо), а ребрами – соціальні зв'язки між ними.

Існує декілька підходів до моделювання соціальних мереж та соціальних

графів, зокрема, найчастіше застосовують наступні [5, 31, 45, 88, 119, 120, 185, 206, 207, 212]: моделі випадкових графів, моделі складних мереж, теоретико-ігрові моделі тощо.

Для моделювання зв'язків між користувачами та об'єктами рекомендаційної системи було вирішено використати теорію *складних (комплексних) мереж* [2, 5, 127, 136, 201, 211]. Було проведено дослідження властивостей складних мереж та методів їх моделювання.

4.1.1. Дослідження методів моделювання складних мереж

Складні мережі (complex networks) – це стохастичні мережі з нетривіальною топологією, зокрема, вони відрізняються від класичних стохастичних мереж наявністю невеликої кількості вузлів з великим числом зв'язків (такі вершини називаються хабами) [2, 5, 127, 136, 201, 211]. Більшість реальних мереж – складні. Складні мережі прийнято ділити на: технічні мережі (наприклад, комп'ютерні мережі, транспортні мережі), біологічні мережі (наприклад, мережі метаболізму, екологічні мережі), соціальні мережі (наприклад, мережі друзів, мережі цитування, мережі телефонного зв'язку) тощо. Найкраще досліджені складні мережі як модель соціальних мереж.

У складних мереж, що відображають соціальні зв'язки, є наступні основні властивості [2, 5-8, 29, 88, 93, 121, 127, 136, 185, 207, 211]:

1. *Безмасштабність.* Розподіл степенів вузлів (vertex degree, кількості зв'язків у вузлів) за степеневим розподілом.

2. *"Тісний світ"* (small-world network). Невеликий діаметр мережі.

3. Високий *коефіцієнт кластеризації* та високий *коефіцієнт транзитивності.* Якщо в соціальній мережі є учасники *A*, *B* та *C*, і є соціальні зв'язки між *A* та *B*, а також між *A* та *C*, то досить висока ймовірність, що у *B* та *C* також є соціальні зв'язки.

4. *Гігантська зв'язна компонента*. Тобто, більше 80% вузлів пов'язані

між собою.

5. Присутні ієрархічні зв'язки.

6. Присутні складні кластерні утворення (*кліки*, *клани* тощо).

7. *Асортативність.* В широкому розумінні асортативність – це виникнення зв'язків між вершинами, які чимось схожі між собою. У вузькому розумінні асортативність – це виникнення зв'язків між вершинами з великою кількістю зв'язків.

Розглянемо основні відомі моделі генерації стохастичних та складних мереж.

Відомою моделлю генерації складних мереж є **модель Барабаши-Альберт (Barabasi-Albert model)** [2, 5, 6, 31, 110, 121]. Автори даної моделі показали, що для виникнення безмасштабних мереж необхідна наявність двох умов:

1. *Ріст.* Починаючи з невеликого числа n_0 вузлів, на кожній новій часовій ітерації додається один новий вузол з n зв'язками (де $n \leq n_0$), які з'єднують новий вузол з n різними уже існуючими вузлами.

2. *Бажане приєднання* (Preferencial attachment). Ймовірність P, з якою новий вузол утворить зв'язок з деяким уже існуючим вузлом i, тим вища, чим більше зв'язків у i-го вузла, та визначається за формулою:

$$P_i = \frac{k_i}{\sum_j k_j} \ , \tag{4.12}$$

де k_i – степінь i-го вузла, а в знаменнику підраховується сума всіх степенів існуючих у мережі вузлів.

Цим принципом можна пояснити причини виникнення степеневого закону у соціальних мережах, асортативності та малого діаметру мережі.

Перевагами даної моделі є те, що мережа, яку вона генерує володіє властивостями розрідженості, "тісного світу", безмасштабності. Недоліками моделі є те, що результуючий граф сильно залежить від початкового параметру n_0, а також є складність з бажаним приєднанням у випадковому виборі вершин.

Модель Ердеша-Ран'ї (Erdős-Renyi model) [25, 110, 121]. Нехай є множина вершин $V_n = \{v_1, v_2, ..., v_n\}$, а в графі не буде петель, кратних ребер і орієнтації, тому потенційних ребер буде C_n^2. Вершини з'єднуються попарно з ймовірністю $p \in [0;1]$, незалежно від інших вершин. У даній моделі відсутнє бажане приєднання. Дана модель дозволить створити стохастичний граф, але він не буде мати важливих властивостей складних мереж, а саме степеневого закону розподілу степенів вершин та високого коефіцієнту кластеризації.

Модель Болобаша-Ріордана (Bollobas-Riordan model) [10, 11, 110]. Спочатку будується множина випадкових графів $\{G_1^n\}$, в якій у графу з номером *n* число вершин та ребер рівне *n.* Потім ця множина перетворюється в множину $\{G_k^n\}$, в якій у графу з номером *n* число вершин рівне *n*, а число ребер рівне *kn*, $k \in N$. Дана модель генерує складні мережі та добре збігається з емпіричними даними.

Дані моделі дозвволяють моделювати структуру соціального графу. Для дослідження соціальних процесів необхідно моделювати динамічну складну мережу. Моделлю динамічної мережі може бути **динамічний граф** [133].

Динамічний граф D, представляє собою послідовність класичних графів G_k, перехід між якими описується різними графовими операціями $\varphi(G_k) = G_{k+1}$. Графові операції можна поділити на базові та складні [133]. До базових операцій відносяться операції:

– додавання/видалення ребра;

– додавання/видалення вершини.

Будь-яку складну графову операцію можна описати послідовністю базових графових операцій.

Операція, що здійснює перехід від графу G_k до графу G_{k+1} може бути як базовою так і складною. Для побудови динамічного графу можна використати множину графових операцій $\Phi = \{\varphi^t\}$. Послідовність графів

$G_1, G_2, G_3, \ldots G_m$, називається траєкторією динамічного графу.

Для прогнозування та моделювання змін у динамічному графі можуть використовуватися ієрархічні, ймовірнісні та реляційні моделі, моделі засновані на властивостях соціальних мереж та моделі засновані на властивостях учасників мережі [48, 68, 88, 93, 109, 133, 231].

Проведене дослідження показало, що для моделювання соціального графу рекомендаційної системи найкраще підійдуть моделі, що будуть розроблятися на базі відомої моделі Барабаши-Альберт, оскільки вона дозволяє відтворити найбільшу кількість властивостей соціальних мереж, зокрема, степеневий закон розподілу степенів вершин, високі коефіцієнти кластеризації та транзитивності, асортативність, малий діаметр мережі, що є важливим для моделювання структури зв'язків у рекомендаційній системі. А для симуляції інформаційних процесів слід застосовувати динамічний граф.

4.1.2. Дослідження базових моделей інформаційних атак на рекомендаційні системи

Було проведено дослідження існуючих моделей атак на рекомендаційні системи [176]. Як показало дослідження, основним видом атак на рекомендаційні системи є атаки ін'єкцією профілів, що мають декілька відомих базових моделей, зокрема, випадкову, середню та популярну моделі атаки [22, 46, 60, 61, 69, 76].

Атаки ін'єкцією профілів – це основний тип атак на рекомендаційні системи, що полягає у створенні певної кількості профілів ботів (мережі ботів), які за допомогою узгоджених дій змінюють рейтинги, а як наслідок, і частоту потрапляння цільових елементів системи у списки рекомендацій. Також на етапі підготовки до атаки профілі ботів можуть збирати статистичні дані про вподобання користувачів, використовуючи, списки рекомендацій, які їм надає система, цей необов'язковий початковий етап атаки часто називають атакою зондом [76]. Щоб дізнатися вподобання цільової групи

користувачів, профілі ботів заповнюють даними характерними для цієї групи.

Для атак ін'єкцією профілів завжди буде цільовий об'єкт та об'єкти для наповнення профілю бота [76, 98]. Рейтинг цільового об'єкту зловмиснику треба збільшити або зменшити, а нецільові об'єкти, будуть оцінюватися для наповнення профілю бота та намагання зробити його максимально схожим на профіль справжніх користувачів атакованої системи.

Зловмисник для здійснення впливу повинен досить точно імітувати дії звичайних користувачів, щоб не бути виявленим. А стійка до атак рекомендаційна система повинна працювати так, щоб результат від дій зловмисників був настільки малоефективним, щоб у них не було стимулів продовжувати атаки, а справжні користувачі продовжували одержувати релевантні невикривлені рекомендації.

Часто перед основною атакою зловмисники намагаються відтворити портрет типового користувача певного сегменту користувачів системи, зібрати інформацію про його вподобання та типові особисті характеристики (демографічні дані тощо). З цією метою створюються профілі ботів, схожі на користувачів з потрібного сегменту, щоб одержати списки рекомендацій такі ж як одержують користувачі цього сегменту, а також створюються парсери для збору відкритих даних на веб-ресурсі.

Атакою на рекомендаційну систему будемо вважати узгоджені зусилля великої кількісті профілів щодо зміщення результатів її роботи таким чином, щоб деяка група користувачів або усі користувачі почали отримувати рекомендації, що суперечать їх потребам.

З точки зору зловмисника, найкраща атака проти рекомендаційної системи – це найбільший вплив на рейтинги за найменшу кількість зусиль з його боку.

Принцип дії інформаційних атак ін'єкцією профілів на рекомендаційну систему з колаборативною фільтрацією зображено на спрощеному прикладі на рис. 4.1.

Рис. 4.1. Принцип дії інформаційних атак ін’єкцією профілів на рекомендаційні системи з колаборативною фільтрацією

На рис. 4.1 зображено приклад частини бази даних рейтингів рекомендаційної системи, на яку здійснюється атака. Позитивні оцінки об’єктам позначено символом «+», негативні символом «-», а відсутні, які система буде намагатися спрогнозувати – «?». В даному випадку системі треба спрогнозувати оцінки користувачів 5 та 6 для об’єкту f. Якщо система спрогнозує позитивні оцінки, то об’єкт f з великою ймовірністю потрапить у списки рекомендацій даним користувачам. Тому боти виставляють позитивні оцінки об’єкту f, а іншим об’єктам системи виставляють оцінки схожі на ті, які виставили користувачі 5 та 6. З огляду на те, що звичайні користувачі, схожі на користувачів 5 і 6, які оцінювали об’єкт f, ставили йому негативні оцінки, то без дій зловмисників рекомендаційна система спрогнозувала б низьку оцінку даному об’єкту, і він не потрапив би до рекомендацій.

Для того, щоб нейтралізувати дану атаку треба визначити, які профілі є ботами, та не враховувати їх оцінки при формуванні списків рекомендацій.

Запропонуємо наступну модель профілю бота, що атакує рекомендаційну систему:

Рис. 4.2. Модель профілю бота, що атакує рекомендаційну систему

Як видно з рис. 4.2, профіль бота містить наступні типи оцінок:

– Оцінки об'єктам з множини I_f для імітації дій справжніх користувачів, оцінки даним об'єктам зловмисник змінювати не прагне, а навпаки намагається підібрати для них значення максимально схожі на справжні для цільової групи користувачів, на яких він прагне впливати.

– Оцінки об'єктам з множини I_{ti}, це максимальні (чи близькі до них) оцінки у системі для цільових об'єктів, яким зловмисник прагне підвищити рейтинг.

– Оцінки об'єктам з множини I_{td}, це мінімальні (чи близькі до них) оцінки у системі для цільових об'єктів, яким зловмисник прагне знизити рейтинг.

Кількість цільових об'єктів у бота може варіюватися від 1 до K та міститися тільки у множині I_{ti} або тільки у множині I_{td}, чи в обох цих множинах, а кількість об'єктів для наповнення профілю – від 0 до N.

Дії ботів можуть дати результат тільки тоді, коли вся мережа ботів виставить оцінки всім цільовим об'єктам і при цьому боти не будуть виявлені та нейтралізовані.

Слід зазначити, що успішна інформаційна атака може збільшити різницю між прогнозованими та справжніми вподобаннями (тобто, знизити точність рекомендацій) у разі, якщо інформаційна атака накручує рейтинги

об’єктам, які переважно не подобаються користувачам. Або не змінить точність роботи системи, у випадку коли рейтинги накручуються об’єктам, які переважно подобаються користувачам, і мета атаки – привернути до них більше уваги.

Розглянемо базові моделі атак на рекомендаційні системи з колаборативною фільтрацією.

Найперші моделі атак було запропоновано в [69] – це випадкові та середні моделі атак. Обидві ці моделі атак передбачають генерацію профілів ботів, що будуть випадковим чином виставляти оцінки об’єктам з множини I_f. В наступних роботах [22, 46, 60, 61, 76] розглянуто також більш складні та інформаційноємкі атаки.

Розглянемо найбільш відомі та поширені моделі атак на рекомендаційні системи.

Випадкова атака (Random Attack)

У профілях ботів множина I_f буде заповнюватися оцінками для об’єктів, вибраних випадковим чином. Оцінки обраним об’єктам будуть підбиратися також випадковим чином, але так, щоб вони були близькі до глобальної середньої оцінки у системі, наприклад, буде використовуватися нормальний розподіл з математичним сподіванням, рівним глобальній середній оцінці. Цільовому об’єкту буте ставитися максимальна r_{max} або мінімальна оцінка r_{min}, в залежності від цілей атаки. Знання та зусилля, необхідні для здійснення такої атаки, є досить мінімальними – глобальну середню оцінку у багатьох системах можна легко дізнатися напряму або за допомогою опосередкованих даних. Ця атака не є особливо ефективною.

Середня атака (Average Attack)

Використовує індивідуальні середні значення оцінок кожного об’єкту для створення множини I_f. Інформації для даної атаки треба зібрати більше. Однак середня атака може бути успішною навіть при використанні невеликого набору елементів у I_f, що дозволяє зменшити кількість необхідної для збору інформації. Але ціною такого зменшення необхідних даних буде

велика кількість профілів з практично однаковими множинами оцінених елементів, що буде, звичайно, легко виявити. Ця атака більш ефективна, ніж випадкова. Але вона практично неефективна для алгоритмів колаборативної фільтрації на основі моделі сусідства типу item-based.

Середня атака вимагає відносно великої кількості знань про статистику дій справжніх користувачів у системі. Розумний захист рекомендаційної системи від таких атак буде ускладнювати нападнику збір необхідних даних. Для обходу такого захисту використовуються інші атаки, для яких вимоги до кількості знань значно нижчі.

Розглянемо існуючі атаки, що вимагають менше знань, ніж середня атака, але працюють ефективніше, ніж випадкова атака.

Атака приєднання до більшості (Bandwagon Attack)

Мета цієї атаки – асоціювати атакований об’єкт з невеликою кількістю об’єктів, які часто оцінюються користувачами (назвемо їх широковідомими). Зловмисник створює профілі ботів, що містять у множині I_f оцінки широковідомим об’єктам. Такі профілі мають високу ймовірність бути схожими на велику кількість користувачів, оскільки широковідомі об’єкти – це ті, які оцінили багато користувачів. Дані для такої атаки одержати досить легко. Отже, серед широковідомих об’єктів випадковим чином обирається декілька. Цим об’єктам ставляться максимальні оцінки разом із цільовим об’єктом. Деякій частині об’єктів у множині I_f можуть ставитися випадкові оцінки, наприклад, як у випадковій атаці для того, щоб урізноманітнити профілі ботів. Це досить ефективна атака, але, як і середня, стає неефективною при використанні проти колаборативної фільтрації на основі моделі сусідства типу item-based.

Сегментна атака (Segment Attack)

Основна ідея даної атаки полягає у тому, щоб змінювати рейтинг об’єкту у цільовій групі користувачів з відомими або легко передбачуваними вподобаннями. Тобто, цільовому об’єкту рейтинг буде накручуватися тільки у певному сегменті користувачів, щоб він потрапляв у рекомендації тільки до

них. Інакше, якщо цільовий об’єкт потрапить у рекомендації користувачам з інших сегментів, він може почати отримувати від них низькі оцінки, яких буде більше, ніж накручених оцінок. Щоб здійснити таку атаку треба знайти реальних користувачів, які належать до цільового сегменту та зібрати дані про оцінки, які вони зазвичай виставляють об’єктам системи. Як і в атаці приєднання до більшості, зазвичай визначається, які об’єкти в цільовому сегменті є широковідомими. Цим об’єктам присвоюється максимальне значення оцінки разом із цільовим об’єктом. Щоб забезпечити максимальний ефект від атаки, деякі об’єкти для множини I_f обираються випадково та одержують мінімальні оцінки, що дозволяє зробити профілі ботів різними. Дана атака є ефективною проти алгоритмів колаборативної фільтрації на основі моделі сусідства типу item-based.

Слід зазначити, що усі розглянуті вище моделі атак можуть використовуватися для пониження рейтингу об’єкту, але існують спеціалізовані атаки, які працюють краще, ніж інші, саме для пониження рейтингів.

Розглянемо моделі атак призначені для пониження рейтингів об’єктів рекомендаційної системи.

Атака любов/ненависть (Love/Hate Attack)

Ця атака дуже проста – без вимог до знань. Цільовому об’єкту присвоюється мінімальна оцінка r_{min}, а об’єкти для множини I_f обираються випадковим чином та одержують максимальні оцінки r_{max}. Незважаючи на надзвичайну простоту, це одна з найефективніших атак на пониження рейтингу проти алгоритмів колаборативної фільтрації на основі моделі сусідства типу user-based.

Атака обернена приєднанню до більшості (Reverse Bandwagon Attack)

Це варіант атаки приєднання до більшості, описаний вище, в якому для I_f вибираються широковідомі об’єкти, яким переважна більшість користувачів ставить низькі оцінки. Цим об’єктам у профілях ботів

присвоюються низькі оцінки, а також низька оцінка присвоюється цільовому об’єкту. Таким чином, цільовий об’єкт починає асоціюватися з об’єктами, що не подобаються великій кількості користувачів, і це збільшує ймовірність того, що для об’єкта будуть прогнозуватися низькі оцінки і він не буде потрапляти у списки рекомендацій. Хоча ця атака не є настільки ефективною, як середня атака з великою кількістю знань для user-based систем, вона є дуже ефективною атакою на пониження рейтингів проти item-based систем.

Атаки з низьким рівнем знань використовують широковідомі об’єкти для наповнення профіля бота оцінками для них. Таким чином зловмисник може створити профіль схожий на середньостатистичного користувача, дослідивши оцінки лише широковідомих об’єктів.

Якщо зловмисник знає, який саме алгоритм використовує рекомендаційна система, він може зібрати більше інформації для атаки.

Таким чином атаки можна класифікувати на:

– Атаки з малою кількістю знань – цей тип атак не потребує детальних знань про розподіли оцінок у системі. Він вимагає системно-незалежних знань, які легко можна отримати за допомогою публічних джерел інформації.

– Атаки з великою кількістю знань – зловмиснику потрібно мати якнайбільше знань про алгоритми системи та розподіли оцінок у об’єктів системи. Наприклад, деякі атаки вимагають, щоб зловмисник знав середню оцінку і середнє квадратичне відхилення для кожного об’єкта системи.

Приладом атаки з великою кількістю знань є популярна атака.

Популярна атака (Popular Attack)

Припустимо, що система використовує стандартний user-based алгоритм колаборативної фільтрації, де подібність між користувачами визначається за допомогою кореляції Пірсона. Аналогічним чином, як і в атаці приєднання до більшості, множина I_f заповнюється з використанням широковідомих об’єктів системи. Однак це не гарантує високої схожості між профілем бота та справжніми профілями. Тому популярна атака використовує середні значення оцінок обраних широковідомих об’єктів для заповнення множини

I_f. А для того, щоб урізноманітнити дані профілів, деяким випадковим чином обраним об'єктам ставить оцінки I_f або (r_{min} + 1), або r_{min}, залежно від того, чи є середня оцінка для об'єкту вищою чи нижчою. Така стратегія призведе до позитивних кореляцій між профілями ботів та автентичними профілями. Для визначення широковідомих об'єктів не потрібно багато знань, але для визначення середніх оцінок обраних об'єктів треба зібрати багато інформації. Популярну атаку можна легко налаштувати також для атак на пониження рейтингу. Цю атаку можна виявляти, якщо порівнювати профілі у системі – профілі ботів однієї бот-мережі будуть сильно схожими.

Атака зондом для зібрання інформації (Probe Attack)

Профілі ботів тим важче розпізнати, чим більш схожі їх оцінки на оцінки справжніх користувачів. Знання про реальні вподобання різних сегментів користувачів можна отримати із самої системи через атаку зондом. Для здійснення цієї атаки зловмисник створює насіннєвий профіль, а потім використовує його для одержання рекомендацій з системи. Ці рекомендації формуються на основі інформації системи про реальних користувачів, тому використання одержаних рекомендацій дозволить створити профілі ботів більш схожі на справжніх користувачів. Можна здійснювати зондування невеликої частини користувачів, щоб потім вплинути на малу групу, як у сегментній атаці, або великої частини – щоб одержати інформацію, наприклад, для середньої атаки. Зловмиснику потрібно використовувати лише невелику кількість насіннєвих профілів для того щоб рекомендаційна система сама надала йому потрібну інформацію у вигляді рекомендацій.

Проведені дослідження моделей інформаційних атак на рекомендаційні системи показали, що найбільш простими в реалізації та найменш ресурсозатратними є випадкова та середня атаки, а найбільш ефективною та непомітною, хоча й досить ресурсозатратною, є популярна атака. Інші досліджені моделі атак, по суті, являються їх модифікаціями, наприклад, для атаки лише певного сегменту (сегментна атака) або для атаки лише на пониження рейтингів (атака любов/ненависть), або для зменшення

ресурсозатратності за рахунок зменшення непомітності бот-мережі (атака приєднання до більшості, атака любов/ненависть). Тому в програмній імітаційній моделі користувачів та об'єктів рекомендаційної мережі було вирішено моделювати бот-мережі на основі трьох основних моделей атак – випадкової, середньої та популярної атаки.

4.1.3. Розробка методу програмного імітаційного моделювання поведінки звичайних користувачів та ботів у рекомендаційній системі на основі теорії складних мереж

У даному розділі було створено програмну імітаційну модель поведінки користувачів віртуальної соціальної мережі з рекомендаційною системою [53] для тестування методів генерації списків рекомендацій, зокрема, з метою визначення та порівняння показників точності та стійкості різних методів.

Для розробки методів моделювання структури зв'язків у соціальній мережі з рекомендаційною системою було вирішено взяти за основу, але модифікувати з врахуванням специфіки задачі, принципи, на яких базується модель Барабаши-Альберт (а саме, «ріст» та «бажане приєднання»), так як вона проста в реалізації та дозволяє створити стохастичний граф з властивостями соціальних мереж.

Для моделювання соціальної мережі з рекомендаційною системою необхідно додати у модель мережі, крім користувачів, об'єкти та рекомендації. Граф соціальної мережі повинен бути динамічним для моделювання поведінки користувачів, появи нового контенту та процесу створення і пропонування рекомендацій, а також для моделювання та відслідковування змін у мережі після надання рекомендацій користувачам.

У даній роботі було створено модель соціальної мережі з рекомендаційною системою [53, 161] за допомогою графової бази даних Neo4j [66] та мови програмування Python.

Соціальна мережа була представлена у вигляді стохастичного графу, у

якого у якості вершин були:

– користувачі соціальної мережі;

– пости (інформаційні блоки) користувачів у соціальній мережі.

У якості ребер були відношення: "друзі"; "підписники"; "опублікування посту"; "пост переглянуто"; "лайк посту"; "подоба між користувачами"; "подоба між постами" та "пост рекомендовано".

Зовнішній вигляд зрізу графу розробленої моделі соціальної мережі з рекомендаційною системою представлено на рис. 4.3.

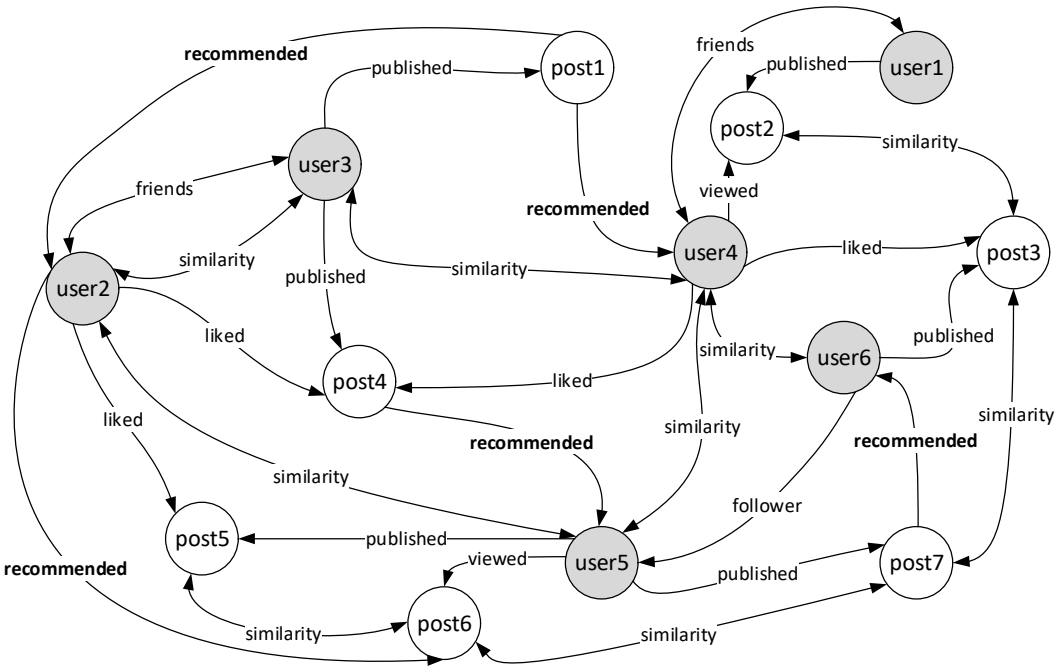


Рис. 4.3. Приклад зрізу графу запропонованої моделі соціальної мережі з рекомендаційною системою

Генерація структури зв’язків соціальної мережі здійснювалася на основі модифікованої моделі Барабаши-Альберт. Для генерації мережі її граф був поділений на наступні підграфи: Users-Friends, Users-Followers, Users-Similarity, Posts-Published, Posts-Viewed, Posts-Liked (або Posts-Rated), Posts-Similarity та Posts-Recommended.

Підграфи Users-Friends, Users-Followers, Posts-Published, Posts-Viewed

та Posts-Liked (або Posts-Rated) створюються генератором графу соціальної мережі.

Підграфи Users-Similarity, Posts-Similarity та Posts-Recommended створюються рекомендаційною системою у процесі формування рекомендацій.

Етапи запропонованого методу генерації структури зв'язків соціальної мережі з рекомендаційною системою:

1 етап. Генерується неорієнтований підграф Users-Friends на основі моделі Барабаши-Альберт.

2 етап. Генерується орієнтований підграф Users-Followers на основі моделі Барабаши-Альберт.

3 етап. Підграфи Users-Friends та Users-Followers об'єднуються у загальний граф.

4 етап. Генерується орієнтований підграф Posts-Published на основі модифікованої моделі Барабаши-Альберт. На першій ітерації випадковим чином обираються n_0 користувачів, які "створюють" m_0 постів. Потім на кожній новій ітерації додається новий пост, ймовірність його опублікування деяким користувачем залежить від кількості друзів та підписників у цього користувача та кількості уже опублікованих постів, і визначається за формулою:

$$P_i = \frac{k_{1i} + k_{2i} + k_{3i}}{\sum_j (k_{1j} + k_{2j} + k_{3j})} \ , \qquad (4.13)$$

де k_{1i} – кількість друзів у i-го вузла, k_{2i} – кількість підписників у i-го вузла, k_{3i} – кількість постів у i-го вузла, а в знаменнику підраховується сума всіх цих значень для усіх існуючих у мережі вузлів.

Для кожного посту генерується набір ключових слів (або властивостей), які вибираються з заданого набору, для подальшої можливості моделювати роботу рекомендаційної системи.

5 етап. Підграф Posts-Published приєднується до загального графу.

6 етап. Генерується орієнтований підграф Posts-Viewed на основі модифікованої моделі Барабаши-Альберт. На першій ітерації випадковим чином обираються n_0 користувачів, які "переглядають" m_0 випадковим чином обраних постів. Потім на кожній новій ітерації додається новий перегляд випадкового посту, ймовірність того, що деякий пост буде переглянутий, залежить від кількості друзів та підписників у автора посту та кількості уже опублікованих ним постів, а також від кількості попередніх переглядів даного поста, та визначається за формулою:

$$P_i = \frac{q_{1i} + q_{2i} + q_{3i}}{\sum_j (q_{1j} + q_{2j} + q_{3j})}, \quad (4.14)$$

де q_{1i} – кількість друзів у автора i-го поста, q_{2i} – кількість підписників у автора i-го поста, q_{3i} – кількість переглядів у i-го поста, а в знаменнику підраховується сума всіх цих значень для усіх існуючих у мережі вузлів.

7 етап. Підграф Posts-Viewed приєднується до загального графу.

8 етап. Генерується підграф Posts-Liked (або Posts-Rated) на основі модифікованої моделі Барабаши-Альберт. На першій ітерації випадковим чином обираються n_0 користувачів, які "ставлять" m_0 лайків (або оцінок) випадковим постам. Потім на кожній новій ітерації додається новий лайк випадковому посту, ймовірність того, що деякий пост отримає лайк, залежить від кількості друзів та підписників у автора посту та кількості уже опублікованих ним постів, а також від кількості переглядів даного поста та кількості попередніх лайків даного поста, і визначається за формулою:

$$P_i = \frac{q_{1i} + q_{2i} + q_{3i} + q_{4i}}{\sum_j (q_{1j} + q_{2j} + q_{3j} + q_{4j})}, \quad (4.15)$$

де q_{1i} – кількість друзів у автора i-го поста, q_{2i} – кількість підписників у автора i-го поста, q_{3i} – кількість переглядів у i-го поста, q_{4i} – кількість лайків (або оцінок) у i-го поста, а в знаменнику підраховується сума всіх цих значень для усіх існуючих у мережі вузлів.

9 етап. Підграф Posts-Liked (або Posts-Rated) приєднується до загального

графу.

10 етап. Підграфи Users-Similarity, Posts-Similarity та Posts-Recommended генеруютья алгоритмами обраної рекомендаційної системи та приєднуються до загального графу.

Розроблена модель соціальної мережі з рекомендаційною системою була протестована для перевірки подоби властивостей структури її зв'язків до структури реальних соціальних мереж.

Приклад частини соціальної мережі, одержаної в результаті моделювання зображено на рис. 4.4.

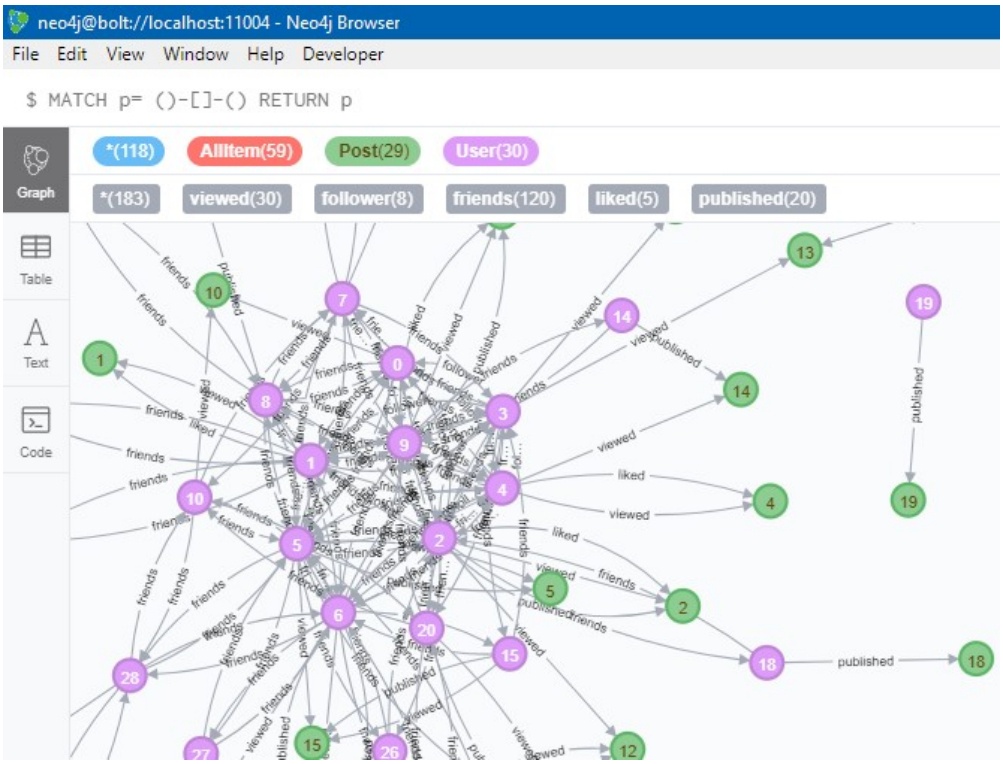


Рис. 4.4. Приклад частини соціальної мережі одержаної в результаті моделювання (скриншот з менеджера СУБД Neo4j Desktop)

З метою тестування розробленої системи згенеровані графи соціальної мережі були експортовані у .csv файли з бази даних Neo4j, а їх параметри досліджені у програмному забезпеченні Gephi [97] – інтерактивній платформі з відкритим кодом для аналізу та візуалізації даних представлених у вигляді

графів, що дозволяє досліджувати всі види мереж, складних систем, статичних та динамічніх графів.

Середні значення параметрів згенерованих у запропонованій програмній імітаційній моделі соціальних графів рекомендаційної системи, які були обчислені за допомогою Gephi:

– Середній степінь вузлів: 5.7.

– Діаметр мережі: 4.0.

– Щільність графу: 0.22.

– Середній коефіцієнт кластеризації: 0.61.

– Середня довжина шляху: 1.98.

Отже, при генерації структури зв’язків соціальних мереж розробленим методом було отримано графи, що були сильно розріджені (мали низьку щільність), діаметр мережі був у середньому рівним 4 (що відповідає сучасним віртуальним соціальним мережам, наприклад, у мережі Facebook цей показник 4.74), коефіцієнт кластеризації був досить високим, візуально на графі спостерігалися різні кластерні утворення (кліки та клани), середня довжина шляху була невисока. Все це відповідає параметрам реальних соціальних мереж.

Запропонований метод також було використано як основу для моделювання структури зв’язків користувачів та об’єктів також і контент-орієнтованого веб-сайту.

Також було здійснено моделювання поведінки користувачів на основі генерації динамічного графу Users-Ratings-Items. Модель розроблена таким чином, щоб одержувати набір даних схожий на MovieLens datasets [30].

Етапи запропонованого методу моделювання структури зв’язків між елементами та поведінки користувачів рекомендаційної системи контент-орієнтованого веб-сайту:

1 етап. Ініціалізація параметрів системи, зокрема, вибір кількості користувачів та об’єктів, кількості можливих властивостей у них, проценту активних користувачів, проценту популярних об’єктів, кількості часових

ітерацій, після яких модель завершить свою роботу тощо. При необхідності моделювати інформаційну атаку обирається тип атаки, кількість ботів та кількість цілей атаки. Відбувається створення набору можливих властивостей для елементів системи, реалізованих у моделі прихованими факторами, та генерація шаблонів кластерів елементів на основі цих властивостей.

2 етап. Створення «Зерна» соціального графу рекомендаційної системи – до графу додається початкова кількість користувачів та об'єктів, деякій кількості об'єктів виставляються оцінки. Початкова кількість користувачів, початкова кількість об'єктів та щільність графу – налаштовувані параметри. Кожному новому користувачу та об'єкту системи привласнюється значення зміщення, певний кластер та відповідний йому вектор значень прихованих факторів, що визначає ступінь його приналежності до кожної з можливих властивостей для елементів системи. Оцінки у системі виставляються на основі ступеня співпадіння прихованих факторів користувача та об'єкта, а також показників їх зміщень. Ймовірність перегляду об'єкту всередині «Зерна» залежить від заданої щільності графу. Ймовірність виставлення оцінки переглянутому об'єкту для аутентичних користувачів залежить від прихованих факторів користувача (що визначають його вподобання), прихованих факторів об'єкту (що визначають його властивості), зміщення користувача (яке вказує на характерне для користувача систематичне заниження або завищення оцінок), зміщення об'єкту (яке вказує на якість об'єкту, що викликає завжди одержання оцінок вище/нижче, ніж у схожих за властивостями об'єктів), а також випадкового зміщення (яке виникає з ймовірністю 0-3% і є налаштовуваним параметром). У «Зерні» профілі ботів відсутні, вони починають приєднуватися до мережі на 3 етапі.

3 етап. На кожній ітерації часу моделі до графу приєднується певна кількість користувачів та фільмів. Ця кількість визначається випадковим чином та лежить у межах від 0 до N, де N менше загальної кількості елементів системи відповідного типу. Також на кожній ітерації часу моделі відбувається вибір деякої кількості пар користувачів та об'єктів для

виставлення оцінок. Ймовірність перегляду об’єкту може визначатися на основі принципу «бажаного приєднання» (4.12) з моделі Барабаши-Альберт або його модифікацій побудованих за принципом рівнянь (4.13)-(4.15). Ймовірність виставлення оцінки переглянутому об’єкту для аутентичних користувачів залежить від прихованих факторів користувача (що визначають його вподобання), прихованих факторів об’єкту (що визначають його властивості), зміщення користувача (яке вказує на характерне для користувача систематичне заниження або завищення оцінок), зміщення об’єкту (яке вказує на якість об’єкту, що викликає завжди одержання оцінок вище/нижче, ніж у схожих за властивостями об’єктів), а також випадкового зміщення (яке виникає з ймовірністю 0-3% і є налаштовуваним параметром). Ймовірність перегляду об’єкту та виставлення йому оцінки для ботів визначається використаною для їх створення моделлю атаки.

4 етап. Зупинка роботи імітаційної моделі, збереження згенерованого набору даних у файл для подальшого використання у методах формування та тестування списків рекомендацій.

Структурна схема розробленої програмної моделі зображена на рис. 4.5. З рисунку видно, що програмна імітаційна модель складається з наступних елементів: «Модель користувача системи», «Модель об’єкту системи» та «Модель інформаційних процесів у рекомендаційній системі».

Модель користувача системи має наступні параметри:

– Id – ідентифікаційний номер користувача, містить цифру;

– Активний – приймає значення True – якщо користувач активний (ставить більше оцінок, ніж інші) та False – якщо рівень активності звичайний;

– Бот – приймає значення True – якщо користувач бот (профіль приймає участь в атаці на рекомендаційну систему) та False – в протилежному випадку;

– Тип – приймає значення «Звичайний», якщо користувач не бот, інакше містить назву застосованої на рекомендаційну систему атаки;

Рис. 4.5. Структурна схема програмної імітаційної моделі користувачів та об’єктів рекомендаційної системи

– Характер – містить тип характеру користувача стосовно стилю виставлення оцінок: «Звичайний», «Частіше ставить хороші оцінки», «Ставить тільки хороші оцінки». «Звичайний» ставить і позитивні, і негативні оцінки. Інші типи користувачів можуть не виставляти оцінку, якщо оцінили об’єкт негативно;

– Час реєстрації – містить час реєстрації користувача у системі;

– Час останньої активності – містить час останньої активності користувача;

– Час зупинки активності – час, коли користувач перестав користуватися системою;

– Приховані фактори – список випадкових змінних, які приймають значення від -1 до 1, довжиною K, що моделює вплив тих чи інших характеристик фільму на судження про нього користувача;

– Зсув користувача – випадкова величина в діапазоні від -1.0 до 1.0, що моделює схильність користувача занижувати чи завищувати оцінки фільмам;

– Час останньої зміни вподобань – використовується при необхідності моделювання змін вподобань користувачів у часі, містить час останнього перезапису списку прихованих факторів користувача.

У розробленому програмному забезпеченні об’єктами системи були фільми. Модель об’єкту системи має наступні параметри:

– Id – ідентифікаційний номер об’єкту, містить цифру;

– Популярний – приймає значення True – якщо об’єкт популярний (отримує більше оцінок, ніж інші) та False – якщо рівень популярності звичайний;

– Тип для ботів – приймає значення «Цільовий» – якщо боти повинні змінити рейтинг об’єкту в певну сторону та False – в протилежному випадку;

– Тип атаки – приймає значення «Немає», якщо об’єкт не цільовий, або вказує тип атаки «На зменшення рейтингу», «На збільшення рейтингу»;

– Час додавання – містить час додавання об’єкту до бази даних рекомендаційної системи;

– Приховані фактори – список випадкових змінних, які приймають значення від -1 до 1, довжиною *K*, що моделює вираженість тих чи інших характеристик у даному об’єкту, які впливають на рівень інтересу до нього користувача;

– Зсув об’єкту – випадкова величина в діапазоні від -1.0 до 1.0, що моделює загальну якість об’єкту, що впливає на оцінки користувачів і призводить до отримання частіше низьких оцінок через низьку якість або частіше високих оцінок через високу якість об’єкту.

Параметри користувачів та об’єктів системи встановлюються під час створення та ініціалізації відповідних екземплярів об’єктів, а деякі з них можуть змінюватися під час роботи програми.

Модель інформаційних процесів у системі має наступні параметри:

– Матриця рейтингів – містить матрицю суміжності графу, у якого вершинами являються користувачі та об’єкти, а ребрами оцінки поставлені користувачами об’єктам.

– Матриця часу – містить матрицю суміжності графу, у якого вершинами являються користувачі та об’єкти, а ребрами час виставлення оцінок поставлених користувачами об’єктам.

– Набір можливих оцінок – містить набір оцінок, які користувачі можуть виставляти об’єктам, в розроблюваній моделі набір оцінок представлений наступним списком [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0], що означає можливість поставити оцінку у вигляді кількості зірочок, зірочок максимум 5, можна вибирати половинку зірочки.

– Список користувачів – містить список користувачів системи, екземплярів класу Користувач.

– Список об’єктів – містить список об’єктів системи, екземплярів класу Об’єкт.

– Початковий час роботи моделі – містить час у форматі Unix time stamp.

– Поточний час роботи моделі – містить час у форматі Unix time stamp, що вказує на час останньої дії у системі.

– Список ботів – містить список ботів у розроблюваній моделі.

– Список цілей – містить список цілей атаки ботів у розроблюваній моделі.

– Кількість кластерів – містить задану кількість кластерів, на яку можна буде розділити елементи системи на основі даних про їх приховані фактори.

– Шаблони кластерів – містить список шаблонів для моделювання прихованих факторів користувачів/об’єктів. Всього у моделі було створено 19 випадкових шаблонів для генерації елементів, що можуть належати 19 різним кластерам. Також в моделі можна обрати опцію моделювання певного проценту елементів, що не належать до згенерованих кластерів.

Було реалізовано наступні функції для моделювання поведінки користувачів і об’єктів та інформаційних процесів рекомендаційної системи:

– Запуск моделі – запускає процес моделювання користувачів та об’єктів рекомендаційної системи.

– Збереження даних моделі у файл – зберігає усі дані та параметри розробленої програмної імітаційної моделі користувачів, об’єктів та процесів рекомендаційної системи у файл.

– Генерація елементів системи – створення заданої кількості користувачів та об’єктів системи та привласнення їм параметрів.

– Генерація шаблонів кластерів елементів – дозволяє згенерувати шаблони прихованих факторів елементів для створення в подальшому набору елементів, що відносяться до певних кластерів.

– Генерація прихованих факторів елементів на основі шаблонів кластерів – на вході одержує шаблон кластера, на виході надає список прихованих факторів, що випадковим чином на деякі величини відрізняються від даних у шаблоні, таким чином, щоб не виходити за межі кластеру, але мати свої унікальні значення факторів.

– Генерація прихованих факторів елементів за межами кластерів – генерація списку прихованих факторів елемента випадковим чином без використання шаблонів.

– Генерація «Зерна» соціального графу – створює початковий соціальний граф рекомендаційної системи на основі заданої початкової кількості користувачів, об’єктів та щільності графу.

– Ітеративна побудова моделі – дозволяє моделювати зміну часу.

– Одна ітерація моделі – всередині даної функції викликаються всі функції, що реалізують поведінку користувачів та роботу рекомендаційної системи в поточний момент часу.

– Генерація поточного часу роботи моделі – генерація часового проміжку між двома подіями у системі – випадкової величини, що лежить в заданому діапазоні.

– Додавання користувача до системи – моделювання процесу реєстрації користувача у рекомендаційній системі, користувач отримує час реєстрації та можливість переглядати об’єкти та ставити їм оцінки.

– Додавання об’єкту до системи – моделювання процесу додавання об’єкту до бази даних рекомендаційної системи, об’єкт одержує час додавання до бази даних та можливість одержувати перегляди та оцінки.

– Визначення ймовірності перегляду об'єкту – на основі принципу «Бажаного приєднання» визначається ймовірність перегляду певного об’єкту певним користувачем.

– Визначення ймовірності появи оцінки – на основі принципу «Бажаного приєднання» визначається ймовірність виставлення оцінки певному об’єкту певним користувачем.

– Генерація оцінки на основі прихованих факторів відповідного об’єкту та користувача. Оцінка для пари користувач-об’єкт визначається за наступними формулами:

$$d_{u,m} = \frac{\sum_{i=0}^{n} \left| f_{u,i} - f_{m,i} \right|}{n}, \tag{4.16}$$

$$r_{u,m} = \Psi\left(5 d_{u,m} + b_u + b_m\right), \tag{4.17}$$

де $d_{u,m}$ – дистанція між користувачем u та об'єктом m у багатомірному просторі прихованих факторів, може приймати значення від 0 до 1; n – кількість прихованих факторів у системі; $f_{u,i}$ – i-тий прихований фактор користувача u; $f_{m,i}$ – i-тий прихований фактор об'єкту m; b_u – зсув користувача в оцінюванні об'єктів (рівень вимогливості до контенту); b_m – зсув об'єкту у одержанні оцінок (рівень якості контенту); Ψ() – функція, що перетворює одержане дробове число у дискретне число з набору оцінок [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0], наприклад, якщо число лежить в діапазоні від (4.000 – k) до (4.500 – k), де k деяке невелике число (у моделі було взято k = 0.05), то воно перетворюється на оцінку 4.0.

– Рішення поставити оцінку та її коригування. Після того як користувач «переглянув» об'єкт, і на основі прихованих факторів, було визначено, яка оцінка для даного об'єкту відповідає його вподобанням, використовується дана функція, що визначає факт того, що користувач прийме рішення поставити переглянутому об'єкту оцінку. А також у даній функції можливе незначне коригування оцінки на основі випадкових чинників. За допомогою даної функції ймовірності появи оцінок можна наблизити до частот появи оцінок у наборі даних MovieLens datasets, наведених у таблиці 4.1.

Таблиця 4.1. Частота появи різних оцінок у відкритому наборі даних MovieLens datasets

Оцінка	**Частота появи**
0.5	0.015558
1.0	0.032405
1.5	0.015509
2.0	0.067723
2.5	0.048238
3.0	0.201993
3.5	0.119742
4.0	0.268933
4.5	0.083401
5.0	0.146498

Також в моделі розроблені генератори оцінок ботів для різних видів атак.

– Генерація оцінки для випадкової атаки – створює оцінки для пар бот-об’єкт, де бот здійснює випадкову атаку на рекомендаційну мережу. Генерація оцінки для випадкової атаки на підвищення рейтингу відбувається відбувається наступним чином:

$$r_{u,m} = \begin{cases} \text{randomPattern}(), \textit{якщо об'єкт} - \text{"}\textit{звичайний}\text{"} \\ 5.0, \textit{якщо об'єкт} - \text{"}\textit{цільовий}\text{"} \end{cases}, \quad (4.18)$$

де randomPattern() – функція, що генерує випадкові оцінки з заданими значеннями ймовірності появи.

– Генерація оцінки для середньої атаки – створює оцінки для пар бот-об’єкт, де бот здійснює середню атаку на рекомендаційну мережу. Генерація оцінки для середньої атаки на підвищення рейтингу відбувається наступним чином:

$$r_{u,m} = \begin{cases} \textit{average}\,\text{Pattern}(), \textit{якщо об'єкт} - \text{"}\textit{звичайний}\text{"} \\ 5.0, \textit{якщо об'єкт} - \text{"}\textit{цільовий}\text{"} \end{cases}, \quad (4.19)$$

де $\textit{average}\,\text{Pattern}()$ – функція, що генерує для випадково обраного об’єкту його середньостатистичну оцінку.

– Генерація оцінки для популярної атаки – створює оцінки для пар бот-об’єкт, де бот здійснює популярну атаку на рекомендаційну мережу. Генерація оцінки для популярної атаки на підвищення рейтингу відбувається відбувається наступним чином:

$$r_{u,m} = \begin{cases} \textit{popular}\,\text{Pattern}(), \textit{якщо об'єкт} - \text{"}\textit{звичайний}\text{"} \\ 5.0, \textit{якщо об'єкт} - \text{"}\textit{цільовий}\text{"} \end{cases}, \quad (4.20)$$

де $\textit{popular}\,\text{Pattern}()$ – функція, що генерує для випадково обраного об’єкту з множини популярних об’єктів його середньостатистичну оцінку.

– Визначення ймовірності зміни вподобань – використовується при необхідності моделювання змін вподобань користувачів системи у часі, визначає ймовірність того, що у поточний момент часу вказаний користувач

змінить свої вподобання, якщо відомий час, коли він останній раз змінював вподобання.

– Зміна вподобань користувача – здійснює заміну прихованих факторів вказаного користувача.

Тож, на основі запропонованого методу моделювання структури зв'язків між елементами та поведінки користувачів рекомендаційної системи контент-орієнтованого веб-сайту, було розроблено програмну імітаційну модель рекомендаційної системи.

Програмна імітаційна модель була розроблена на мові програмування Python з використанням принципів об'єктно-орієнтованого програмування. Усі дані програмної імітаційної моделі записувалися до графової СУБД Neo4j, а також по завершенню роботи моделі зберігалися у файл власного формату.

Соціальний граф рекомендаційної системи у програмній імітаційній моделі мав наступний формат: вершини – користувачі та об'єкти, ребра – зв'язки типу «оцінив», «подоба», «рекомендовано» тощо. І вершини, і ребра містять набори параметрів, що відповідають наявній про них інформації. Наприклад, ребро «оцінив» містить значення оцінки і часову мітку, а вершина типу користувач містить усі поля даних, що відповідають параметрам моделі користувача у системі.

На рис. 4.6-4.8 наведено приклади частин (зрізів соціального графу) бази даних програмної імітаційної моделі рекомендаційної системи.

На рис. 4.6 наведено виконання наступного запиту до СУБД Neo4j Desktop:

```
match p=(u1:User)-[r1:Rated{goal: "testing"}]->(m1:Movie)
return p LIMIT 20
```

Цей запит виводить на екран зріз графу з вершинами представленими користувачами і фільмами та ребрами представленими оцінками, віднесеними за часом виставлення до тестового набору даних. У запиті встановлено ліміт виведення на екран у кількості 20 вершин графу.

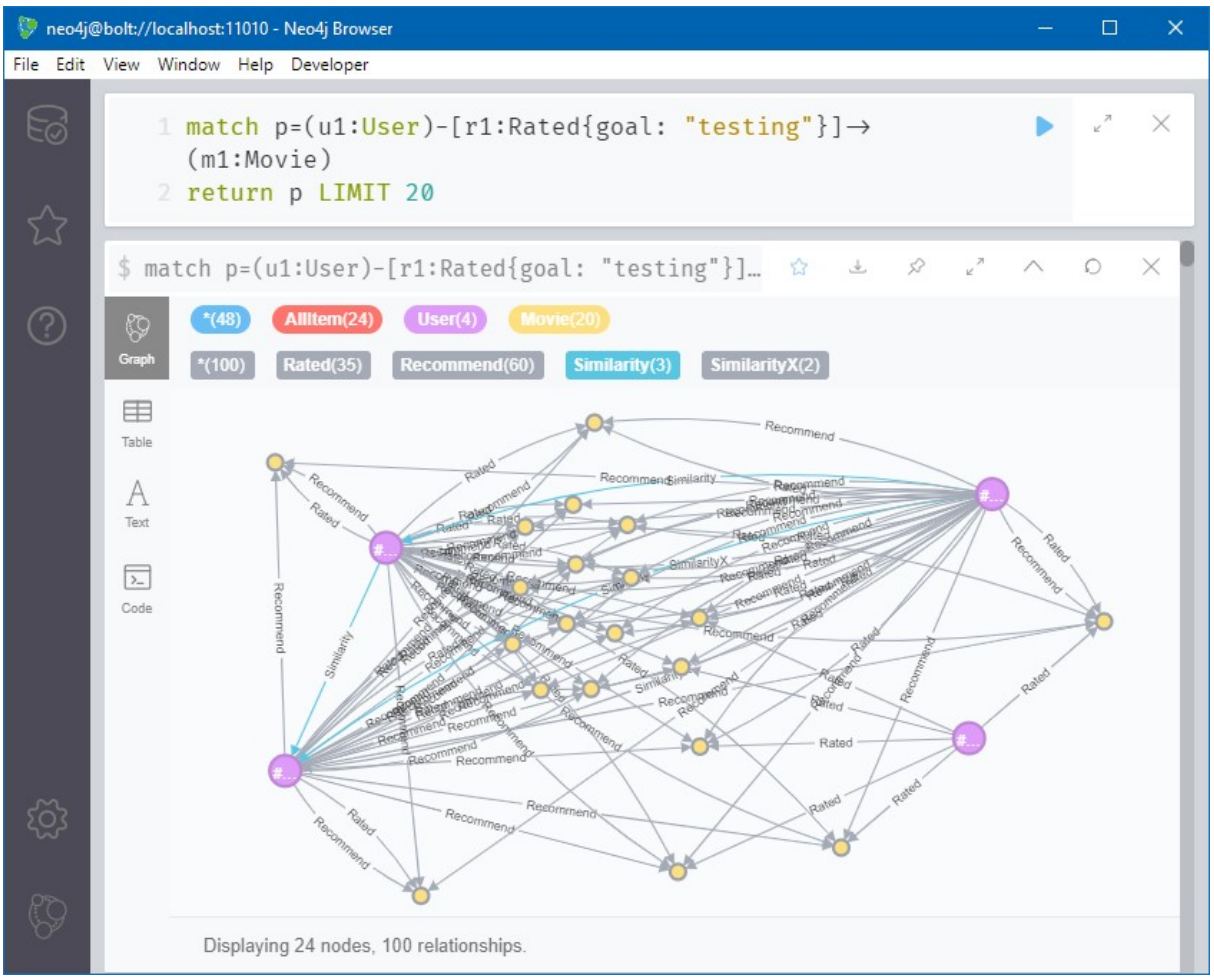


Рис. 4.6. Приклад частини бази даних запропонованої програмної імітаційної моделі рекомендаційної системи №1 (скриншот з менеджера СУБД Neo4j Desktop)

На рис. 4.6 видно частину даних з записами 4 користувачів, що оцінили деяку множину фільмів. Для відображення даних по запиту СУБД Neo4j обрала випадковим чином 24 вершини графу серед користувачів та фільмів. Також були відображені ребра :Rated та :Recommend, що містять поставлені користувачами оцінки у тестовій вибірці та рекомендації спрогнозовані рекомендаційною системою на основі робочої вибірки даних.

На рис. 4.7 наведено виконання наступного запиту до СУБД Neo4j Desktop:

```
match p=(u1:User)-[r1:Rated{goal: "testing"}]-(m1:Movie)-[r2:Recommend]-(u1:User)
return r1.rating, r2.Recommend
```

Цей запит виводить на екран таблицю з порівнянням прогнозованих та реальних оцінок користувачів у розробленій програмній імітаційній моделі.

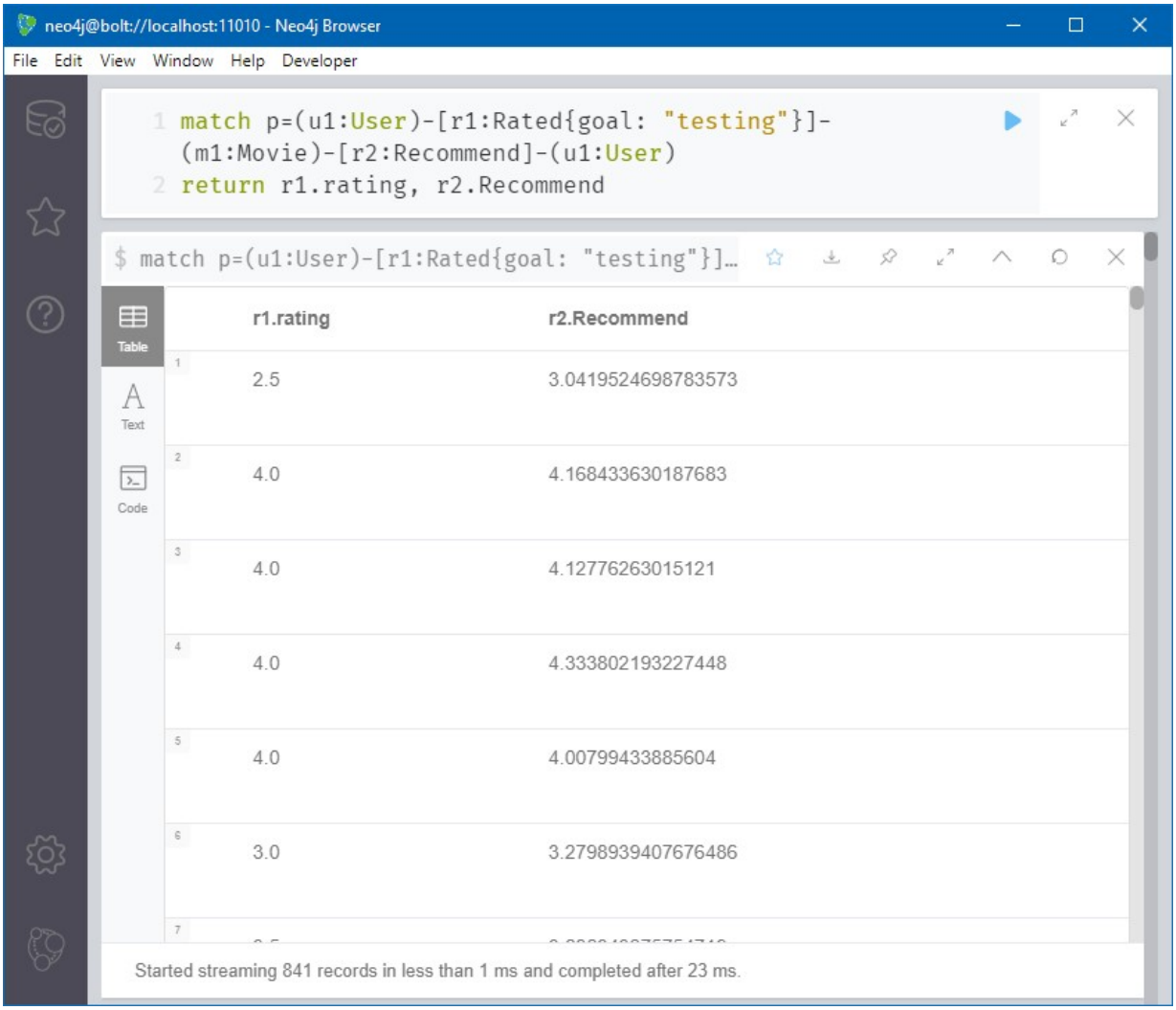


Рис. 4.7. Приклад частини бази даних запропонованої програмної імітаційної моделі рекомендаційної системи №2 (скриншот з менеджера СУБД Neo4j Desktop)

На рис. 4.8 наведено виконання наступного запиту до СУБД Neo4j Desktop:

```
match p=(u1:User{Bot:1})-[r:Similarity]-(u2:User{Bot:1})
where r.SimilarityCoefficient > 0.6
return p
```

Цей запит виводить на екран згенерованих у програмній імітаційній моделі ботів, коефіцієнти подоби яких між собою більші, ніж 0.6.

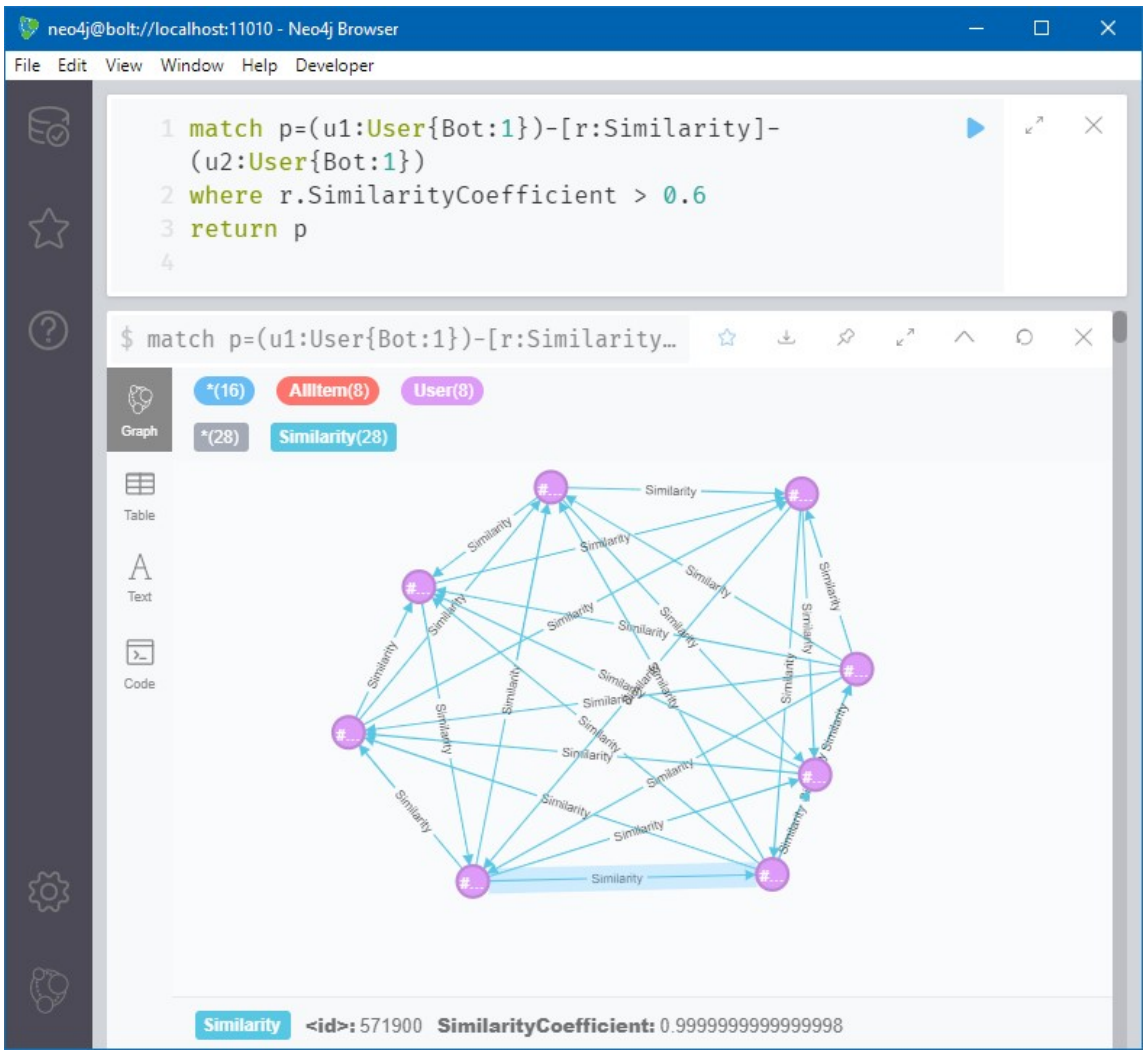


Рис. 4.8. Приклад частини бази даних запропонованої програмної імітаційної моделі рекомендаційної системи №3 (скриншот з менеджера СУБД Neo4j Desktop)

Було проведено серію експериментів для перевірки працездатності розробленої програмної імітаційної моделі. У ході експериментів було згенеровано множину наборів даних. Кожен набір даних було розділено на робочу та тестову вибірку. На основі робочої вибірки здійснювалося прогнозування вподобань користувачів методом колаборативної фільтрації заснованої на моделі сусідства. На основі тестової вибірки перевірялася точність прогнозування вподобань. В ході проведення експериментів перевірялося наступне твердження – якщо розроблена програмна імітаційна модель працездатна, вподобання користувачів можна спрогнозувати, а отже,

точність їх прогнозування буде у більшості випадків вищою 50% – випадкового угадування. А якщо модель непрацездатна, точність вподобань буде в середньому дорівнювати 50% – випадковому угадуванню.

Результати проведених експериментів наведено у табл. 4.2.

Таблиця 4.2. Результати прогнозування вподобань користувачів для наборів даних, згенерованих у запропонованій програмній імітаційній моделі рекомендаційної системи

№ експ.	Кількість користувачів, що поставили оцінку, на момент завершення роботи моделі	Кількість оцінених об'єктів на момент завершення роботи моделі	Точність (Precision) прогнозування вподобань	Повнота (Recall) прогнозування вподобань	RMSE розпізнавання вподобань
1.	30	1737	0.7719292440211308	0.81034497839453300	1.0543084707727113
2.	30	1320	0.7693525717102206	0.85070819206974820	1.2024823221108385
3.	30	825	0.8781937144980624	0.74982600732600730	1.1784218012548980
4.	30	1265	0.7055425381478558	0.70130991574624740	1.1535842351508123
5.	30	1669	0.7081726354453627	0.48944559025204190	1.2525459159734247
6.	30	958	0.6700892857142858	0.82327590534421650	1.1563594372355381
7.	30	958	0.6700892857142858	0.82327590534421650	1.1563594372355381
8.	30	2647	0.6494785868255742	0.47179061322736494	1.3241637740428220
9.	30	1024	0.7231150793650794	0.66049382716049380	1.1018879806992712
10.	30	1272	0.8673120238165188	0.63188541844455820	1.2627171601468419
11.	30	2317	0.7462400388259947	0.72796448930824710	1.1163801390679262
12.	30	1760	0.7961536326920942	0.75995420461483820	1.1459373792948027
13.	30	2004	0.7184573002754821	0.75056579079834900	1.1035103155092110
14.	30	2182	0.8138317861894723	0.58607290729828270	1.0750869260761295
15.	30	2102	0.7228903043681369	0.54988968641206355	1.1892137948318726
16.	30	1696	0.7599456800017810	0.75416257497279740	1.1886189880656495
17.	30	1644	0.7491315678181803	0.78056544345655740	1.3398043974843240
18.	30	2362	0.7234745877322469	0.49111816546027070	1.3663005116857696
19.	30	1076	0.7998294976705490	0.76530531090084440	1.2574914173017848
20.	30	2037	0.6592393411053430	0.63676462672561870	1.1782081826062742
21.	30	755	0.7408821198294883	0.66660555131143350	1.0870096354084289
22.	30	1772	0.8039272030651341	0.54432210121607830	1.1678233575543164
23.	30	1472	0.8292063052124028	0.83682796944701020	1.1602096935207790
24.	30	3306	0.7611304898164580	0.82155367497914270	1.0674856942125281
25.	30	2003	0.7355188197686138	0.73551881976861380	1.0998788461432647
26.	30	1364	0.6345914044027252	0.60017835458409220	1.1185243772901814
27.	30	2904	0.6543126582434176	0.76058052326761510	1.2549791505683001
28.	30	1478	0.7200315602726313	0.60939598997493730	1.2949416002271243
29.	30	1691	0.7236795919354060	0.61797535797535794	1.0683220460139589
30.	30	1661	0.6029202279202278	0.68546821305841930	1.2808915371227032
31.	30	1833	0.8163724334807294	0.71495427810093800	1.1665772049713357
32.	30	1845	0.7909850275172856	0.82016710717499880	0.9796258848538932
33.	30	1227	0.5854447250280584	0.62100425436632330	1.2338610540509551
34.	30	1143	0.7917517814815983	0.84853884344393220	1.3705954145997710
35.	30	1626	0.7718753873004287	0.51078322566047010	1.2850965914591896
36.	30	1800	0.7870922259012808	0.68851761485372480	1.0940759694315463
37.	30	2182	0.7732028317379852	0.76897289974976270	1.1415323425157300
38.	30	2305	0.7179638882649061	0.53469737374796715	1.1941778101815683
39.	30	1478	0.7981777173009934	0.85508073770984580	1.0325882209697739
40.	30	1191	0.7492266791774180	0.66912496258471480	1.2327120883504040
41.	30	2522	0.8355522897372046	0.80943594488147920	1.1384273018745728
42.	30	1619	0.6712910981156595	0.81184715821812600	1.2930302575511432
43.	30	1411	0.7858131624742920	0.62270360008090066	1.2547954480582795
44.	30	1025	0.7626872163757118	0.77722739820565890	1.3332960636567992
45.	30	2010	0.7460056533720757	0.67858866422450730	1.2057441226372030

46.	30	2044	0.7862353148246565	0.86916264206917930	1.1966181229724382
47.	30	1178	0.7870395354681692	0.60132719122023930	1.2131533312576699
48.	30	1234	0.7376982565601492	0.72968020541549950	1.2903361281564805
49.	30	1499	0.7246752032205798	0.66434013633739080	1.2634134283925504
50.	30	1315	0.7561757499140482	0.69055925422024470	1.2395841550268318
51.	30	1055	0.9093861863403642	0.80637051108753450	1.1281565620475476
52.	30	1981	0.6190629518442113	0.69421929964489750	1.2137375702475484
53.	30	1571	0.7433831482947375	0.69062631057378910	1.2779420051175236
54.	30	1171	0.7245861269489319	0.67714199028537250	1.3299468688755590
55.	30	2718	0.6237410096254360	0.61422784326255240	1.1421946430256604
56.	30	1440	0.6987111379766762	0.87698390506717570	1.2168354318456930
57.	30	1216	0.6498627860235003	0.60206793702297360	1.2351014884482580
58.	30	1597	0.6795146316102199	0.75194285119914410	1.1530565351876767
59.	30	1487	0.9071294835158878	0.64420396302701360	1.2156848894777983
60.	30	1591	0.6539017343365170	0.64272271123017390	1.0938211810811211
61.	30	2585	0.8016425409828118	0.88783379133398700	1.0505374622197290
62.	30	1850	0.6785959477289335	0.70763610878090280	1.2707557435854904
63.	30	1403	0.8363161692109061	0.61885430719396574	1.1946120255910464
64.	30	2628	0.7206840161924396	0.69152314221367420	1.2834066176860042
65.	30	950	0.8388651988199500	0.78522714753917770	1.0367224084124684
66.	30	1314	0.7205205964066724	0.83090983449371680	1.1111735804309664
67.	30	1656	0.6907811191849814	0.86319241335102790	1.1449414819164996
68.	30	1328	0.7103995934184614	0.93835978835978850	1.2305567281983627
69.	30	1120	0.8597939182030092	0.73394705271850090	1.2016927146680807
70.	30	1149	0.8109783767678503	0.68239712668284100	1.1282655558217929
71.	30	1614	0.7346358492465949	0.88713350421815360	1.0305545533647313
72.	30	1075	0.7676427133830784	0.69653175622033740	1.3374079886998003
73.	30	1778	0.6706772162297769	0.79568744747316170	1.2361072078032570
74.	30	1054	0.7516243441588176	0.87149794649794640	1.2911674053840763
75.	30	957	0.7841430871865654	0.75690364875014480	1.2160626654003002
76.	30	1243	0.6934588071439592	0.79545220545220540	1.3099281276397132
77.	30	1642	0.7062178499846962	0.72839008412174780	1.2788675387699697
78.	30	1496	0.7775218267440780	0.82611184456947920	1.0968595856499228
79.	30	1615	0.8244532887337056	0.71468036466560970	1.1897358686370540
80.	30	1545	0.8301914696241509	0.63316029158725726	1.3091879123458865
81.	30	1711	0.7720842461536473	0.75576482748236960	1.2371890326622104
82.	30	2298	0.5446106751397283	0.72472444423412970	1.4086415643334036
83.	30	1992	0.7274723945165779	0.83418217234972900	1.0790455235855120
84.	30	4613	0.6843610812079855	0.60481691208313244	0.9354551108307452
85.	30	1769	0.7944903151062570	0.79052289839777370	1.3115309923784677
86.	30	1307	0.8848469642099992	0.63292290706605215	1.1056561609143560
87.	30	2707	0.7230429750287689	0.67390478716330110	1.2196873310133570
88.	30	2400	0.8274170694973871	0.64164519595057750	1.3564158816741718
89.	30	1006	0.7194627851140456	0.81998791117897810	1.3771257940234007
90.	30	897	0.7819219995690584	0.67566548358473820	1.2853640099650787
91.	30	1462	0.6480093846035875	0.86011046083816770	1.1365575795068679
92.	30	1106	0.7539191379030950	0.51051104923445350	1.1350213678551269
93.	30	1003	0.7517476558724766	0.79779692923709160	1.1396777908727800
94.	30	1610	0.7393066655566655	0.71733687190593410	1.0965343667900826
95.	30	979	0.6510954616588418	0.79326007326007330	1.1509375048290660
96.	30	2250	0.6361704718317080	0.60574563778095374	1.0706082770333802
97.	30	1187	0.7702431385040082	0.78722170939427930	1.2403869629612654
98.	30	951	0.7627383216199006	0.90999798404210170	1.0666970714051713
99.	30	2707	0.8370880866266613	0.73393959530577670	1.2903658772361128
100.	30	1315	0.6683519314158413	0.79666769946845150	1.1174845283280856
Середні значення:			**0.743807391500000**	**0.720925222800000**	**1.192780925000000**

На рис. 4.9-4.11 наведено графіки зміни випадкових характеристик частоти правильного прогнозування вподобань $P_{точн}^{(i)}$, частоти повного прогнозування вподобань $P_{повн}^{(i)}$ та RMSE розпізнавання вподобань

відповідно. Також на цих рисунках наведено допустимі межі коливання цих випадкових величин, що отримані в результаті проведеного моделювання.

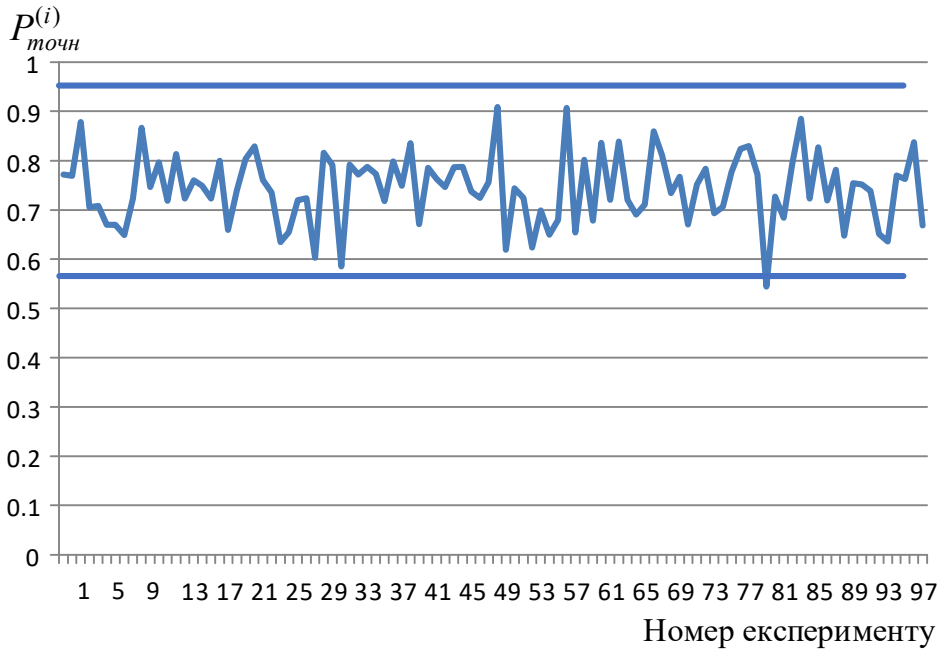


Рис. 4.9. Графік зміни випадкових характеристик частоти правильного прогнозування вподобань $P^{(i)}_{точн}$

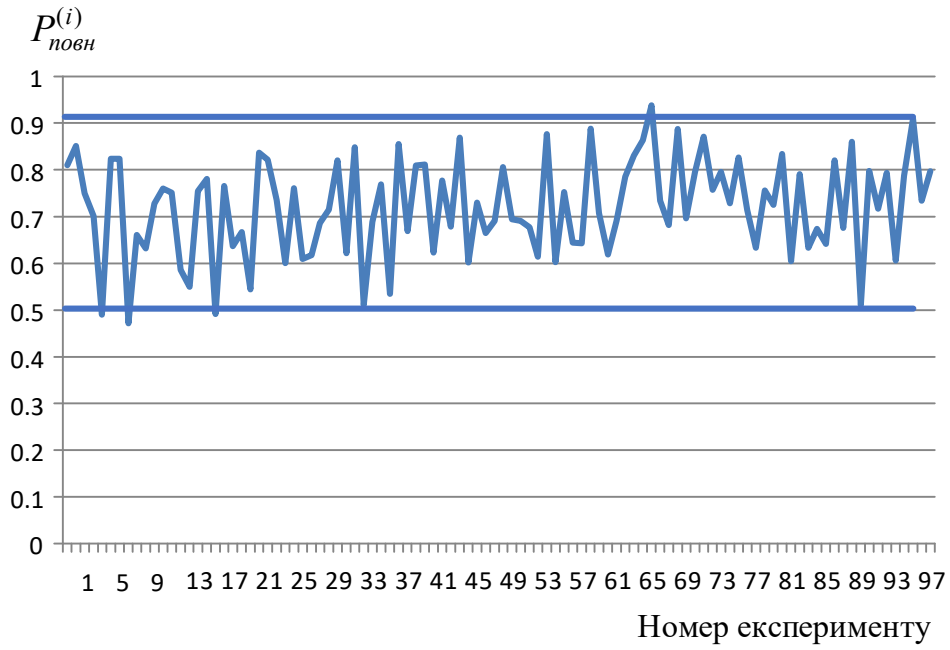


Рис. 4.10. Графік зміни випадкових характеристик частоти правильного прогнозування вподобань $P^{(i)}_{повн}$

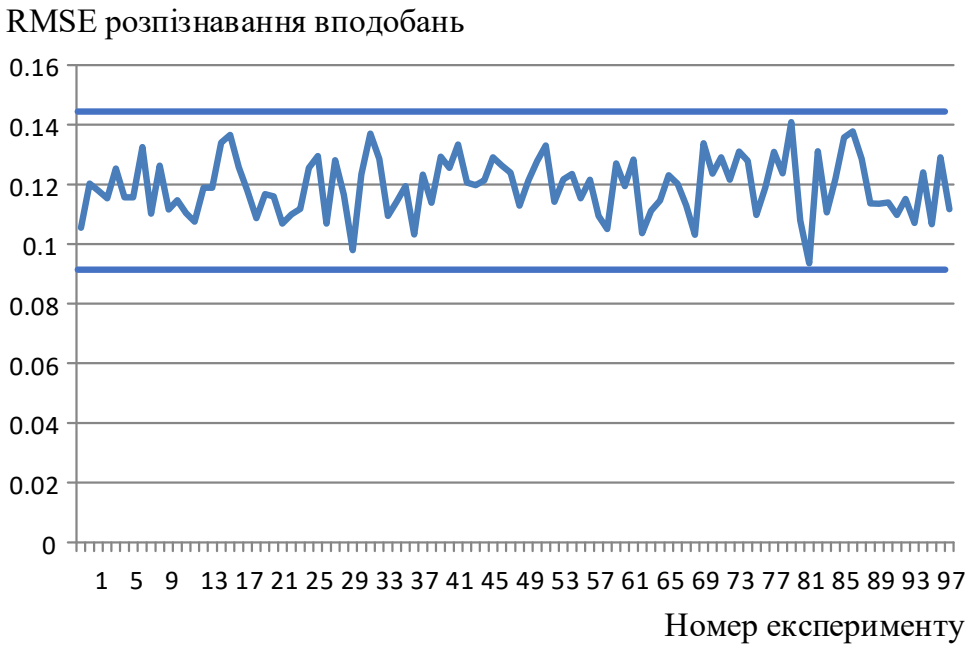


Рис. 4.11. Графік зміни випадкових характеристик RMSE розпізнавання вподобань

Як видно з наведених рисунків, джиттер досліджуваних випадкових характеристик $P_{точн}^{(i)}$, $P_{повн}^{(i)}$ та RMSE розпізнавання вподобань у більшості практичних випадків впевнено вкладається в допустимі межі коливань. Це підтверджує достовірність розробленої програмної імітаційної моделі рекомендаційної системи.

4.2. Розробка математичної моделі зміни у часі вподобань користувачів рекомендаційної системи

Для повноцінного моделювання поведінки звичайних користувачів рекомендаційної системи треба враховувати, що їх вподобання з часом змінюються. Зміни у вподобаннях користувачів можуть бути як періодичними так і неперіодичними. Прикладом періодичних змін можуть бути вподобання до сезонних товарів або вподобання музики на веб-радіо в залежності від часу доби. Для того, щоб зрозуміти яким саме чином

вподобання користувачів змінюються з часом у системах, де відсутня явна періодичність (циклічність) вподобань, таких як, наприклад, рекомендаційна система фільмів, було проведено наступне дослідження з використанням даних MovieLens datasets [30].

В наборі даних MovieLens datasets було здійснено обчислення коефіцієнтів подоби між користувачами для різних часових вікон та досліджено як змінювалися ці коефіцієнти з часом.

Як коефіцієнт подоби між користувачами було використано коефіцієнт кореляції Пірсона [214]:

$$k(u_1,u_2)=\frac{\sum_{i=0}^{n}(r_{1i}-\bar{r}_1)(r_{2i}-\bar{r}_2)}{\sqrt{\sum_{i=0}^{n}(r_{1i}-\bar{r}_1)^2}\sqrt{\sum_{i=0}^{n}(r_{2i}-\bar{r}_2)^2}}, \quad (4.21)$$

де u_1 та u_2 – користувачі, між якими визначається коефіцієнт подоби; r_1, r_2 – оцінки виставлені 1-им та 2-им користувачами відповідно; n – кількість об’єктів у каталозі; $\bar{r}_1$ і $\bar{r}_2$ – середні оцінки 1-го та 2-го користувачів відповідно.

Значення $k(u_1,u_2)$ належить інтервалу від -1 до 1, де -1 відповідає абсолютній несхожості користувачів, а 1 – абсолютній схожості.

Формула (4.21) передбачає, що необхідно взяти різницю між кожною оцінкою r_{1i} та середнім значенням оцінок $\bar{r}_1$. В цілях оптимізації в реальних додатках для здійснення розрахунків дану формулу приводять до наступного вигляду:

$$k(u_1,u_2)=\frac{n\cdot\sum_{i=0}^{n}(r_{1i}\cdot r_{2i})-(\sum_{i=0}^{n}r_{1i}\cdot\sum_{i=0}^{n}r_{2i})}{\sqrt{n\cdot\sum_{i=0}^{n}r_{1i}^2-(\sum_{i=0}^{n}r_{1i})^2\cdot n\cdot\sum_{i=0}^{n}r_{2i}^2-(\sum_{i=0}^{n}r_{2i})^2}}. \quad (4.22)$$

Періодично перераховуючи $k(u_1,u_2)$ можна фіксувати відносну зміну вподобань користувачів u_1 та u_2, тобто, при зміні $k(u_1,u_2)$ або u_1 змінив свої

вподобання, або – u_2, або обидва користувача. Для фіксування не відносної, а абсолютної зміни вподобань користувача слід використовувати інші підходи, але в даній роботі нас цікавить саме відносна зміна вподобань, адже саме через неї виникає необхідність перераховувати коефіцієнти подоби.

В наборі даних MovieLens містяться наступні дані: id користувачів, оцінки користувачів фільмам, теги до фільмів (поставлені користувачами), id фільмів, назви фільмів, жанри фільмів, часові мітки здійснення дій користувачів (а саме час виставлення оцінки, час створення тегу). Так як оцінки користувачі даної системи повинні ставити у вигляді кількості зірочок, і можна використовувати половинку зірочки, а максимальна кількість зірочок – п’ять, то оцінки можуть бути наступними: [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0].

Зв’язки між елементами набору даних MovieLens вказані за допомогою списків суміжності у файлах електронних таблиць формату .csv, у яких в одному рядку містяться елементи пов’язані між собою, наприклад, id користувача, id фільму та оцінка, яку виставив даний користувач даному фільму.

Час у MovieLens записаний у форматі Unix-часу та містить дані починаючи з наступної дати 28.07.1996.

Для проведення експерименту увесь час у наборі даних було поділено на рівні інтервали, в наведеному у даній роботі прикладі ці інтервали були обрані тривалістю 1 млн. секунд – приблизно по 11 днів. Даний проміжок часу був обраний з огляду особливостей використовуваного набору даних, а саме, вподобання фільмів у людей змінюється не так часто, бажання переглянути фільми інших жанрів може виникнути на протязі декількох днів. Спочатку було використано різні інтервали часу, на вказаному інтервалі можна було спостерігати періоди стабільності та зміни коефіцієнтів подоби користувачів системи, а число розрахунків на всіх наборах даних не перевищувало 700 перерахунків, тому не потребувало багато часу на експеримент.

Для кожного з інтервалів коефіцієнти подоби між користувачами перераховувалися, дані для перерахунку бралися по накопиченню. Дані вимірювань записувалися до .csv файлу у форматі наведеному у таблиці 4.3.

Таблиця 4.3. Формат даних для експериментів з порівняння коефіцієнтів подоби користувачів у різних часових інтервалах

User Id 1	**User Id 2**	**Коефіцієнти подоби користувачів 1 та 2, k**				
		time 1	**time 2**	**...**	**...**	**time N**
1	2	$k_{12} \in [-1;1]$	$k_{12} \in [-1;1]$	...	...	$k_{12} \in [-1;1]$
1	3	$k_{13} \in [-1;1]$	$k_{13} \in [-1;1]$	...	...	$k_{13} \in [-1;1]$
…	…	...	...	...	...	...
n	m	$k_{nm} \in [-1;1]$	$k_{nm} \in [-1;1]$	...	...	$k_{nm} \in [-1;1]$

В результаті було отримано часові ряди для кожної пари користувачів, що містили значення їх коефіцієнтів подоби у різні інтервали часу. Коефіцієнти подоби враховувалися тільки для періодів активності досліджуваних користувачів. Тобто, коли користувачі припиняли свою активність у системі, коефіцієнти подоби для них більше не обчислювалися, так як вони б залишалися незмінними не з причини незмінності вподобань, а з причини відсутності нової інформації, будучи обчисленими на основі старих даних, що перестали оновлюватися.

Для опрацювання отриманих даних було обрано обмежену кількість пар користувачів, для яких періоди активності перекривалися достатньо для проведення аналізу.

Метою обробки даних зібраних у форматі з табл. 4.3 є підрахунок періодів постійності коефіцієнтів подоби k_{ij}, що дозволить визначити час, за який жоден з пари користувачів не змінив вподобання (тобто, між парою користувачів не змінився коефіцієнт подоби). Для визначення такого проміжку часу проводиться пошук в таблиці валідного значення кореляції оцінок $k_{ij}(t_1)$. Потім шукається проміжок часу $[t_1; t_2]$ на якому:

$$|k_{i,j}(t_1) - k_{i,j}(t_2)| > d, \tag{4.23}$$

де d є межею чутливості, в конкретному випадку d обрано рівним 0.01.

У великій кількості часових періодів досліджуваного набору даних активність певних користувачів є недостатньою, тому для них на цих часових періодах коефіцієнти подоби розрахувати не є можливим. З цієї причини, часові проміжки, які завершуються невалідним значенням, в подальший аналіз не поступали. В результаті для визначених проміжків отримано масив: $T(n)$ – де n означає порядковий номер визначеної послідовності; T – тривалість послідовності в умовних одиницях часу.

Отриманий масив дозволяє побудувати діаграму частоти подій появи проміжків стабільності коефіцієнтів подоби пар користувачів $N(n)$ (рис. 4.12).

Рис. 4.12. Діаграма частоти інтервалів стійкості вподобань від їх довжини, де: n – довжина інтервалу часу стабільності коефіцієнта подоби пари користувачів; N – кількість часових інтервалів стабільності коефіцієнтів подоби користувачів довжиною n

Таким чином на рисунку 4.12 була побудована діаграма частоти інтервалів стійкості вподобань користувачів системи від їх тривалості, що

дає можливість отримати нормуванням, при якому проводиться наступна операція $p(n) = N(n) / \sum_i N(i)$, функцію ймовірності отримання ряду вказаної довжини n. Попередньо, при побудові діаграми, дані були згладжені ковзним середнім. Після чого були побудовані регресії, які відповідають популярним спадаючим розподілам. Найкращі наближення дала показникова регресія $N(n) \approx 80e^{-0.17n}$ з середнім квадратичним відхиленням $S \approx 19$. В той же час, якщо прийняти наявність розподілу Парето, то матимемо наближення $N(n) \approx 79n^{-0.93}$ з середнім квадратичним відхиленням $S \approx 24$. Отримані дані дають перевагу екпоненційному розподілу, бо його використання дало змогу отримати менше середньоквадратичне відхилення результату апроксимації від експериментальних даних. На жаль, різниця між відхиленнями найкращих наближень є незначною на фоні випадкових відхилень графіку, тому твердження про експоненційний розподіл потрібно розуміти лише як робочу гіпотезу.

Також для моделювання процесу зміни вподобань застосуємо припущення, що кожне вподобання змінюється у випадковий час і незалежно одне від одного, бо саме такий процес має експоненційний розподіл, аналогом якого є радіоактивний розпад. Тоді можна перейти до визначення часу, протягом якого користувач змінить вподобання з ймовірністю 0.5.

Визначення ймовірності незмінності вподобань (в межах d) в залежності від часу можливе, якщо представити дані з рис. 4.12 таким чином, як показано на рис. 4.13.

Графік з рис. 4.13 отриманий наступним чином. На початок часу було відмічено загальну кількість користувачів за якими велося спостереження. Далі, на кожен відлік часу відмічалася кількість користувачів, які ще зберегли свої вподобання незмінними, що визначалося як різниця від попереднього по часу значення на даному графіку та відповідного по часу значення на графіку з рис. 4.12. З метою підтвердження припущення про експоненційний закон розподілу була побудована експонентна регресія, яка показала достатньо мале середньоквадратичне відхилення.

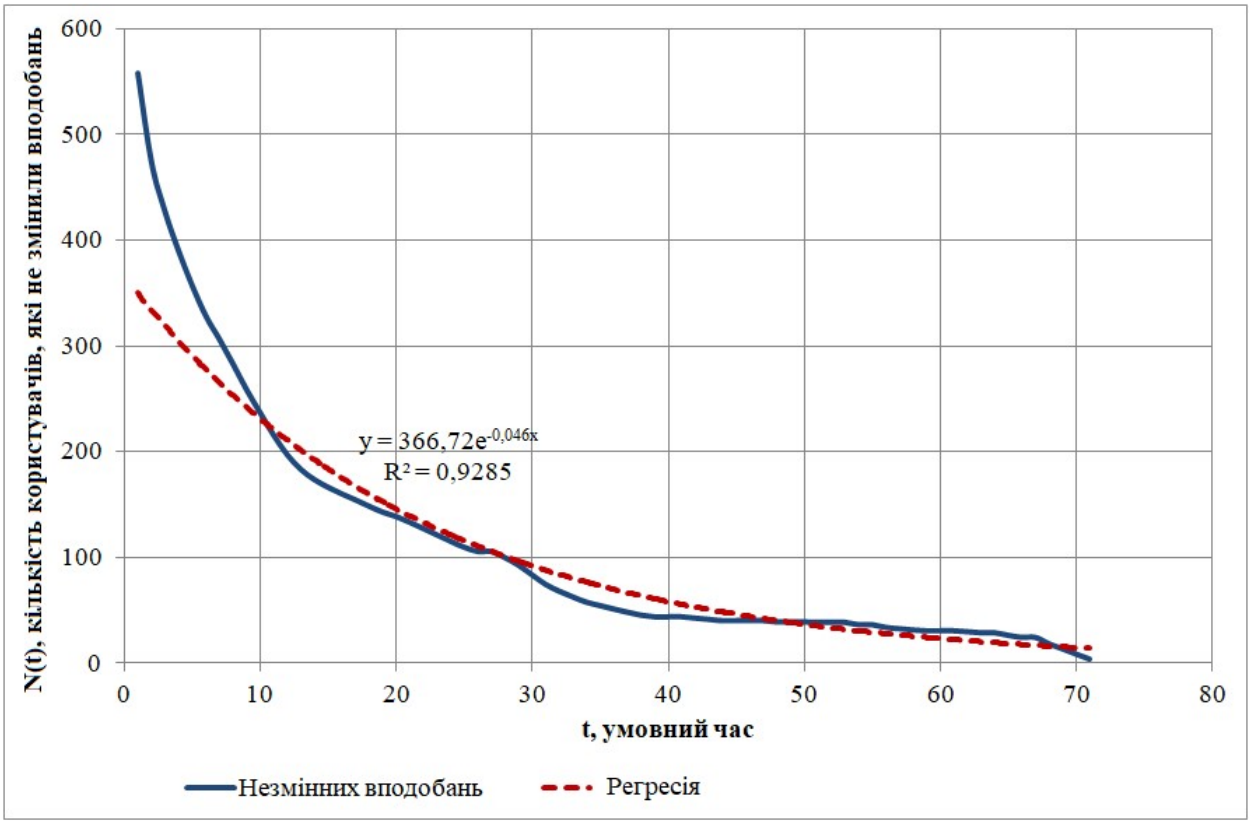


Рис. 4.13. Залежність кількості користувачів, які не змінили своїх вподобань, від часу, де: t – умовний час; k – кількість пар користувачів, які не змінили свого коефіцієнту подоби за певний час t

За прийнятою гіпотезою про експоненційний закон розподілу зміни вподобань користувачів у часі, можна до даного процесу застосувати закони радіоактивного розпаду елементів, з причини співпадіння прийнятих базових законів розподілу.

Графік з рис. 4.13 було використано для полегшення пошуку періоду, коли змінить вподобання задана доля користувачів (наприклад, 50%), за аналогією до закону радіоактивного розпаду.

Дійсно, рис. 4.13 показує графік кількості користувачів $N(t)$, що не змінили своїх вподобань за час t, який можна наблизити наступною експонентою [106]:

$$N(t)=N_0 e^{-\lambda t}. \quad (4.24)$$

де N_0 – початкова кількість користувачів; λ – шуканий коефіцієнт регресії; t – умовний час.

Для статистичного опису змін вподобань достатньо визначити коефіцієнти N_0 та λ за допомогою експонентної регресії. Регресія проводиться логарифмуванням лівої та правої частин рівняння:

$$\ln(N(t_i)) = \ln(N_0) - \lambda t_i.$$

Регресія проводиться на експериментальних даних, які наведено наступною таблицею 4.4.

Таблиця 4.4. Дані для експонентної регресії

t_i	1	2	3	...	n-1	n
$\ln(N(t_i))$	$\ln(N_1)$	$\ln(N_2)$	$\ln(N_3)$		$\ln(N_{n-1})$	$\ln(N_n)$

Квадрат похибки регресії знаходиться за наступною формулою:

$$\Delta_i^2 = \left(\ln(N(t_i)) - (\ln(N_0) - \lambda t_i)\right)^2.$$

Якщо розкрити дужки, то формула набуде вигляду, який можна використати для пошуку середнього квадратичного відхилення R^2:

$$\Delta_i^2 = \ln^2(N(t_i)) - 2\ln(N(t_i))\ln(N_0) + 2\lambda t_i \ln(N(t_i)) + \\ + \ln^2(N_0) - 2\lambda t_i \ln(N_0) + \lambda^2 t_i^2,$$

$$R^2 = \overline{\ln^2(N(t_i))} - 2\ln(N_0)\overline{\ln(N(t_i))} + 2\lambda\overline{t_i \ln(N(t_i))} + \\ + \ln^2(N_0) - 2\lambda\ln(N_0)\overline{t_i} + \lambda^2\overline{t_i^2},$$

де рискою зверху позначено операцію обчислення середнього арифметичного по всім даним регресії: i=1..n.

Для знаходження екстремуму середньоквадратичного відхилення регресії від експериментальних даних, що є єдиним мінімумом, прирівняємо похідні по шуканим коефіцієнтам до нуля:

$$\left(R^2\right)_\lambda' = 2\overline{t_i \ln(N(t_i))} - 2\ln(N_0)\overline{t_i} + 2\lambda\overline{t_i^2}, \qquad (4.25)$$

$$\left(R^2\right)_{\ln(N_0)}{}' = -2\overline{\ln(N(t_i))} + 2\ln(N_0) - 2\lambda\overline{t_i}\,, \tag{4.26}$$

що дає наступну систему лінійних рівнянь:

$$\begin{cases} \overline{t_i \ln(N(t_i))} - \ln(N_0)\overline{t_i} + \lambda\overline{t_i^2} = 0, \\ -\overline{\ln(N(t_i))} + \ln(N_0) - \lambda\overline{t_i} = 0. \end{cases} \tag{4.27}$$

В результаті розв'язання системи матимемо наступні вирази для розрахунку шуканих коефіцієнтів:

$$\lambda = \frac{\overline{t_i} \cdot \overline{\ln(N(t_i))} - \overline{t_i \ln(N(t_i))}}{\overline{t_i^2} - \overline{t_i}}, \tag{4.28}$$

$$N_0 = \exp\left(\frac{\overline{t_i^2} \cdot \overline{\ln(N(t_i))} - \overline{t_i} \cdot \overline{t_i \ln(N(t_i))}}{\overline{t_i^2} - \overline{t_i}}\right). \tag{4.29}$$

Після розрахунків, крива буде дійсно відображати експоненційний розподіл $N(n) \approx 366.72e^{-0.046n}$. В результаті, показник $\lambda = 0.046$ дозволяє визначити наступні величини як аналог закономірності розпаду радіоактивних ізотопів атомів хімічних елементів [131]:

– Середній час життя вподобання $\tau = 1/\lambda$. В розглянутому прикладі середній час стабільності коефіцієнту подоби пари користувачів рекомендаційної системи складає $\tau = 21.7$ обраних проміжків часу у проведеному експерименті. З огляду на те, що розглядаються вподобання для рекомендаційної системи фільмів, можна припустити, що трохи менше одного року смаки користувача стосовно жанрів кінематографу перебувають у деякій стабільності.

– Період напіврозпаду – час, за який у половини пар користувачів рекомендаційної системи зміняться коефіцієнти подоби: $T_{1/2} = \tau\ln 2$, $T_{1/2} \approx 15$ обраних проміжків часу у проведеному експерименті.

– Ймовірність зміни вподобання користувача рекомендаційної системи за час t:

$$p(t) = 1 - e^{-\lambda t}. \tag{4.30}$$

– Ймовірність залишити незмінним вподобання за час t**:**

$$q(t)=e^{-\lambda t}. \tag{4.31}$$

Для моделювання та обслуговування рекомендаційних систем найбільш важливим є критерій впевненості в тому, що користувач за час $t_н$ не змінив вподобання з ймовірністю $p_н$:

$$p_н = \exp(-\lambda t_н), \tag{4.32}$$

$$t_н = -\ln(p_н)/\lambda. \tag{4.33}$$

Також більш корисною може бути формула, яка показує час, за який користувач змінить вподобання з ймовірністю $q_н$:

$$t_н = -\ln(1-q_н)/\lambda. \tag{4.34}$$

Таким чином проведене дослідження показало, що для моделювання нециклічних змін вподобань користувачів у часі можна використовувати експоненційний закон розподілу та закон радіоактивного розпаду елементів. Моделювання змін вподобань у часі у програмній імітаційній моделі користувачів та об’єктів рекомендаційної системи дозволить наблизити згенеровані нею набори даних за статистичними властивостями до таких, що створюються в процесі роботи реальних систем.

Висновки до розділу 4

У даному розділі запропоновано метод програмного імітаційного моделювання користувачів та об’єктів рекомендаційної системи для соціальної мережі та для контент-орієнтованого веб-сайту, що дозволяє генерувати набори даних для тестування алгоритмів роботи рекомендаційних систем. 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 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.

Також у даному розділі запропоновано спосіб моделювання змін вподобань у часі звичайних користувачів рекомендаційної системи, що дозволяє генерувати тестові набори даних, більш схожі за статистичними характеристиками на реальні. Спосіб засновано на математичній моделі нециклічних змін вподобань користувачів у часі з використанням експоненційного закону розподілу та закону радіоактивного розпаду елементів.

РОЗДІЛ 5.
МОДЕЛЬ ТА МЕТОД ВИЯВЛЕННЯ ІНФОРМАЦІЙНИХ АТАК НА РЕКОМЕНДАЦІЙНУ СИСТЕМУ

У даному розділі запропоновані математична модель та метод виявлення інформаційних атак на рекомендаційну систему.

Найчастіше інформаційні атаки на рекомендаційні системи здійснюються у маркетингових цілях з метою підвищення рейтингів товарів атакуючої сторони або зниження рейтингів товарів конкурентів. Хоча атаки на рекомендаційні системи можуть мати на меті також і поширення інформаційних впливів під час інформаційних протиборств, наприклад, у політичних цілях. Через соціальні мережі часто здійснюються різні інформаційні впливи [91, 111, 118, 120, 135, 140, 156, 195], і рекомендаційні системи, як їх складова, стали однією з цілей для інформаційних атак з метою здійснення таких впливів [22, 28, 40, 46, 47, 61, 69, 148]. Виконавши успішну атаку на рекомендаційну систему соціальної мережі або агрегатора новин, можна змінити наповнення та порядок показу об’єктів у стрічках новин користувачам системи. Це можна використати не тільки в маркетингових, а й політичних чи шахрайських цілях.

Для реалізації атак на рекомендаційні системи використовуються мережі ботів, так як тільки певна сукупність профілів у системі здатна вплинути на формування рекомендацій своїми згуртованими діями [22, 28, 40, 46, 47, 69, 76].

Переважно в існуючих дослідженнях [22, 69, 76, 98, 103, 104] пропонується вважати виявлення інформаційної атаки на рекомендаційну систему тотожним виявленню профілів ботів.

Оскільки виявлення профілів ботів досить ресурсномістка задача, у даній роботі пропонується розділити задачу захисту рекомендаційної системи від інформаційних атак на дві частини: 1) виявлення ознак атаки та 2) виявлення і нейтралізація профілів ботів.

Виявлення атаки може бути менш ресурсозатратною задачею і полягати у відслідковуванні динаміки рейтингів об’єктів системи, усіх або тільки критично важливих з точки зору інформаційної безпеки. Наприклад, якщо рейтинги об’єктів, що потребують захисту, починають стрімко змінюватися, а нові оцінки, що призводять до зміни рейтингів, не відповідають попереднім середнім оцінкам цих об’єктів, слід здійснити перевірку на наявність ботів серед користувачів, що почали ставити такі оцінки. Цей підхід скоротить кількість перевірок профілів користувачів. По-перше, тому що перевіряти їх буде потрібно тільки при виявленні підозри на атаку. А, по-друге, тому що треба буде перевіряти профілі не всіх користувачів, а тільки тих, що здійснюють підозрілі дії.

5.1. Розробка математичної моделі підсистеми інформаційної безпеки рекомендаційної системи

Було розроблено набір можливих станів підсистеми безпеки рекомендаційної системи в умовах інформаційних атак ін’єкцією профілів [57]. На основі раніше розробленого методу визначення динаміки ймовірностей перебування системи в своїх можливих станах [58] отримано аналітичні співвідношення для розрахунку ймовірностей перебування системи в цих станах в довільний момент часу [57].

Враховуючи проведене дослідження загроз інформаційній безпеці рекомендаційним системам, базових моделей атак на рекомендаційні системи та способів виявлення і нейтралізації таких інформаційних загроз [57, 142, 148, 164, 166, 167, 174, 176-178], було запропоновано наступний набір станів підсистеми інформаційної безпеки рекомендаційної системи:

1) *Нормальна робота* (стан H_1). В цьому режимі відбуваються додаткові витрати на організацію виявлення наявності інформаційної атаки, витрати пропорційні часу роботи системі в поточному стані та інтенсивності заходів контролю: vtL_1.

2) *Система атакована* (стан H_2). У системі наявні активні бот-мережі. Боти підсистемою безпеки не виявлені, під їх впливом спотворюються рекомендації користувачам. У цьому режимі роботи накопичуються збитки від діяльності ботів tL_2 пропорційно часу. В той самий час проводяться перевірки на наявність інформаційної атаки з витратами tL_1.

3) *Система відбиває атаку* (стан H_3). Наявність інформаційної атаки виявлено. Втрати від неправильних рекомендацій продовжуються tL_2. Ресурси витрачаються на пошук та ліквідацію ботів. У цьому режимі додатково витрачаються ресурси tL_3 на організацію повернення роботи системи в нормальний стан H_1 пропорційно часу роботи.

Граф системи показано на рис. 5.1.

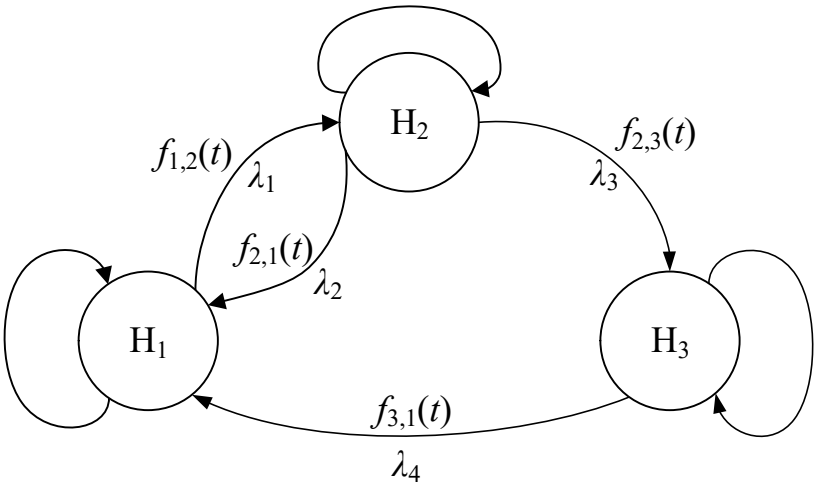


Рис. 5.1. Граф динаміки станів рекомендаційної системи в умовах інформаційної атаки

Для розробки математичної моделі динаміки станів рекомендаційної системи було використано метод ймовірнісного аналізу динаміки станів багатомірних марківських та напівмарковських динамічних систем, розроблений у роботі [58].

Розглянемо можливі сценарії, які можуть відбуватися з рекомендаційною системою, вразливою до інформаційних атак.

Система знаходиться в нормальному стані H_1. Вона може перейти в стан

H_2, ймовірність такого переходу є випадковим процесом з інтенсивністю потоку λ_1. Тоді густина розподілу перебування системи в стані H_1 до переходу у стан H_2 матиме вигляд: $f_{1,2}(t) = \lambda_1 e^{-\lambda_1 t}$. Перехід зі стану H_1 до стану H_3 є малоймовірним, з огляду на те, що час початку атаки на систему не може збігатися з часом виявлення факту ураження системи. Тому потік зі стану H_1 до стану H_3 є відсутнім: $f_{1,3}(t) = 0$. Але стан H_2 може змінитися двома шляхами:

1) атака пройшла непомітно, і через деякий час система стабілізувалася самостійно у стан H_1: $f_{2,1}(t) = p_{2,1} \cdot \lambda_2 e^{-\lambda_2 t}$, до розподілу додано ймовірність $p_{2,1}$, яка визначає, що процес розвиватиметься саме таким шляхом;

2) атаку було помічено, по факту наявності атаки проводяться активні протидії – система перейшла у стан H_3: $f_{2,3}(t) = p_{2,3} \cdot \lambda_3 e^{-\lambda_3 t}$; ця послідовність подій є доповненням до 1 сценарію, що визначається множником-ймовірністю подовження події із вказаною ймовірністю. Завдяки використанню підсистеми визначення наявності інформаційної атаки, ця ймовірність сягає більше, ніж 80%, тому приймемо нижню межу $p_{2,3} = 0.8$, а ймовірність $p_{2,1} = 1 - p_{2,3} = 1 - 0.8 = 0.2$.

Якщо система знаходиться у стані H_3, то через деякий час, вона повертається у стан нормального функціонування H_1. Навіть при здійсненні атаки на рекомендаційну систему у стані H_3, її можна не враховувати, бо атака є досить довгим процесом, і система гарантовано повернеться до стану H_1 до переходу у стан H_2. Тому густини ймовірності знайти рекомендаційну систему у іншому стані через час t є $f_{3,1}(t) = \lambda_4 e^{-\lambda_4 t}$, $f_{3,2}(t) = 0$.

Позначимо як $G_{i,j}(t)$, де $i, j = 1, 2, 3$, ймовірність системи опинитися у стані H_j через час t, якщо в початковий момент часу система знаходилася у стані H_i. Тоді можна записати систему інтегральних рівнянь (5.1).

Тут ймовірність виду $\left(1 - \int f_{1,2}(\tau) d\tau - \int f_{1,3}(\tau) d\tau\right)$ означає, що подія переходу в інший стан за час t не відбулася. Завдяки цьому система є

незалежною і допускає нетривіальні розв’язки.

$$\begin{cases} G_{1,1}(t)=\left(1-\int f_{1,2}(\tau)d\tau-\int f_{1,3}(\tau)d\tau\right)+\int f_{1,2}(\tau)\cdot G_{2,1}(t-\tau)d\tau+ \\ \qquad +\int f_{1,3}(\tau)\cdot G_{3,1}(t-\tau)d\tau \\ G_{2,1}(t)=\int f_{2,1}(\tau)\cdot G_{1,1}(t-\tau)d\tau+\int f_{2,3}(\tau)\cdot G_{3,1}(t-\tau)d\tau \\ G_{3,1}(t)=\int f_{3,1}(\tau)\cdot G_{1,1}(t-\tau)d\tau+\int f_{3,2}(\tau)\cdot G_{2,1}(t-\tau)d\tau \\ G_{1,2}(t)=\int f_{1,2}(\tau)\cdot G_{2,2}(t-\tau)d\tau+\int f_{1,3}(\tau)\cdot G_{3,2}(t-\tau)d\tau \\ G_{2,2}(t)=\left(1-\int f_{2,1}(\tau)d\tau-\int f_{2,3}(\tau)d\tau\right)+\int f_{2,1}(\tau)\cdot G_{1,2}(t-\tau)d\tau+. \\ \qquad +\int f_{2,3}(\tau)\cdot G_{3,2}(t-\tau)d\tau \\ G_{3,2}(t)=\int f_{3,1}(\tau)\cdot G_{1,2}(t-\tau)d\tau+\int f_{3,2}(\tau)\cdot G_{2,2}(t-\tau)d\tau \\ G_{1,3}(t)=\int f_{1,2}(\tau)\cdot G_{2,3}(t-\tau)d\tau+\int f_{1,3}(\tau)\cdot G_{3,3}(t-\tau)d\tau \\ G_{2,3}(t)=\int f_{2,1}(\tau)\cdot G_{1,3}(t-\tau)d\tau+\int f_{2,3}(\tau)\cdot G_{3,3}(t-\tau)d\tau \\ G_{3,3}(t)=\left(1-\int f_{3,1}(\tau)d\tau-\int f_{3,2}(\tau)d\tau\right)+\int f_{3,1}(\tau)\cdot G_{1,3}(t-\tau)d\tau+ \\ \qquad +\int f_{3,2}(\tau)\cdot G_{2,3}(t-\tau)d\tau \end{cases} \tag{5.1}$$

Для розв’язання даної системи рівнянь можна скористатися перетворенням Лапласа, де результат перетворення позначимо суфіксом ^. Врахування властивостей перетворення дає змогу отримати систему рівнянь у наступному вигляді:

$$\begin{cases} \hat{G}_{1,1}(s)=\dfrac{\left(1-\hat{f}_{1,2}(s)-\hat{f}_{1,3}(s)\right)}{s}+\hat{f}_{1,2}(s)\cdot\hat{G}_{2,1}(s)+\hat{f}_{1,3}(s)\cdot\hat{G}_{3,1}(s) \\ \hat{G}_{2,1}(s)=\hat{f}_{2,1}(s)\cdot\hat{G}_{1,1}(s)+\hat{f}_{2,3}(s)\cdot\hat{G}_{3,1}(s) \\ \hat{G}_{3,1}(s)=\hat{f}_{3,1}(s)\cdot\hat{G}_{1,1}(s)+\hat{f}_{3,2}(s)\cdot\hat{G}_{2,1}(s) \\ \hat{G}_{1,2}(s)=\hat{f}_{1,2}(s)\cdot\hat{G}_{2,2}(s)+\hat{f}_{1,3}(s)\cdot\hat{G}_{3,2}(s) \\ \hat{G}_{2,2}(s)=\dfrac{\left(1-\hat{f}_{2,1}(s)-\hat{f}_{2,3}(s)\right)}{s}+\hat{f}_{2,1}(s)\cdot\hat{G}_{1,2}(s)+\hat{f}_{2,3}(s)\cdot\hat{G}_{3,2}(s). \\ \hat{G}_{3,2}(s)=\hat{f}_{3,1}(s)\cdot\hat{G}_{1,2}(s)+\hat{f}_{3,2}(s)\cdot\hat{G}_{2,2}(s) \\ \hat{G}_{1,3}(s)=\hat{f}_{1,2}(s)\cdot\hat{G}_{2,3}(s)+\hat{f}_{1,3}(s)\cdot\hat{G}_{3,3}(s) \\ \hat{G}_{2,3}(s)=\hat{f}_{2,1}(s)\cdot\hat{G}_{1,3}(s)+\hat{f}_{2,3}(s)\cdot\hat{G}_{3,3}(s) \\ \hat{G}_{3,3}(s)=\dfrac{\left(1-\hat{f}_{3,1}(s)-\hat{f}_{3,2}(s)\right)}{s}+\hat{f}_{3,1}(s)\cdot\hat{G}_{1,3}(s)+\hat{f}_{3,2}(s)\cdot\hat{G}_{2,3}(s) \end{cases} \tag{5.2}$$

Завдяки тому, що початковий стан системи відомий і він є станом H_1, достатньо визначити наступні функції: $G_{1,1}(t), G_{1,2}(t), G_{1,3}(t)$.

Розв'язком системи рівнянь (5.2) є:

$$\begin{aligned}
&\hat{G}_{1,1}(s) = \\
&= \frac{\left(\hat{f}_{1,3}(s)+\hat{f}_{1,2}(s)-1\right)\left(\hat{f}_{2,3}(s)\hat{f}_{3,2}(s)-1\right)}{\left(\left(\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\hat{f}_{2,1}(s)\right)\hat{f}_{3,2}(s)+\left(\hat{f}_{1,2}(s)\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\right)\hat{f}_{3,1}(s)+\hat{f}_{1,2}(s)\hat{f}_{2,1}(s)-1\right)\cdot s}, \\
&\hat{G}_{1,2}(s) = \\
&= \frac{\left(\hat{f}_{1,3}(s)\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\hat{f}_{2,1}(s)-\hat{f}_{1,3}(s)\right)\hat{f}_{3,2}(s)+\hat{f}_{1,2}(s)\hat{f}_{2,3}(s)+\hat{f}_{1,2}(s)\hat{f}_{2,1}(s)-\hat{f}_{1,2}(s)}{\left(\left(\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\hat{f}_{2,1}(s)\right)\hat{f}_{3,2}(s)+\left(\hat{f}_{1,2}(s)\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\right)\hat{f}_{3,1}(s)+\hat{f}_{1,2}(s)\hat{f}_{2,1}(s)-1\right)\cdot s}, \\
&\hat{G}_{1,3}(s) = \\
&= \frac{\left(\hat{f}_{1,2}(s)\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\right)\hat{f}_{3,2}(s)+\left(\hat{f}_{1,2}(s)\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\right)\hat{f}_{3,1}(s)-\hat{f}_{1,2}(s)\hat{f}_{2,3}(s)-\hat{f}_{1,3}(s)}{\left(\left(\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\hat{f}_{2,1}(s)\right)\hat{f}_{3,2}(s)+\left(\hat{f}_{1,2}(s)\hat{f}_{2,3}(s)+\hat{f}_{1,3}(s)\right)\hat{f}_{3,1}(s)+\hat{f}_{1,2}(s)\hat{f}_{2,1}(s)-1\right)\cdot s}.
\end{aligned} \tag{5.3}$$

Далі можна використати $f_{1,2}(t)=\lambda_1 e^{-\lambda_1 t}$, $f_{1,3}(t)=0$, $f_{2,1}(t)=p_{2,1}\cdot\lambda_2 e^{-\lambda_2 t}$, $f_{2,3}(t)=p_{2,3}\cdot\lambda_3 e^{-\lambda_3 t}$, $f_{3,1}(t)=\lambda_4 e^{-\lambda_4 t}$, $f_{3,2}(t)=0$, для яких перетворення Лапласа матиме вигляд: $\hat{f}_{1,2}(s)=\lambda_1/(s+\lambda_1)$, $\hat{f}_{1,3}(s)=0$, $\hat{f}_{2,1}(s)=p_{2,1}\cdot\lambda_2/(s+\lambda_2)$, $\hat{f}_{2,3}(s)=p_{2,3}\lambda_3/(s+\lambda_3)$, $\hat{f}_{3,1}(s)=\lambda_4/(s+\lambda_4)$, $\hat{f}_{3,2}(s)=0$. Тоді перетворення Лапласа для шуканих функцій:

$$\begin{aligned}
&\hat{G}_{1,1}(s) = \\
&= \frac{(\lambda_2+s)(\lambda_3+s)(\lambda_4+s)}{s\cdot\left(s^3+k_0+s^2k_2+s\cdot\left(k_1+\lambda_2k_4+\lambda_1\left(k_3-p_{2,1}\lambda_2\right)\right)+\lambda_1\left(p_{2,1}k_1+p_{2,3}\lambda_2k_4\right)\right)}, \\
&\hat{G}_{1,2}(s) = \\
&= \frac{\lambda_1\left(s+\lambda_2-p_{2,1}\lambda_2+p_{2,1}\lambda_3\right)(\lambda_4+s)}{s\cdot\left(s^3+k_0+s^2k_2+s\cdot\left(k_1+\lambda_2k_4+\lambda_1\left(\lambda_2-p_{2,1}\lambda_2+k_4\right)\right)+\lambda_1\left(p_{2,1}k_1+p_{2,3}\lambda_2k_4\right)\right)}, \\
&\hat{G}_{1,3}(s) = \\
&= \frac{p_{2,3}\lambda_1\lambda_3(\lambda_2+s)}{s\cdot\left(s^3+k_0+s^2k_2+s\cdot\left(k_1+\lambda_2k_4+\lambda_1\left(\lambda_2-p_{2,1}\lambda_2+k_4\right)+\lambda_1\left(p_{2,1}k_1+p_{2,3}\lambda_2k_4\right)\right)\right)},
\end{aligned} \tag{5.4}$$

де $k_0=\lambda_2\lambda_3\lambda_4$, $k_1=\lambda_3\lambda_4$, $k_2=\lambda_1+\lambda_2+\lambda_3+\lambda_4$, $k_3=\lambda_2+\lambda_3+\lambda_4$, $k_4=\lambda_3+\lambda_4$.

Так як аналітичне рішення є досить складним та громіздким, було

запропоновано використати чисельні методи. Наявні конкретні значення дозволяють знайти корені або точно, або наближено, тому, для прикладу, використаємо наступні параметри: $\lambda_1 = 0.01$; $\lambda_2 = 0.01$; $\lambda_3 = 0.1$; $\lambda_4 = 0.1$; $p_{21} = 0.2$; $p_{23} = 0.8$, тут λ є інтенсивностями подій і відповідають за час перебування системи в тому або іншому стані. Наприклад, при $\lambda = 0.01$ на 100 умовних одиниць часу буде спостерігатися всередньому одна подія. В реальних рекомендаційних системах значення даних інтенсивностей та ймовірностей може бути будь-яким, їх значення залежить від параметрів системи та факту наявності і параметрів бот-мереж. Дані значення можуть бути визначені власниками рекомендаційної системи на основі аналізу статистичних даних, доступних адміністраторам системи.

В результаті підстановки маємо наступні образи Лапласа шуканих функцій:

$$G_{1,1}^{\wedge}(s) = \frac{25}{34 \cdot s} + \frac{562500 \cdot s^2 + 102500 \cdot s + 3500}{17 \cdot \left(125000 \cdot s^3 + 27500 \cdot s^2 + 1760 \cdot s + 17\right)},$$

$$G_{1,2}^{\wedge}(s) = \frac{7}{34 \cdot s} + \frac{437500 \cdot s^2 + 75000 \cdot s + 3440}{17 \cdot \left(125000 \cdot s^3 + 27500 \cdot s^2 + 1760 \cdot s + 17\right)},$$

$$G_{1,3}^{\wedge}(s) = \frac{1}{17 \cdot s} + \frac{125000 \cdot s^2 + 27500 \cdot s + 60}{17 \cdot \left(125000 \cdot s^3 + 27500 \cdot s^2 + 1760 \cdot s + 17\right)}.$$

Кожна з шуканих ймовірностей має доданок у вигляді числа діленого на s, що відповідає константі. Тобто, для системи, яка стабілізувалася у часі, ймовірності виявити систему у станах H_1, H_2, H_3 відповідно рівні $G_{1,1}$=25/34, $G_{1,2}$=7/34 та $G_{1,3}$=1/17. Ці ймовірності у сумі складають 1, що є одним з аргументів правильності отриманих залежностей.

Наступний дріб, що був представлений у другому доданку образів Лапласа $G_{1,1}^{\wedge}(s)$, $G_{1,2}^{\wedge}(s)$, та $G_{1,3}^{\wedge}(s)$, можна розкласти на елементарні. Для цього потрібно знайти корені знаменника:

$$125000 \cdot s^3 + 27500 \cdot s^2 + 1760 \cdot s + 17 = 0, \text{ звідки:}$$

$$s_1 \approx -0.0117, s_2 \approx 0.02825 \cdot i - 0.1042, s_3 \approx -0.02825 \cdot i - 0.1042.$$

Всі дійсні частини коренів мають від'ємні значення, що відповідає експоненціально спадним залежностям. Тобто, з часом ймовірності збігаються до вказаних констант.

Отже, шукані ймовірності у загальному випадку можна визначити за наступними формулами:

$$G_{1,1}=\frac{\lambda_2\lambda_3\lambda_4}{Z},\ G_{1,2}=\frac{\lambda_1\left(p_{2,3}\left(\lambda_2-\lambda_3\right)+\lambda_3\right)\lambda_4}{Z},\ G_{1,3}=\frac{p_{2,3}\lambda_1\lambda_2\lambda_3}{Z}, \qquad (5.5)$$

де $Z=\left(\lambda_1+\lambda_2\right)\lambda_3\lambda_4+p_{2,3}\lambda_1\left(\lambda_2\left(\lambda_3+\lambda_4\right)-\lambda_3\lambda_4\right).$

Також було розглянуто випадок, коли рекомендаційна система моделюється напівмарківськими процесами. Часто в якості моделі потоку подій використовують розподіли Ерланга другого та більшого порядків, що дозволяє моделювати напівмарківські процеси. Розподіл є продуктом суми двох (для розподілу другого порядку) експоненційних розподілів, що в арифметиці Лапласа представляється як добуток образів:

$$e(t)=\int_0^t f(\tau)f(t-\tau)d\tau,\hat{e}(s)=\hat{f}(s)\hat{f}(s),\hat{e}(s)=\frac{\lambda^2}{(s+\lambda)^2}. \qquad (5.6)$$

В результаті використання величин потоків λ_1=0.01, λ_2=0.01, λ_3=0.1, λ_4=0.1 матимемо наступні вирази:

$$\hat{G_{1,1}}(s)=\frac{25}{34\cdot s}+\frac{1}{17}\cdot$$
$$\cdot\frac{2812.5\cdot k\cdot s^6+1237.5\cdot k\cdot s^5+214.375\cdot k\cdot s^4+18.24\cdot k\cdot s^3+76468750\cdot s^2+1312500\cdot s+8125}{625\cdot k\cdot s^7+275\cdot k\cdot s^6+47.875\cdot k\cdot s^5+4.1525\cdot k\cdot s^4+1860.05\cdot k\cdot s^3+415200\cdot s^2+4280\cdot s+17},$$

$$\hat{G_{1,2}}(s)=\frac{7}{34\cdot s}+\frac{1}{17}\cdot$$
$$\cdot\frac{2187.5k\cdot s^6+962.5k\cdot s^5+166.5k\cdot s^4+14.0875k\cdot s^3+58718250\cdot s^2+1084300\cdot s+7330}{625k\cdot s^7+275k\cdot s^6+47.875k\cdot s^5+4.1525k\cdot s^4+18600500\cdot s^3+415200\cdot s^2+4280\cdot s+17},$$

$$\hat{G_{1,3}}(s)=\frac{1}{17\cdot s}+\frac{1}{17}\cdot$$
$$\cdot\frac{625k\cdot s^6+275k\cdot s^5+47.875k\cdot s^4+4.1525k\cdot s^3+17750500\cdot s^2+228200\cdot s+795}{625k\cdot s^7+275k\cdot s^6+47.875k\cdot s^5+4.1525k\cdot s^4+18600500\cdot s^3+415200\cdot s^2+4280\cdot s+17}.$$

де k=10^8.

Важливо відмітити, що значення ймовірностей для стабілізованої

системи є незмінними: 25/34, 7/34, 2/34. Це є логічним, бо ймовірність отримати систему в тому або іншому стані через час, який перевищує час стабілізування системи не залежить від розподілу, а лише від потоку подій. Але тут є значні зміни в перехідних процесах, динаміка яких визначається степеневими дробами. Нулями знаменника є:

$$s_1=0.0034i - 0.0104,\ s_2= - 0.0034i - 0.0104,$$
$$s_3=0.0140i - 0.0261,\ s_4=- 0.0140i - 0.0261,\ s_5=0.0059i - 0.0796,$$
$$s_6= - 0.0059i - 0.0796,\ s_7= - 0.1079,$$

де всі дійсні частини є від'ємними і гарантують збіжність змін ймовірностей до сталих значень. За модулем дійсні частини коренів є сумірними зі значеннями потоків подій, що означає стабілізацію ймовірностей системи всього при настанні кількох найбільш малоймовірних подій.

В результаті при розв'язанні задач визначення динаміки процесів змін ймовірності є доцільним значно ускладнити обчислення, але для вивчення закономірностей в роботі стаціонарної системи достатньо перевірити стійкість отриманих розв'язків.

Таким чином, розроблено математичну модель динаміки станів підсистеми інформаційної безпеки рекомендаційної системи в умовах інформаційних атак з використанням математичного апарату марківських та напівмарківських процесів.

Також було розроблено спосіб визначення витрат, що зазнає рекомендаційна система внаслідок моніторингу власної інформаційної безпеки та внаслідок інформаційних атак.

Нехай, в ситуації, коли рекомендаційна система перебуває у стані H_2, завдяки неправильно створеним рекомендаціям, втрачаються прибутки пропорційно часу перебування в цьому стані $З_1 = tK_1$. Тут K_1 – це кількість умовних грошових одиниць (*гр.од.*) за одиницю часу, що втрачає власник системи внаслідок вдалої атаки мережі ботів. Якщо проводити тестування на наявність інформаційної атаки, потрібно використати додаткові обчислювальні ресурси, що можна виразити витратами $З_2 = tK_2\lambda_3$. Тут K_2 – це

кількість умовних грошових одиниць за одиницю часу, що втрачає власник системи внаслідок використання додаткових обчислювальних ресурсів для перевірки системи на наявність інформаційних атак. Витрати $З_2$ пропорційні не лише часу перебування системи у стані H_1 та H_2, але й інтенсивності тестування на наявність ботів. Частота такого тестування відповідає частоті виявлення наявності ботів, якщо система була атакована (перехід від стану H_2 до H_3). Стан H_3 відповідає продовженню одержання збитків L_1.

Позначимо час t_1, як долю часу, коли маємо одержання збитків L_1 за рахунок активного втручання ботів. А час t_2, як долю часу перевірки наявності втручання ботів (наявності інформаційної атаки) з витрачанням ресурсів L_2. І час t_3, як долю часу ідентифікації конкретних профілів ботів та усунення наслідків їх діяльності з витрачанням обчислювальних ресурсів tK_3. Тут K_1 – це кількість умовних грошових одиниць за одиницю часу, що втрачає власник системи внаслідок використання додаткових обчислювальних ресурсів для ідентифікації окремих профілів ботів та нейтралізації їх діяльності. Тоді долі часу можна визначити як суми ймовірностей:

– $t_1=G_{1,2}+G_{1,3}$, бо в станах H_2 та H_3 маємо збитки від діяльності ботів;

– $t_2=G_{1,1}+G_{1,2}$, бо в станах H_1 та H_2 маємо збитки від тестування на наявність активності ботів;

– $t_3=G_{1,3}$, бо лише в стані H_3 проводиться активний пошук ботів та нейтралізація їх діяльності.

Ймовірності $G_{1,1}$, $G_{1,2}$ та $G_{1,3}$ визначаються за формулою (5.5).

Відповідно, повні витрати у рекомендаційній системі складатимуть:

$$L=\left(G_{1,2}+G_{1,3}\right)\cdot K_1+\left(G_{1,1}+G_{1,2}\right)\cdot K_2\lambda_3+G_{1,3}\,K_3. \qquad (5.7)$$

Коефіцієнти K_1, K_2, K_3 є параметрами моделі, які залежать від структури та алгоритмів рекомендаційної системи, обсягу її бази даних та обчислювальних потужностей комп’ютерних систем, на яких її розгорнуто. Ці коефіцієнти для кожної конкретної системи будуть різними, і можуть бути

визначені та відомі лише власникам конкретного веб-ресурсу. K_2 та K_3 власники веб-ресурсів визначають на основі тарифних планів своїх хостинг-провайдерів. K_1 визначається на основі середньостатистичних збитків від попередніх атак мереж ботів, наслідками яких могла бути втрата клієнтів, втрата доходів від реклами, збитки на подолання результатів атаки тощо.

Таким чином, за допомогою формули (5.7) можна визначити повні витрати рекомендаційної системи від моніторингу інформаційної безпеки, нейтралізації ботів та діяльності бот-мереж.

Також у даній роботі було розроблено спосіб визначення оптимальної частоти перевірки рекомендаційної системи на наявність інформаційної атаки та профілів ботів.

В підсистемі інформаційної безпеки рекомендаційної системи можна керувати частотою перевірок на наявність інформаційної атаки та діяльності ботів v, що відповідає за значення параметру λ_3.

Для того, щоб визначити оптимальну частоту перевірок системи на наявність інформаційних атак v_{opt}, треба знайти таке значення λ_3, при якому загальні збитки системи L будуть мінімальними:

$$v_{opt} = \arg\min_{\lambda_3} L(\lambda_3) = \arg\min_{\lambda_3} \left\lfloor \left(G_{1,2}+G_{1,3}\right) \cdot K_1 + \left(G_{1,1}+G_{1,2}\right) \cdot K_2 \lambda_3 + G_{1,3}\, K_3 \right\rfloor \quad (5.8)$$

Це рівняння є нелінійним з причини впливу значення λ_3 на коефіцієнти G. Тому його розв'язання в загальному випадку можливе лише за допомогою чисельних методів або методів оптимізації.

Розглянемо приклад визначення оптимальної частоти перевірки системи на наявність інформаційних атак за допомогою чисельних методів.

Для прикладу, візьмемо наступні значення усіх видів витрат рекомендаційної системи:

– K_1=5 *гр.од./хв*;

– K_2=1 *гр.од./хв*;

– K_3=2 *гр.од./хв*.

Такі значення витрат взяті з наступних міркувань, збитки від діяльності

ботів K_1, як правило, більші, ніж витрати на моніторинг стану системи K_2 та на ідентифікацію і нейтралізацію профілів ботів K_3. А також витрати на моніторинг системи з метою виявлення наявності атак K_2 менші, ніж витрати на ідентифікацію та нейтралізацію окремих профілів ботів K_3.

Значення решти констант залишається тим самим, що й раніше:

$$\lambda_1=0.01;\ \lambda_2=0.01;\ \lambda_4=0.1;\ p_{21}=0.2;\ p_{23}=0.8.$$

Визначимо v_{opt} за допомогою проведення табуляції функції витрат (5.7) для λ_3=0; 0.05; 0.1; 0.15; 0.2; 0.25; 0.3, де значення ймовірностей $G_{1,1}$, $G_{1,2}$ та $G_{1,3}$ отримані з (5.5), результати наведено у табл. 5.1.

Вираз для функції витрати при вказаних значеннях параметрів має вигляд:

$$L(\lambda_3)=\frac{0.000195\ +\ 0.0389\lambda_3\ +\ 1.231\lambda_3^2\ +\ 0.9375\lambda_3^3}{0.0000391\ +\ 0.0125\lambda_3\ +\ \lambda_3^2}.$$

Таблиця 5.1. Приклад розрахунків для визначення оптимальної частоти перевірок рекомендаційної системи на наявність інформаційних атак мереж ботів v_{opt}

$\boldsymbol{v=\lambda_3}$	0.00	0.05	0.10	0.15	0.20	0.25	0.3
$\boldsymbol{G_{1,1}}$	1/2	25/36	25/34	75/100	25/33	125/164	75/98
$\boldsymbol{G_{1,2}}$	1/2	9/36	7/34	19/100	6/33	29/164	17/98
$\boldsymbol{G_{1,3}}$	0	2/36	2/34	6/100	2/33	10/164	6/98
$\boldsymbol{L}$	2.5	1.67	1.54	1.51	1.52	1.55	1.58

Отже, було визначено v_{opt} для наведеного прикладу. Графічне відображення побудованих точок для значень повних збитків підсистеми інформаційної безпеки рекомендаційної системи L при різних значеннях частоти перевірок на наявність інформаційних атак v, які поєднані плавною кривою, показані на рис. 5.2.

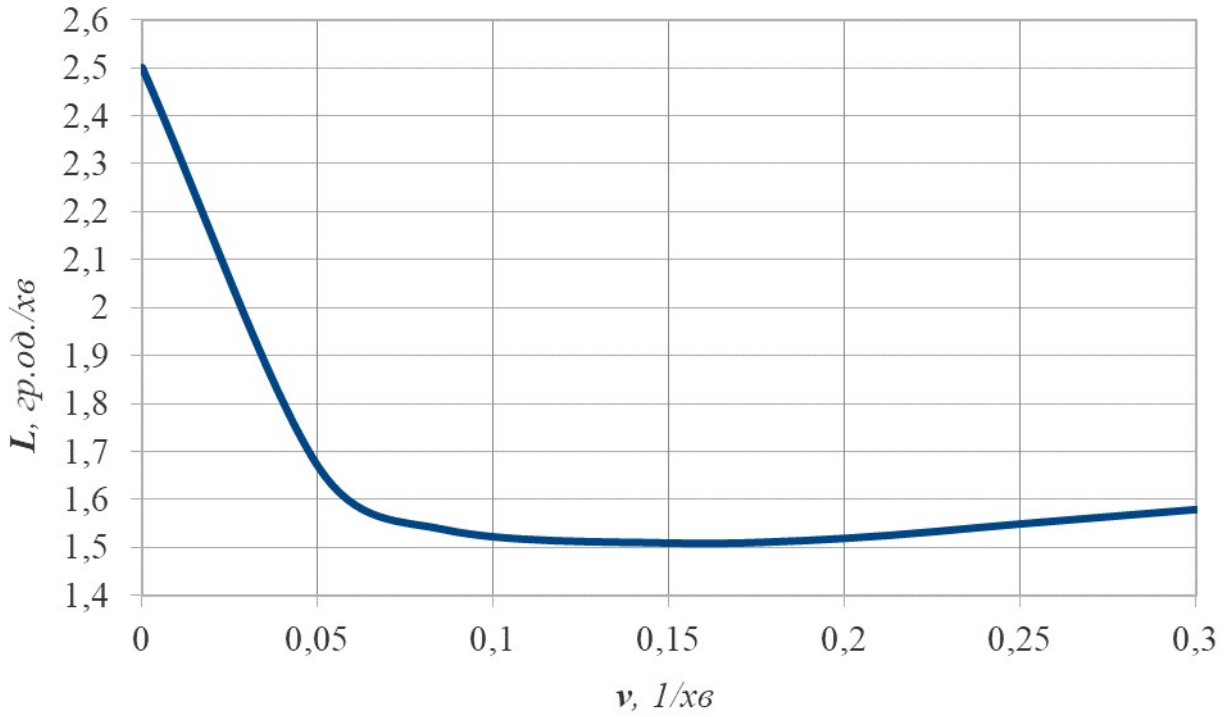


Рис. 5.2. Залежність розміру повних витрат рекомендаційної системи L від частоти її перевірки на наявність інформаційних атак v

З рис. 5.2 та табл. 5.1 можна зробити висновок, що мінімальними загальні витрати системи будуть $L_{min} = 1.51$ *гр.од./хв* при частоті перевірки на наявність інформаційних атак $v_{opt} = 0.16$ *1/хв*, це відповідає періодичності перевірок на наявність ботів $T = 1/\lambda_3 = 6.25$ *хв*. Якщо пошук наявного втручання у рекомендаційну систему проводити з більшою інтенсивністю, витрати зростуть за рахунок додаткових обчислень; якщо виявляти активність ботів з меншою інтенсивністю – матимемо ситуацію зі збільшенням витрат за рахунок неправильної роботи системи під впливом ботів.

Якщо б інтенсивність перевірок для розглянутого прикладу була максимальною $v = \lambda_3 = 1$, тобто, перевірки відбувалися би безперервно, то загальні збитки системи сягнули б $L = 2.18$ *гр.од./хв*. Тож, при застосуванні оптимальної частоти перевірок рекомендаційної системи на наявність інформаційних атак, загальні витрати системи у розглянутому прикладі зменшилися на (2.18–1.51)/2.18 = 30.7%.

Таким чином, запропоновано спосіб визначення оптимальної частоти перевірок рекомендаційної системи на наявність інформаційних атак. Також розглянуто на конкретному прикладі спосіб використання розробленої математичної моделі підсистеми інформаційної безпеки, що описує ймовірнісну динаміку станів рекомендаційної системи в умовах інформаційних атак.

5.2. Розробка методу виявлення інформаційної атаки на рекомендаційну систему

Перш ніж почати розробку методу виявлення інформаційних атак на рекомендаційну систему, було проведено дослідження існуючих підходів до визначення факту наявності таких атак.

5.2.1. Дослідження основних підходів до виявлення інформаційних атак на рекомендаційні системи

Існуючі методи виявлення інформаційних атак на рекомендаційні системи базуються на встановленні факту наявності групи профілів користувачів, що здійснюють атаку, тобто, ідентифікації профілів ботів [22, 69, 76, 98, 103, 104]. Після виявлення профілів ботів, можна вилучити їх інформацію з бази даних, що використовується для формування списків рекомендацій. Таким чином поставлені ними оцінки та виконані дії (перегляди, коментарі тощо) не будуть впливати на рейтинги об’єктів системи.

В такому випадку виявлення атаки на рекомендаційну систему можна розглядати як задачу бінарної класифікації профілів системи [69, 76, 104] з двома можливими результатами для кожного профілю, а саме:

– Профіль справжнього користувача (Authentic);

– Профіль бота, створеного для атаки на систему (Attack).

Для створення такого класифікатору можна використовувати різні методи машинного навчання, які навчати на виборці, що містить як профілі аутентичних користувачів, так і профілі ботів.

Якщо класифікатор використовує алгоритм навчання з учителем, то на етапі навчання використовується анотований набір даних профілів, тобто, набір профілів, позначених як Authentic або Attack. Оскільки більшість моделей атак використовують групи профілів ботів, які працюють узгоджено, є корисним розглядати саме групи профілів разом під час класифікації.

Якість роботи такого класифікатору можна оцінювати за допомогою стандартних метрик, таких як точність позитивного прогнозу (5.9) та повнота (5.10):

$$precision = \frac{tp}{tp + fp}, \tag{5.9}$$

де *tp* – правильна класифікація профілю як Attack; *fp* – неправильна класифікація профілю як Attack.

$$recall = \frac{tp}{tp + fn}, \tag{5.10}$$

де *tp* – правильна класифікація профілю як Attack; *fn* – неправильна класифікація профілю як Authentic.

Також корисними будуть й інші метрики, такі як точність негативного прогнозу (5.11) та специфічність (5.12):

$$NPV = \frac{tn}{tn + fn}, \tag{5.11}$$

де *tn* – правильна класифікація профілю як Authentic; *fn* – неправильна класифікація профілю як Authentic.

$$specificity = \frac{tn}{tn + fp}, \tag{5.12}$$

де *tn* – правильна класифікація профілю як Authentic; *fp* – неправильна класифікація профілю як Attack.

Неправильна класифікація аутентичних профілів призводить до вилучення користної достовірної інформації із бази даних рекомендаційної системи, що може негативно вплинути на загальну якість її роботи. Один із способів оцінити цей вплив – обчислити MAE системи до та після виявлення атаки та нейтралізації профілів ботів.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|y(t) - \hat{y}(t)\right|. \tag{5.13}$$

Ефективність нейтралізації атаки можна оцінити за допомогою зсуву прогнозування рейтингів цільового об'єкту до та після виявлення атаки і вилучення профілів ботів з процесу обчислень рекомендаційної системи, наприклад, з використанням формули (2.47).

Виявлення профілю бота

Розподіл оцінок у профілі бота з високою ймовірністю буде відрізнятися від розподілу оцінок у профілях справжніх користувачів. Незважаючи на те, що зловмиснику вигідно створювати профілі ботів якомога більш схожими на профілі аутентичних користувачів системи, у нього ніколи не може бути достатньо інформації та ресурсів для повного усунення відмінностей між ботами та звичайними користувачами.

Ознаками профілю бота можуть бути наступні статистичні особливості [69, 76]: відхилення від середнього значення оцінок є більшим, ніж зазвичай; деяка група профілів, має вищу, ніж зазвичай, подобу до профілю, який перевіряється.

Відхилення від середнього значення оцінок можна оцінити за допомогою наступного показника – відхилення оцінок від середньої угоди (RDMA):

$$RDMA_u = \sum_{i=0}^{n_u}\frac{\left|\frac{r_{u,i} - \bar{r}_i}{l_i}\right|}{n_u}, \tag{5.14}$$

де n_u – кількість об'єктів, які оцінив користувач u; r_u – оцінка, яку поставив користувач u елементу i; l_i – кількість оцінок, виставлених об'єкту i всіма

користувачами; $\bar{r}_i$ – середнє значення усіх оцінок об'єкту *i*.

Ступінь подібності з топ-сусідами дозволяє виявляти цілі групи ботів, оскільки профіль бота для деяких моделей атак може бути сильніше схожий на профілі найбільш схожих на нього користувачів, ніж це відбувається з профілями аутентичних користувачів:

$$DegSim_u = \sum_{v=1}^{k} \frac{sim_{u,v}}{k}, \tag{5.15}$$

де $sim_{u,v}$ – коефіцієнт подоби між користувачами *u* та *v*.

Виявлення групи профілів ботів

Як правило, ці алгоритми використовують кластеризацію профілів системи та намагаються відрізнити кластери профілів атаки від кластерів аутентичних профілів.

Основні методи виявлення груп ботів передбачають розділення профілів користувачів на два кластери на основі їх коефіцієнтів подоби [62, 76]. Аналізуючи статистику кластерів приймається рішення про наявність нападу і, якщо так, то який кластер містить профілі атаки. Усі профілі кластера атаки ігноруються у процесі формування рекомендацій. Вважається, що напад відбувся, якщо різниця в статистичних особливостях профілів для двох кластерів досить велика. Кластер з меншим стандартним відхиленням визначається як кластер ботів. Однак, особливо для невеликих за розміром бот-мереж, значна частина кластеру, ідентифікованого як кластер ботів, може містити аутентичних користувачів.

Використання мереж репутації користувачів для визначення рівня довіри до даних їх профілів

Для боротьби з інформаційними атаками на рекомендаціні системи може бути корисним додавання параметру репутації для користувачів системи [61, 63, 72, 76, 88, 181]. Такий параметр можна додати в рекомендаційну систему різними способами, наприклад, через створення репутаційної системи [38, 88 226], коли користувачі можуть оцінювати рівень довіри один до одного, що часто застосовується на веб-сайтах дошок оголошень та маркетплейсів.

Також іншим способом введення параметру репутації у рекомендаційну систему є автоматичне його обчислення на основі статистики дій користувача [63], якщо можна ідентифікувати деяку множину його дій як шкідливу чи корисну для системи.

При наявності параметру репутація, можна використовувати коефіцієнти репутації та зважувати вклад кожного користувача в прогноз рейтингів при формуванні рекомендацій.

Тож найпростіший спосіб визначення репутації користувача – зробити на веб-ресурсі можливість оцінювати його. Таким шляхом ідуть Інтернет-магазини, дошки об’яв, ресурси з оренди житла, де можна виставляти оцінки та переглядати загальні рейтинги продавців та покупців.

Визначена таким чином репутація може використовуватися рекомендаційною системою, але вона сама по собі вразлива до інформаційних атак, таку репутацію можна підробити діями ботів. А також вона лише опосередковано може вказувати на нечесність користувача у виставлені оцінок об’єктам системи, адже така репутація є індикатором чесності/нечесності поведінки в інших діях.

Також цей параметр може визначатися за допомогою різних алгоритмів на основі статистики дій користувачів, наприклад, значення репутації підвищується, коли користувач оцінює об’єкт більш правдоподібно (для кластеру, у якому знаходиться) та зменшується, коли користувач неправдиво оцінює об’єкт (його оцінка протилежна до оцінок користувачів з його кластеру) [76]. В такій системі для користувача, що прагне максимізувати свій вплив на рекомендації іншим користувачам, доцільно чесно оцінювати об’єкти.

При використанні параметру репутації у рекомендаційних системах та врахуванні його при створенні рекомендацій боти будуть одержувати нижчі значення даного параметру, ніж аутентичні користувачі, тож дані їх профілів будуть мати менший вплив на формування списків рекомендацій або взагалі не будуть враховуватися.

В результаті проведеного дослідження було вирішено для розробки методів ідентифікації профілів ботів використовувати аналіз статистичних характеристик даних з їх профілів та алгоритми кластеризації профілів як найбільш універсальні підходи. Також було вирішено здійснювати ідентифікацію профілів ботів тільки при виявленні ознак атаки, оскільки аналіз та кластеризація усіх профілів системи досить ресурсозатратна задача. Оскільки в усіх досліджених роботах виявлення інформаційної атаки на рекомендаційну систему тотожне знаходженню у системі профілів ботів, було запропоновано інші ознаки атаки на рекомендаційну систему та розроблено метод виявлення атаки з їх використанням.

5.2.2. Розробка методу виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів рейтингів об’єктів системи

Наявність інформаційної атаки на рекомендаційну систему викликає зміну рейтингів деякої групи об’єктів (підвищує або знижує їх), але зміна рейтингів об’єктів не достатня причина, щоб вважати, що відбувається атака, так як рейтинги можуть змінюватися і внаслідок дій автентичних користувачів. Тому було виділено ряд додаткових ознак, що можуть вказувати на інформаційну атаку, зокрема:

– у об’єкта, на досліджуваному невеликому проміжку часу зросла кількість оцінок – це може вказувати на інформаційну атаку, адже, щоб зсунути рейтинги треба створити суттєву кількість ботів, які поставлять цільові оцінки об’єкту.

– об’єкту, на досліджуваному проміжку часу поставили цільові оцінки (найвищі або найнижчі у шкалі оцінювання, в залежності типу атаки, який підозрюється) – адже бот-мережа змінює рейтинги виставленням саме цільових оцінок; також, більш підозріло, якщо об’єкту виставили дуже багато цільових оцінок, а особливо якщо більше, ніж усіх інших оцінок в поточному

проміжку часу, що може вказувати на поспіх атакуючого систему, коли йому треба якнайшвидше просунути свій контент.

– велика дисперсія оцінок об'єкту – природньо, що у об'єкту буде багато різних оцінок, адже при атаці намагаються змінити рейтинги, тобто, наприклад, до атаки було багато низьких оцінок, а бот-мережа виставляє багато високих оцінок.

– об'єкт багато разів потрапив у списки рекомендацій на досліджуваному проміжку часу, більше разів, ніж потрапляв раніше – якщо це атака, то вона успішна, а тільки успішні атаки нас і мають цікавити, тому що їх треба нейтралізувати, а неуспішні атаки можна і проігнорувати у випадку економії ресурсів системи.

Отже, запропонована множина показників, за значеннями яких можна визначити наявність чи відсутність інформаційної атаки на об'єкт рекомендаційної системи має наступний вигляд:

$$Q_{a,i} = \{tr, pr, d_r, d_t, n_r, n_{tr}, n_{rec}\}, \tag{5.16}$$

де tr – тренд динаміки рейтингів об'єкту i, що може приймати наступні значення $\{-1, 0, 1\}$ – відповідно «тренд зменшення рейтингу», «немає змін» та «тренд збільшення рейтингу»; pr – прогнозування тренду динаміки рейтингів об'єкту i, наприклад, на основі показника Херста [129]; d_r – дисперсія рейтигів об'єкту i; d_t – дисперсія часу виставлення оцінок об'єкту i; n_r – кількість оцінок у об'єкту i на досліджуваному проміжку часу; n_{tr} – кількість цільових оцінок у об'єкту i на досліджуваному проміжку часу; n_{rec} – кількість потраплянь об'єкту i у списки рекомендацій користувачам на досліджуваному проміжку часу.

Будемо вважати, що атака на рекомендаційну систему відбувається при цілеспрямованій зміні рейтингів у одного або декількох об'єктів системи згуртованими діями групи профілів користувачів системи. При чому розмір шкоди від атаки не завжди залежить від кількості об'єктів. Успішна зміна рейтингів навіть одного об'єкту може мати великі наслідки, якщо об'єкт

важливий і мова йде, наприклад, про соціальну, політичну чи медичну сферу тощо.

Якщо об’єкти системи можна ранжувати за ступенем їх важливості та потреби у захисті від інформаційних атак, то їм можна призначити відповідні коефіцієнти. І в першу чергу відслідковувати стан та динаміку рейтингів найбільш важливих об’єктів, ігноруючи або в останню чергу відслідковуючи стан рейтингів менш важливих об’єктів.

Усі показники з (5.16) визначити легко, окрім одного – тренду рейтингів. Оскільки визначенням трендів займається економічна наука, наприклад, для визначення змін курсу валют, акцій тощо – звернемося до її методів.

Для того, щоб визначити наявність та напрямок тренду у рейтингах об’єкту рекомендаційної системи можна скористатися одним з наступних методів технічного аналізу [187, 202, 204, 225]:

– За ковзним середнім;

– За декількома ковзними середніми;

– За вершинами Зиґзаґу;

– За свідченнями ADX;

– За NRTR;

– За кольором свічок Heiken Ashi.

Оскільки найбільш універсальними є перші три методи визначення наявності та напрямку тренду, що дозволить адаптувати їх до задачі пошуку трендів у рейтингах об’єктів рекомендаційної системи, розглянемо їх детальніше нижче.

Визначення напрямку тренду за ковзним середнім

Один з найпростіших способів визначення наявності тренду і його напрямку – за ковзним середнім. Можна використовувати як одне ковзне середнє, так і цілий набір, який іноді називають "віялом".

Правило визначення тренду для одного ковзного середнього:

– тренд спрямований вгору, якщо на заданому проміжку часу останнє значення числового ряду знаходиться вище ковзного середнього;

– тренд спрямований вниз, якщо на заданому проміжку часу останнє значення числового ряду знаходиться нижче ковзного середнього.

Коли на заданому проміжку часу останнє значення числового ряду вище/нижче ковзного середнього, то наступне значення часто після цього розгортається в протилежному напрямку. Тобто, даний спосіб дає велику кількість хибних відповідей. Через це використання його як індикатора тренду вельми обмежене. Його можна використовувати тільки в якості самого грубого фільтру тренду.

Визначення напрямку тренду за декількома ковзними середніми

Щоб поліпшити якість визначення тренду за ковзним середнім можна, наприклад, використовувати два і більше ковзних середніх з різними періодами. Тоді правило визначення тренду для будь-якого числа (більше, ніж одного) ковзних середніх з різними періодами буде виглядати так:

– тренд спрямований вгору, якщо на заданому проміжку часу всі ковзні середні збудовані в правильному порядку підвищення до кінця числового ряду;

– тренд спрямований вниз, якщо на заданому проміжку часу всі ковзні середні збудовані в правильному порядку зниження до кінця числового ряду.

У даному методі число помилкових сигналів про зміну напрямку тренду буде меншим, ніж у попередньому. Але збільшаться часові затрати на визначення тренду.

Визначення напрямку тренду за максимумами та мінімумами індикатору Зигзаг

У даному методі використовується правило Чарльза Доу [187]:

– тренд спрямований вгору, якщо кожний наступний локальний максимум графіку числового ряду вище попереднього локального максимуму і, при цьому, кожен наступний локальний мінімум графіку числового ряду також вище попереднього локального мінімуму;

– тренд спрямований вниз, якщо кожний наступний локальний мінімум графіку числового ряду нижче попереднього локального мінімуму і, при

цьому, кожен наступний локальний максимум графіку числового ряду також нижче попереднього локального максимуму.

Локальні максимуми/мінімуми можна знаходити за вершинами індикатору Зигзаґ.

Індикатор Зиґзаґ (ZigZag) – це індикатор тренду у технічному аналізі, що з'єднує локальні мінімуми і максимуми на графіку числового ряду і дозволяє фільтрувати шум. Існує багато різноманітних модифікацій індикатору ZigZag для аналізу фондових ринків, що враховують різні вподобання трейдерів.

Параметр мінімальної зміни значення часового ряду визначає кількість пунктів, на які значення повинне переміститися, щоб сформувати нову "Зіґ" або "Заґ" лінію. Таким чином, Зиґзаґ відображає тільки найбільш значущі зміни і розвороти.

Головний недолік цього способу визначення тренду – в реальному часі неможливо зрозуміти утворився вже екстремум або ще ні.

Цей недолік робить даний спосіб малоцінним для практичного використання в реальному часі. Зате він дуже корисний при технічному аналізі зібраних раніше даних з метою пошуку закономірностей і для оцінки якості роботи системи.

У даній роботі пропонується для визначення наявності тренду рейтингу об’єкту використати декілька ковзних середніх.

Також пропонується здійснювати прогнозування динаміки трендів рейтингів об’єктів системи у найближчому майбутньому. Прогноз збереження виявлених трендів у майбутньому може бути одною з ознак успішної атаки. Для прогнозування динаміки трендів рейтингів об’єктів було запропоновано використовувати *R/S-аналіз* [202].

Алгоритм R/S-аналізу складається з наступних кроків [202]:

1. Дано початковий часовий ряд S_t. Розраховуємо логарифмічне відношення:

$$N_t = \ln \frac{S_t}{S_{t-1}} \tag{5.17}$$

2. Розділимо ряд N на A суміжних періодів довжиною n. Позначимо кожен період як I_a, де $a = 1, 2, \ldots, A$. Визначимо для кожного I_a середнє значення:

$$E(I_a) = \frac{1}{n}\sum_{k=1}^{n} N_{k,a} \tag{5.18}$$

3. Розраховуємо відхилення від середнього значення для кожного періоду I_a:

$$X_{k,a} = \sum_{i=1}^{k}\left(N_{i,a} - E(I_a)\right) \tag{5.19}$$

4. Розрахуємо розмах в межах кожного періоду:

$$R_{I_a} = \max\left(X_{k,a}\right) - \min\left(X_{k,a}\right) \tag{5.20}$$

5. Розрахуємо стандартне відхилення для кожного періоду I_a:

$$S_{I_a} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(N_{k,a} - E(I_a)\right)^2} \tag{5.21}$$

6. Кожен R_{I_a} ділимо на S_{I_a}. Далі розраховуємо середнє значення *R/S*:

$$R/S(n) = \frac{\sum_{a=1}^{A} R_{I_a} / S_{I_a}}{A} \tag{5.22}$$

7. Збільшуємо n і повторюємо кроки 2-6 до тих пір, поки $n \le N/2$.

8. Будуємо графік залежності $\log(R/S(n))$ від $\log(n)$ і за допомогою методу найменших квадратів знаходимо регресію виду: $\log(R/S(n)) = H \cdot \log(n) + c$, де H – показник Херста.

9. Далі перевіряємо отриманий результат на значимість. Для цього перевіряємо гіпотезу про те, що аналізована структура є нормально-розподіленою. Якщо *R/S* є випадковими змінними, нормально розподіленими, тоді можна припустити, що H також розподілені нормально. Асимптотичною границею для незалежного процесу є показник Херста рівний 0.5. Еніс і Ллойд [3], а також Петерс [202] запропонували використовувати такі очікувані показники *R/S*:

$$E(R/S(n)) = \frac{n-0.5}{n} \cdot \left(n \cdot \frac{\pi}{2} \right)^{-0.5} \cdot \sum_{r=1}^{n-1} \sqrt{\frac{n-r}{r}} \quad (5.23)$$

Для n спостережень знаходимо очікуваний показник Херста $E(H)$.

10. Розраховуємо очікувану дисперсію показника Херста за формулою:

$$Variance(H) = \frac{1}{N} \quad (5.24)$$

де H – показник Херста; N – число спостережень у вибірці.

11. Перевіряємо значимість отриманого коефіцієнта Херста шляхом оцінки кількості стандартних відхилень, на які H перевершує $E(H)$. Значущим вважається результат, коли показник значущості по модулю більше 2.

Інтерпретація показників індексу Херста [129]:

– $H = 0.5$ – процес з відсутністю пам'яті, певного тренду немає.

– $H > 0.5$ – процес характеризується персистентністю – має тенденцію до збереження тренду.

– $H < 0.5$ – процес характеризується антіперсистентністю – будь-яку тенденцію прагне змінити протилежна.

Значення показника Херста природних процесів групуються поблизу наступних значень [129]: 0.72-0.73.

При дослідженні стану рекомендаційної системи є сенс звертати увагу на об'єкти, у яких $H > 0.5$ на досліджуваному проміжку часу, такі об'єкти будуть змінювати свої рейтинги відповідно певного тренду на протязі довгого проміжку часу, отже, серед них можуть бути цілі успішних інформаційних атак, особливо якщо $H > 0.73$.

Було розроблено метод виявлення об'єктів інформаційної атаки бот-мережі ін'єкцією профілів у рекомендаційній системі, що складається з наступних етапів:

Етап 1. Формуємо множину об'єктів I для перевірки, вона може містити всі об'єкти рекомендаційної системи або тільки критично важливі об'єкти, що потребують захисту від інформаційних атак.

Етап 2. Визначаємо за допомогою декількох ковзних середніх (або інших методів визначення тренду) для кожного об'єкту з множини I показник tr на проміжку часу τ.

Етап 3. Визначаємо за допомогою R/S-аналізу для кожного об'єкту з множини I індекс Херста H на проміжку часу τ.

Етап 4. Визначаємо для кожного об'єкту з множини I на проміжку часу τ дисперсію оцінок d_r та дисперсію часових проміжків між виставленням цільових оцінок d_t, а також їх середньостатистичні значення у системі $d_{\tau,cep}$ та $d_{t,cep}$.

Етап 5. Визначаємо для кожного об'єкту з множини I на проміжку часу τ кількість виставлених йому цільових n_{tg} та усіх оцінок n_r, а також середньостатистичну кількість цільових $n_{tg,cep}$ та усіх оцінок $n_{r,cep}$ для об'єктів системи.

Етап 6. Визначаємо для кожного об'єкту з множини I на проміжку часу τ кількість потраплянь у списки рекомендацій n_{rec}, а також середньостатистичну кількість потраплянь у списки рекомендацій $n_{rec,cep}$ для всіх об'єктів системи.

Етап 7. Визначаємо наявність та тип атаки за наступними правилами:

Правило 1. Якщо у об'єкта наявні будь-які 5 ознак з даних: тренд на зростання рейтингу $tr_\tau = 1$, $H > 0.73$, $d_\tau \le d_{\tau,cep}$, $d_t \le d_{t,cep}$, $n_r > n_{r,cep}$, $n_{tg} > n_{tg,cep}$, $n_{rec} > n_{rec,cep}$, то вважаємо, що існує висока ймовірність наявності атаки на підвищення рейтингу для цього об'єкту.

Правило 2. Якщо у об'єкта наявні будь-які 5 ознак з даних: тренд на зменшення рейтингу: $tr_\tau = -1$, $H > 0.73$, $d_\tau \le d_{\tau,cep}$, $d_t \le d_{t,cep}$, $n_r > n_{r,cep}$, $n_{tg} > n_{tg,cep}$, $n_{rec} < n_{rec,cep}$, то вважаємо, що існує висока ймовірність наявності атаки на зниження рейтингу для цього об'єкту.

Для тестування розробленого методу було реалізовано програмне забезпечення на мові програмування Python з використанням бази даних Neo4j. Вхідні дані для експериментів були згенеровані у розробленій програмній імітаційній моделі рекомендаційної системи [53], їх формат та

статистичні характеристики були максимально наближені до відповідних характеристик MovieLens Datasets [30]. Атаки моделювалися за допомогою моделей інформаційних атак, наведених у [28, 40, 76].

Було проведено серію експериментів для перевірки ефективності запропонованого методу. Результати експериментів наведені у таблиці 5.2.

Таблиця 5.2. Результати тестування розробленого методу виявлення інформаційної атаки на рекомендаційну систему та об'єктів атаки

№ експерименту	Модель інформаційної атаки ін'єкцією профілів	Кількість об'єктів у системі	Кількість об'єктів атаки ботів	Вірно розпізнані об'єкти атаки, %	Помилково розпізнані як об'єкти атаки, %	RMSE
1	Випадкова атака	200	20	100.00	5.55	0.223
2		200	10	80.00	23.68	0.484
3		200	5	40.00	22.05	0.479
4		200	1	100.00	16.58	0.406
5	Середня атака	200	20	90.00	0.00	0.100
6		200	10	50.00	13.68	0.393
7		200	5	60.00	13.84	0.380
8		200	1	100.00	12.06	0.346
9	Популярна атака	200	20	75.00	0.00	0.158
10		200	10	80.00	12.63	0.360
11		200	5	40.00	17.94	0.435
12		200	1	100.00	15.57	0.393
Середні значення:				**76.25**	**12.79**	**0.346**

Як показали результати експериментів розроблений метод в середньому дозволяє виявити 76% об'єктів інформаційних атак у рекомендаційній системі. Об'єкти, які не зазнавали інформаційної атаки, але помилково були віднесені до можливих цілей атак, в середньому складали 13% від усіх об'єктів системи.

З об'єктів, у яких наявні ознаки інформаційної атаки, можна сформувати множину I_g. Це дозволить при пошуку бот-мереж методами кластеризації та статистичного аналізу профілів користувачів перевіряти не всі профілі системи, а тільки ті, які взаємодіяли з об'єктами множини I_g. Після знаходження бот-мережі можна точніше визначити I_g, виходячи з аналізу діяльності ідентифікованих профілів ботів.

5.2.3. Розробка способу ідентифікації профілів ботів на основі нейронних мереж у рекомендаційній системі

Для ідентифікації профілів ботів було розроблено багатошарову нейронну мережу прямого поширення [56] (рис. 5.3).

Як вхідні дані для штучної нейронної мережі було обрано кількості різних оцінок у профілі користувача.

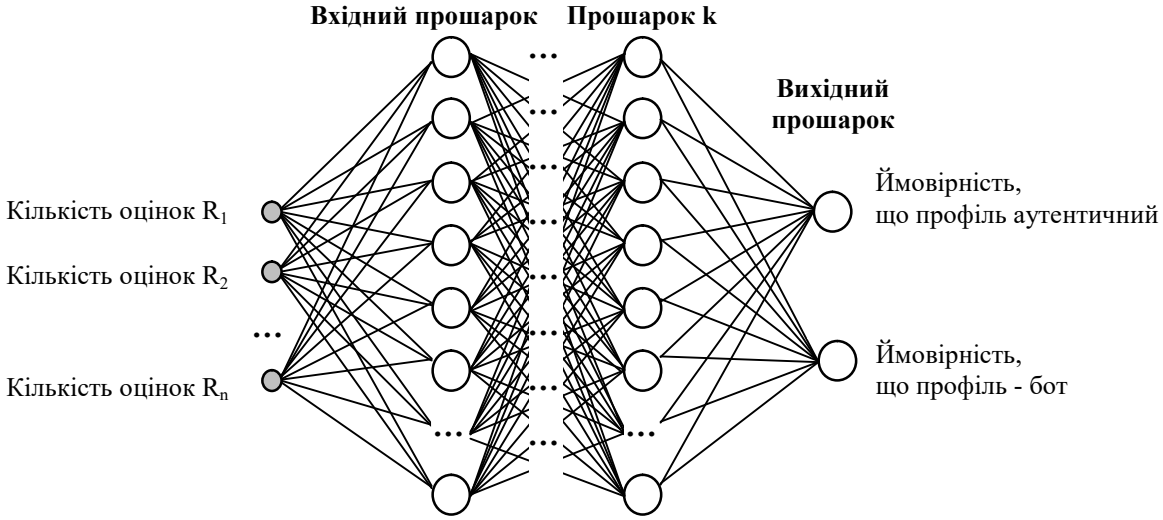


Рис. 5.3. Загальна архітектура штучної нейронної мережі для ідентифікації ботів у рекомендаційній системі

Для реалізації нейронної мережі використана бібліотека TensorFlow [86] та мова програмування Python. Експериментальним шляхом було виявлено, що баланс між точністю та складністю нейронної мережі вдається досягнути при наступних параметрах, що наведені у табл. 5.3.

Таблиця 5.3. Параметри розробленої нейронної мережі

Layer	**Type**	**Кількість нейронів**	**Кількість входів на кожному нейроні**	**Функції активації**
1	вхідний	100	10	sigmoid
2	прихований	100	100	sigmoid
3	прихований	100	100	sigmoid
4	вихідний	2	100	softmax

Для проведення експериментів було використано набір даних, згенерований у розробленій програмній імітаційній моделі рекомендаційної системи [53], його формат та статистичні характеристики були максимально наближені до статистичних характеристик відкритого набору даних MovieLens Datasets [30]. Атаки моделювалися за допомогою моделей інформаційних атак наведених у [28, 40, 76].

Оскільки в MovieLens Datasets десять варіантів оцінок від 0.5 до 5, то входів у нейронній мережі для даного випадку повинно бути 10. Було вирішено створити 2 прихованих прошарки у нейронній мережі. У вхідному та прихованих прошарках використана сигмоїдальна функція активації. Вихідний прошарок складається з 2 нейронів з функцією активації softmax, що показують ймовірність приналежності профіля до аутентичного або бота.

Для навчання нейронної мережі використано алгоритм Адам [191], це модифікація алгоритму стохастичного градієнтного спуску, що отримав широке застосування у системах з глибоким навчанням. Він відрізняється від класичного градієнтного спуску тим, що обчислює індивідуальні адаптивні швидкості спуску для різних ваг, а градієнтний спуск підтримує єдину швидкість для всіх оновлень ваг, і швидкість спуску не змінюється під час тренувань.

Було проведено серію експериментів, результати яких наведено у таблиці 5.4. Позначення використані у таблиці: RA – випадкова атака, AA – середня атака, PA – популярна атака.

Для кожної досліджуваної моделі атаки було створено по 1 навчаючій та по 8 тестових вибірок. Кожна вибірка містила 3000 користувачів. У навчаючій вибірці половина користувачів була ботами. У тестових вибірках серед користувачів було від 5% до 30% ботів. Також були створені набори з різною кількістю цільових об’єктів у ботів – 1 та 10 цілей.

Було створено 3 однакові нейронні мережі для кожної моделі атаки, кожна з яких навчалася на відповідній навчаючій вибірці та тестувалася на відповідних тестових вибірках.

Таблиця 5.4. Результати тестування розробленого методу

№ експ.	Кількість ботів, %	Кількість цілей у кожного бота, шт.	Помилка I роду, %			Помилка II роду, %			Точність (Precision)			Повнота (Recall)			F-міра, %		
			RA	AA	PA	RA	AA	PA	RA	AA	PA	RA	AA	PA	RA	AA	PA
1	5	1	0.001	0.002	0.002	0.023	0.049	0.047	0.95	0.12	0.25	0.52	0.006	0.006	0.67	0.013	0.013
2	10	1	0.0006	0.001	0.0006	0.036	0.099	0.099	0.98	0.25	0.33	0.63	0.003	0.003	0.77	0.006	0.006
3	20	1	0.002	0.001	0.002	0.080	0.199	0.197	0.98	0.28	0.14	0.59	0.003	0.001	0.74	0.006	0.003
4	30	1	0.0006	0.002	0.002	0.116	0.297	0.296	0.99	0.25	0.30	0.61	0.002	0.003	0.75	0.004	0.006
Середні значення:			**0.001**	**0.001**	**0.001**	**0.063**	**0.161**	**0.159**	**0.97**	**0.22**	**0.25**	**0.58**	**0.003**	**0.003**	**0.73**	**0.007**	**0.007**
5	5	10	0.002	0.001	0.001	0.001	0.024	0.009	0.96	0.93	0.97	0.98	0.50	0.81	0.97	0.65	0.88
6	10	10	0.001	0.001	0.002	0.002	0.045	0.014	0.98	0.98	0.96	0.97	0.55	0.85	0.98	0.70	0.90
7	20	10	0.001	0.001	0.001	0.005	0.047	0.013	0.99	0.99	0.99	0.97	0.76	0.93	0.98	0.86	0.96
8	30	10	0.001	0.001	0.002	0.005	0.024	0.006	0.99	0.99	0.99	0.98	0.92	0.97	0.98	0.95	0.98
Середні значення:			**0.001**	**0.001**	**0.001**	**0.003**	**0.035**	**0.010**	**0.98**	**0.97**	**0.97**	**0.97**	**0.68**	**0.89**	**0.97**	**0.79**	**0.93**

Для оцінки якості роботи нейронної мережі було обрано наступні метрики:

1. *Помилки першого роду* – «помилкова тривога», коли звичайний користувач ідентифікований як бот.

2. *Помилки другого роду* – «пропуск цілі», коли бот ідентифікований як аутентичний користувач.

3. *Точність* (*Precision*) – міра, що характеризує, скільки позитивних прогнозів нейронної мережі є правильними. Вона обчислювалася за формулою:

$$Precision = \frac{tp}{tp + fp}, \tag{5.25}$$

де *tp* – позитивні прогнози нейронної мережі, які виявилися вірними; *fp* – позитивні прогнози нейронної мережі, які виявилися помилковими.

4. *Повнота* (*Recall* або *Sensitivity*) – міра, що характеризує можливість нейронної мережі створювати вірні позитивні прогнози, при цьому не враховуються невірні позитивні прогнози. Визначалася за формулою:

$$recall = \frac{tp}{tp + fn}, \tag{5.26}$$

де *tp* – позитивні прогнози нейронної мережі, які виявилися вірними; *fn* – негативні прогнози нейронної мережі, які виявилися помилковими.

5. *F-міра* (*F-score*) – середнє гармонічне точності та повноти.

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{5.27}$$

Як показали експерименти, розроблена нейронна мережа досить точно може розпізнавати ботів, які використовують випадкову атаку незалежно від кількості їх цілей, в середньому точність розпізнавання 0.97. Але у неї виникають значні проблеми при розпізнаванні ботів, що використовують середню та популярну атаки. Нейронна мережа здатна розпізнати таких ботів, тільки якщо у них є декілька цілей. Також з результатів експериментів видно, що у розробленому способі ідентифікації ботів досить рідко виникають помилки першого роду – коли система ідентифікує звичайних користувачів як ботів.

Висновки до розділу 5

У даному розділі запропоновано математичну модель підсистеми безпеки рекомендаційної системи на основі запропонованого раніше методу визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило визначати оптимальну частоту перевірки на наявність інформаційної атаки та профілів ботів. У межах математичної моделі розроблено набір можливих станів, у яких може перебувати рекомендаційна система в умовах інформаційних атак ін’єкцією профілів. Запропоновано три стани роботи рекомендаційної системи в умовах інформаційної атаки, а саме, «нормальний стан», «система атакована» та «система відбиває атаку». Визначено можливі переходи між цими станами. Розроблено аналітичні співвідношення для розрахунку ймовірностей перебування рекомендаційної системи в своїх можливих станах в довільний момент часу.

На основі розробленої математичної моделі розроблено спосіб

визначення повних витрат, що зазнає рекомендаційна система внаслідок моніторингу власної інформаційної безпеки, нейтралізації діяльності бот-мереж та внаслідок інформаційних атак ін'єкцією профілів. Запропоновано формулу для визначення повних витрат рекомендаційної системи в умовах інформаційних атак ін'єкцією профілів. Розроблений спосіб дозволяє при відомих витратах на обчислювальні ресурси та відомих збитках при атаках бот-мереж визначати загальні витрати на обслуговування підсистеми безпеки рекомендаційної системи. Також розроблено спосіб визначення оптимальної частоти перевірки рекомендаційної системи на наявність інформаційної атаки та профілів ботів для оптимізації загальних витрат системи. Запропонований спосіб ґрунтується на використанні чисельних методів. Розглянуто на конкретному прикладі рішення задачі визначення оптимальної частоти перевірки рекомендаційної системи на наявність інформаційної атаки ін'єкцією профілів. Якщо відома середня інтенсивність потоку появи активних бот-мереж у рекомендаційній системі та швидкість роботи алгоритмів по їх виявленню, то можна знизити загальні витрати системи за рахунок оптимізації частоти пошуку атак. В розглянутому конкретному прикладі використання оптимальної частоти перевірки системи на наявність атаки знижує загальні витрати власників системи на 30.7% в порівнянні з постійною перевіркою системи. Оскільки частота появи активних бот-мереж в реальних рекомендаційних системах не буде неперервною у часі, то максимальна частота перевірок на наявність атаки ніколи не буде оптимальною. Таким чином, застосування способу визначення оптимальної частоти перевірки рекомендаційної системи на наявність інформаційної атаки дозволить власникам веб-ресурсів мінімізувати свої фінансові витрати на забезпечення інформаційної безпеки рекомендаційних систем.

Запропоновано метод виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів рейтингів об'єктів, що дозволило знизити кількість витрат на моніторинг безпеки системи за рахунок зняття необхідності пошуку ботів при відсутності ознак атаки. Як показали

результати експериментів розроблений метод в середньому дозволяє виявити 76% об’єктів інформаційних атак у рекомендаційній системі. Об’єкти, які не зазнавали інформаційної атаки та помилково були визначені як такі, що зазнали атаки, в середньому складали 13%. Тож розроблений метод формує множину ймовірних цілей атаки ботів. Множину ймовірних цілей можна використати для подальшого пошуку профілів ботів та уточнення інформації про їх дійсні цілі. Це дозволить при пошуку бот-мереж перевіряти не всі профілі системи, а тільки ті, які взаємодіяли з ймовірними об’єктами атаки.

Запропоновано спосіб ідентифікації профілів ботів на основі нейронних мереж у рекомендаційних системах. Як показали проведені експерименти, найлегше виявити ботів, що здійснюють випадкову атаку, їх можна виявити, навіть якщо ціль у ботів одна. Значно тяжче виявити середню та популярну атаку, їх вдавалося виявити з достатньою точністю, якщо у бота було декілька цілей. При наявності 10 цілей у ботів, незалежно від моделі атаки, розроблений метод ідентифікував профілі ботів в середньому з точністю 0.97. Для підвищення точності роботи нейронної мережі, можна враховувати й інші параметри профілів користувачів, зокрема, час виставлення кожної оцінки, характеристики об’єктів атаки тощо. Це значно підвищить точність розпізнавання, особливо для складних випадків, наприклад, коли ціль у ботів тільки одна.

РОЗДІЛ 6.
МЕТОД ВИЯВЛЕННЯ І НЕЙТРАЛІЗАЦІЇ ЗОВНІШНІХ ДЕСТАБІЛІЗУЮЧИХ ФАКТОРІВ У РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ ТА ДОСЛІДЖЕННЯ ДОСТОВІРНОСТІ ОДЕРЖАНИХ РЕЗУЛЬТАТІВ

У даному розділі запропоновано метод виявлення та нейтралізації мережі ботів у рекомендаційній системі на основі графової кластеризації [132, 144, 151, 152, 192] та аналізу дій користувачів. Даний метод пропонується використовувати після застосування запропонованого у попередньому розділі методу виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів рейтингів об'єктів системи і тільки за умови, якщо той виявив наявність атаки та визначив множину атакованих об'єктів. Проведено експерименти для визначення показників точності його роботи.

Також у даному розділі було проведено серію експериментів для дослідження показників стійкості до зовнішніх дестабілізуючих факторів розроблених методів синтезу стійкої рекомендаційної системи, об'єднаних у єдину систему, що складається з гібридного методу колаборативної фільтрації [55, 64] та підсистеми інформаційної безпеки [57]. Розроблений гібридний метод складається з відомого методу колаборативної фільтрації на основі моделі сусідства, запропонованого методу колаборативної фільтрації з використанням продукційних правил для визначення відсутніх коефіцієнтів подоби між користувачами та запропонованого методу колаборативної фільтрації з врахуванням показників активності користувачів для вирішення проблеми холодного страту. Розроблена підсистема інформаційної безпеки рекомендаційної системи складається з методу виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів у рейтингах об'єктів, методу виявлення бот-мереж у рекомендаційній системі на основі графової кластеризації і аналізу дій користувачів, способу ідентифікації профілів ботів на основі нейронних мереж у рекомендаційних системах.

Вхідні дані для експериментів проведених у даному розділі

генерувалися у розробленій програмній імітаційній моделі користувачів та об’єктів рекомендаційної системи [53], що дозволило симулювати вплив зовнішніх дестабілізуючих факторів на систему.

Також у даному розділі було здійснене обґрунтування достовірності одержаних результатів наукових досліджень.

6.1. Розробка методу виявлення та нейтралізації мережі ботів у рекомендаційній системі

Якщо у системі виявлені об’єкти з аномальною зміною трендів рейтингів, що гіпотетично могло виникнути внаслідок атаки бот-мережі, логічно наступним кроком зробити спробу виявлення профілів користувачів, що вплинули своїми діями на цю зміну та спробувати з’ясувати чи поєднані вони у скоординовану мережу або мережі.

Природно, що на зміну трендів рейтингів об’єктів вплинули усі користувачі, що поставили їм оцінки рівні цільовим – високі, якщо у об’єкту зросли рейтинги, або низькі, якщо у об’єкту знизилися рейтинги. Множину таких користувачів легко виявити простими запитами до бази даних рекомендаційної системи. А от визначити, які профілі користувачів дійсно є частиною бот-мережі та виконують скоординовані дії з іншими її учасниками, а які просто поставили оцінку, що відповідала їх вподобанням – значно складніше.

Задачу виділення бот-мережі серед профілів користувачів системи можна звести до задачі пошуку підграфу у соціальному графі рекомендаційної системи, вершинами якого будуть профілі користувачів пов’язані між собою деякими спільними діями, що мали вплив на зміну трендів рейтингів усіх або більшості об’єктів з множини ймовірних цілей інформаційної атаки [76, 143, 145, 176]. Тому перед розробкою методу виявлення мережі ботів розглянемо існуючі методи графової кластеризації та проведемо тестування їх роботи на даних рекомендаційної мережі.

6.1.1. Дослідження методів кластеризації графів

Методи кластеризації графів, які також називають методами пошуку співтовариств (якщо їх застосовують до соціальних мереж) за принципом роботи можна поділити на наступні: засновані на оптимізації модулярності, засновані на спектральних особливостях графу та засновані на оцінці ентропії системи [121, 132, 151, 158, 221]. За результатами роботи методи пошуку співтовариств можна поділити на такі, що розбивають граф на кластери, які не перетинаються (наприклад, Edge Betweenness, Label Propagation, FastGreedy, WalkTrap, Infomap, Leading Eigenvector, MultiLevel тощо), та на ті, що розбивають граф на кластери, які перетинаються (наприклад, k-Clique Perlocation, BigCLAM, DEMON, CONGO тощо) [192].

Для розділення користувачів рекомендаційної системи на аутентичних та ботів слід застосувати методи, що розбивають граф на кластери, які не перетинаються, адже в даному разі профіль не може потрапляти одразу до двох категорій. Було проведено дослідження найбільш відомих та часто використовуваних методів даного типу, наведене нижче.

Методи кластеризації графів, засновані на оптимізації модулярності

Переважна більшість алгоритмів кластеризації графів заснована на оптимізації модулярності [192, 221].

Модулярність – деяка числова характеристика, яка описує вираженість структури кластерів в певному графі [132, 192, 221]. Для оцінки модулярності можна використовувати формулу:

$$Q = \frac{1}{2n_e}\sum_{ij}(A_{ij} - \frac{d_i d_j}{2n_e})\delta(C_i, C_j), \qquad (6.1)$$

де n_e – кількість ребер у графі; A – матриця суміжності графу; d_i – кількість ребер, суміжних з вершиною i; d_j – кількість ребер, суміжних з вершиною j; $\delta(C_i, C_j)$ – дельта-функція, рівна одиниці, якщо $C_i = C_j$ та нулю в іншому випадку.

Дана величина дорівнює різниці між кількістю ребер всередині кластера при поточному розбитті і кількістю ребер, якби вони були випадково згенеровані [221].

Значення модулярності показує вираженість кластерів, вона буде:

– рівна одиниці для повного графу, в якому всі вершини помістили в один кластер;

– рівна нулю для розбиття на кластери, при якому кожна вершина знаходиться в окремому кластері;

– для невдалих розбиттів модулярність може приймати негативне значення.

По-суті, за допомогою значення модулярності можна оцінити якість розбиття графів на кластери. Якісне розбиття характеризується тим, що кількість внутрішніх зв’язків всередині кожного кластера має бути більша, ніж кількість його зовнішніх зв’язків.

Значення модулярності характеризує не те, наскільки для даного розбиття внутрішньо-кластерні зв'язки більш щільні, ніж міжкластерні, а те, наскільки вони більш щільні в порівнянні з деякою початковою щільністю. Тому відбувається порівняння з «нульовою гіпотезою», яка полягає у тому, що дуги розподілені випадково, тобто, немає закономірностей в розподілі щільності дуг всередині графу.

Принцип алгоритмів кластеризації графів, заснованих на оптимізації модулярності, полягає у тому, що на кожному кроці алгоритму кожній вершині графу деяким чином ставиться у відповідність деякий кластер, обчислюється значення модулярності та здійснюється перерозподіл вершин графу між кластерами таким чином, щоб збільшити значення модулярності. Робота таких алгоритмів припиняється тоді, коли вже не можна покращити значення модулярності.

Розглянемо деякі найбільш відомі методи кластеризації графів на основі оптимізації модулярності.

Метод FastGreedy – полягає в жадібній оптимізації модулярності [192, 199]. На першому кроці методу кожній вершині графу ставиться у відповідність окремий кластер, а потім об'єднуються такі пари кластерів, об’єднання яких призводить до максимального збільшення модулярності. При цьому об'єднуються тільки інциденті пари вершин, тому що інакше модулярність не може збільшитися. Ітерації об’єднання кластерів продовжуються поки продовжує збільшуватися значення модулярності після них.

Метод Louvain (або **Multilevel**) – заснований на багаторівневій оптимізації функції модулярності [66, 192]. Більш якісно розбиває граф на кластертери порівняно з попереднім методом. Даний метод складається з двох частин. Перша частина, по суті, ідентична методу FastGreedy. Друга частина методу полягає у наступному: створюється новий граф з метавершинами у вигляді знайдених кластерів і ребрами з сумарною вагою всіх ребер, що йдуть від одного кластера до іншого (також створюються петлі з сумарними вагами зв'язків всередині кластера). Такий граф називається метаграф. Алгоритм знову запускається на новому графі. Це один з найбільш широковідомих методів за рахунок своєї швидкості роботи.

Метод Leading Eigenvector – спектральний метод, заснований на власних векторах матриці модулярності [192], яка визначається наступним способом:

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2n_e}, \tag{6.2}$$

де A – матриця суміжності графу; d_i – кількість ребер, суміжних з вершиною i; d_j – кількість ребер, суміжних з вершиною j; n_e – кількість ребер у графі.

Для даної матриці знаходиться перший власний вектор (з максимальним власним числом). Ті вершини, у яких відповідне значення менше нуля, належать одному кластеру, а де більше нуля – іншому. Подібним чином можливий поділ на більшу кількість кластерів.

Дані методи, засновані на оптимізації модулярності, є досить ефективними, але не найпростішими у реалізації.

Методи кластеризації графів, засновані на розмітці графів

Для пошуку кластерів у графі можна використовувати різні методи розмітки та розфарбовування графів. В основі даних методів лежить ідея, що вершина належить до того кластеру, до якого належить найбільша кількість її сусідніх вершин. В даних методах дуже важливий порядок перебору вершин. Методи розмітки графу допомагають перебрати вершини певним чином та з врахуванням їх сусідства визначитися з їх приналежністю до певних кластерів.

Розглянемо найвідоміший метод кластеризації графів з даної групи методів – LabelPropagation.

Метод LabelPropagation – розбиває граф на кластери наступним чином: кожна вершина у графі відноситься до того кластеру, якому належить більшість її сусідів, якщо ж таких кластерів декілька, то вибирається випадково один з них [66, 192, 221].

Розглянемо принцип роботи даного методу. У початковий момент часу всім вершинам ставиться у відповідність окремий кластер. Кожна вершина одержує мітку або колір відповідного кластеру. Потім для кожної вершини перевіряється, до яких кластерів належать її сусіди та здійснюється перерозподіл приналежності до кластерів. Через випадковості важливо на кожній ітерації змінювати порядок обходу вершин. Алгоритм закінчує роботу, коли уже нема чого змінювати – всі вершини відносяться до тих кластерів, що і більшість їх сусідів. Для покращення результатів можна застосувати наступну хитрість – запустити алгоритм декілька разів, кожного разу зберігати результат його роботи та обрати найкращий варіант розбиття графу. Головна перевага даного алгоритму – майже лінійна складність. Недоліком алгоритму є те, що на зашумлених графах часто відбувається об'єднання всіх вершин в один кластер або невелику кількість кластерів.

Методи кластеризації графів, засновані на випадкових блуканнях

Для розбиття графу на кластери можна застосовувати алгоритми випадкового блукання.

Випадкове блукання – математична модель процесу випадкових змін – кроків в дискретні моменти часу. При цьому передбачається, що зміна на кожному кроці не залежить від попередніх змін і від часу.

Розглянемо найвідоміші методи кластеризації графів, засновані на випадкових блуканнях.

Метод Walktrap – використовує ідею про те, що короткі випадкові блукання не призводять до виходу з поточної спільноти (кластеру) [192, 199]. Відстань між вершинами або між групами вершин розраховують на основі ймовірності досяжності шляху від однієї вершини до іншої в процесі випадкового блукання. Даний показник є великим, якщо вершини знаходяться в різних кластерах в графі, і маленьким, якщо вони знаходяться в одному кластері. Далі здійснюється ієрархічна кластеризація агломеративним способом на основі методу Уорда: вершини об'єднуються у кластери на основі вибору найменшого середнього квадрата відстаней між ними. Після того, як вершина приєднана до якого-небудь кластеру, відстані між вершинами і кластерами перераховуються.

Метод Infomap – заснований на понятті інформаційних потоків в мережах, кодуванні і стисненні інформації [192]. В даному методі застосовується підхід, що базується на випадкових блуканнях та кодах Хаффмана. У кожної вершини є певна ймовірність її відвідування. За допомогою кодів Хаффмана, відповідно до цих ймовірностей, можна закодувати шлях блукання. Ця послідовність матиме деяку довжину. Якщо використовувати ієрархічне кодування, можна скоротити довжину послідовності. Infomap ґрунтується на жадібному способі мінімізації довжини коду блукання.

Методи розбиття графів на кластери, що перетинаються

Розглянуті вище методи дозволяють розбити граф на кластери, що не перетинаються. В той же час для рішення деяких практичних задач може виникнути необхідність розбити граф на кластери, що можуть перетинатися між собою. В контексті задачі пошуку бот-мереж серед профілів

користувачів рекомендаційної системи такі методи можуть знадобитися для виділення різних бот-мереж (серед множини профілів ботів), у яких є спільні та відмінні об’єкти у списках цілей для атак.

Розглянемо деякі методи, що дозволяють вирішити цю задачу.

Метод Clique perlocation method (CPM) – призначений для розбиття графу на кластери, що перетинаються, використовуючи особливості структури графу, а саме наявності *клік*. Починає роботу з пошуку всіх клік розміру l, після чого будується новий граф, вершинами якого є знайдені кліки [192, 199]. Ребро утворюється у разі, якщо перетин вершин-клік складається з $(l-1)$ вершин початкового графу. Компоненти зв'язності нового графу і будуть визначати знайдені кластери. Перевагою методу є його інтуїтивність та простота для розуміння. Недоліком методу є непридатність його використання на графах з дуже великою кількістю вершин.

Слід пояснити термін кліка у теорії графів.

Кліка – підмножина вершин неорієнтованого графу, будь-які дві з яких з'єднані ребром (рис. 6.1).

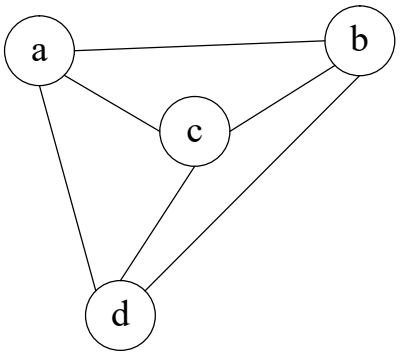


Рис. 6.1. Приклад структури типу *кліка* у графі

Метод Cluster Affiliation Model for Big Networks – ймовірнісна генеративна модель, що зводить задачу виділення кластерів до задачі факторизації невід’ємних матриць [192, 199]. Застосовується до дводольного графу, в одній частині якого знаходяться кластери, а в іншій – вершини, причому кожна вершина $v \in V$ не просто належить кластеру $c \in C$, а належить йому з деякою невід’ємною вагою. Задача оптимізації полягає у

визначенні методом максимізації правдоподібності для даного графу оптимальної матриці зв'язків приналежності до кластерів. Для того, щоб відповісти на питання належить, чи не належить певний об'єкт до певного кластеру, до зв'язків подоби застосовується відсікання за пороговим значенням.

Метод Democratic Estimate of the Modular Organization of a Network (DEMON) – полягає у тому, що для кожної вершини графу будується его-мережа, потім для кожної такої его-мережі окремо застосовується алгоритм Label Propagation, у результаті якого отримуються розбиття на кластери $C_i(u_i)$, для кожного користувача u_i [192]. Усі такі покриття потім об'єднуються у загальне покриття C, яке на початку ініціалізувалося порожнім. Під час такого об'єднання два кластери об'єднуються в один тільки в тому разі, якщо не більше ε відсотків меншого з них не міститься в більшому з них, наприклад, для $\varepsilon = 0$ об'єднання буде відбуватися тільки, коли один з кластерів повністю міститься в іншому, а для $\varepsilon = 1$ об'єднання буде відбуватися завжди.

Оскільки для реалізації та тестування розроблених у даній роботі методів синтезу рекомендаційних систем використовувалася СУБД Neo4j, доречним є дослідження уже готових вбудованих у неї інструментів для кластеризації графів.

Засоби для кластеризації графів у графовій СУБД Neo4j

Neo4j – це система управління базами даних типу NoSQL [52, 220], заснована на представлені даних у вигляді графів [52, 66, 220]. Вона має вбудовану бібліотеку Graph algorithms з розпаралеленими алгоритмами для роботи з графами. Для кластеризації графів дана бібліотека містить реалізації наступних алгоритмів [66]:

– **Louvain** (функція algo.louvain) – алгоритм кластеризації графів, заснований на оптимізації модулярності. Вузли об'єднуються у кластери так, щоб збільшити модулярність. Є одним з найшвидших алгоритмів на основі модулярності і добре працює з великими графами.

– **Label Propagation** (функція algo.labelPropagation) – кластеризує граф, використовуючи лише його структуру. Кожна вершина в графі поміщається в той кластер, якому належить більшість його сусідів. Якщо ж таких кластерів декілька, то вибирається випадково один з них. У початковий момент часу всім вершинам ставиться у відповідність окремий кластер.

– **Triangle Counting / Clustering Coefficient** (функція algo.triangleCount) – визначає кількість трикутників, що проходять через кожен вузол у графі. Трикутник являє собою набір з трьох вузлів, в якому кожен вузол має зв'яки з усіма іншими вузлами. На основі одержаних даних визначає коефіцієнт кластеризації. Хоча розробники СУБД Neo4j відносять даний алгоритм до алгоритмів кластеризації, слід зазначити, що це не зовсім коректно, адже знаходяться трикутники, а не кластери у графі, а трикутник лише частковий випадок кластеру.

Також серед реалізацій алгоритмів кластеризації графів у документації до бібліотеки Graph algorithms Neo4j [66] пропонуються до використання **Connected Components** та **Strongly Connected Components**, які знаходять зв'язані підграфи незв'язного графу для неорієнтованих та орієнтованих графів відповідно, та **Balanced Triads** – алгоритм оцінки структурного балансу графу соціальної мережі, що знаходить збалансовані та незбалансовані тріади у мережі. Оскільки ці алгоритми не розбивають граф на кластери, а виконують трохи інші функції, їх тестування у рамках даного дослідження не проводилося.

Для застосування інших методів кластеризації графів всередині СУБД Neo4j необхідно розробляти власні функції, що можна зробити з використанням її власної мови запитів Cypher.

Було здійснене дослідження існуючих засобів кластеризації графів СУБД Neo4j шляхом тестування їх на згенерованому випадковим чином соціальному графі [144, 152, 158].

Для роботи з СУБД Neo4j була використана бібліотека neo4j.v1 для мови Python [66, 161].

Для підключення до Neo4j використовувався наступний код:

```
    driver = GraphDatabase.driver
("bolt://localhost", auth=basic_auth(user = <login>, password = <password>))
    session = driver.session()
```

Для здійснення запитів до бази даних використовувалася функція:

```
session.run("""<запити_до_бази даних>""", [<змінні_через_кому>])
```

Запис у базу даних інформації можна здійснити за допомогою наступних запитів:

```
//створення вузлів
MERGE (U1:User {id:$id1, name:$name1})
MERGE (U2:User {id:$id2, name:$name2})
// створення ребер
MERGE (U1)-[:Friend]->(U2)
```

Дані в Neo4j зберігаються у вигляді показаному на рис. 6.2.

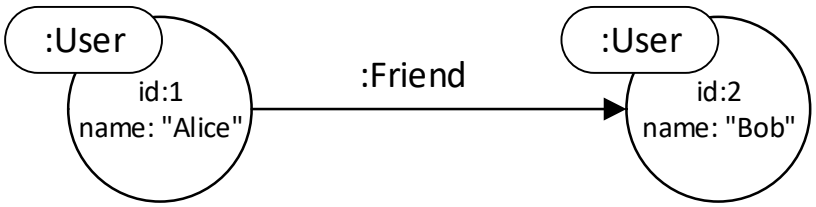


Рис. 6.2. Формат запису даних у СУБД Neo4j

Для тестування алгоритмів кластеризації графів було згенеровано випадковий граф з властивостями складної мережі.

Наведемо приклад запиту для виділення співтовариств методом Louvain:

```
    CALL algo.louvain('User', 'Friend',  {write:true, writeProperty:
'community'})
    YIELD nodes, communityCount, iterations, loadMillis, computeMillis,
writeMillis;
```

Після виконання даного запиту у кожної вершини графу з міткою :User з’явиться властивість community, що буде містити номер кластеру, до якого метод Louvain віднесе відповідну вершину.

Результати роботи даного методу для графу з 2348 вузлами (користувачами) та 10762 зв’язками (відношенями «друзі») наведені у табл. 6.1.

Таблиця 6.1. Приклад результатів виклику функції Louvain для графу з 2348 вузлами та 10762 зв’язками

Кількість кластерів	**Кількість ітерацій**	**Час завантаження даних, мс**	**Час роботи алгоритму, мс**	**Час запису результатів у граф, мс**
11	2	18	34	93

Запит до бази даних для виконання алгоритму кластеризації Label Propagation:

```
CALL algo.labelPropagation('User', 'Friend','BOTH', {iterations:10, partitionProperty:'partition', write:true})
YIELD nodes, iterations, loadMillis, computeMillis, writeMillis, write, partitionProperty;
```

Після виконання даного запиту у кожної вершини графу з міткою :User з’явиться властивість partition, що буде містити номер кластеру, до якого метод Label Propagation віднесе відповідну вершину.

Результати роботи даного методу для графу з 2348 вузлами (користувачами) та 10762 зв’язками (відношенями «друзі») наведені у табл. 6.2.

Таблиця 6.2. Приклад результатів виклику функції labelPropagation для графу з 2348 вузлами та 10762 зв’язками

Кількість кластерів	**Кількість ітерацій**	**Час завантаження даних, мс**	**Час запуску алгоритму, мс**	**Час запису результатів у граф, мс**
3	3	25	1	3

Запит до бази даних для виконання алгоритму кластеризації Triangle Counting:

```
CALL algo.triangleCount('User', 'Friend', {concurrency:4, write:true, writeProperty:'triangles',clusteringCoefficientProperty:'coefficient'})
YIELD loadMillis, computeMillis, writeMillis, nodeCount, triangleCount, averageClusteringCoefficient;
```

Після виконання даного запиту у кожної вершини графу з міткою :User з'явиться властивість triangles, що буде містити номер трикутника, у якому знаходиться вершина.

Результати роботи даного методу для графу з 2348 вузлами (користувачами) та 10762 зв'язками (відношенями «друзі») наведені у табл. 6.3.

Таблиця 6.3. Приклад результатів виклику функції triangleCount для графу з 2348 вузлами та 10762 зв'язками

Кількість трикутників	**Середній коефіцієнт кластеризації**	**Час завантаження даних, мс**	**Час роботи алгоритму, мс**	**Час запису результатів у граф, мс**
10630	0.1756	18	3	7

У результаті проведеного дослідження можна зробити висновок, що для пошуку профілів ботів у рекомендаційних системах слід застосовувати методи кластеризації графів, що розбивають граф на кластери, які не перетинаються. Серед даних методів найбільш поширеними є методи Louvain та Label Propagation, які навіть реалізовані у графовій СУБД Neo4j як готові інструменти. Тестування відповідних алгоритмів у Neo4j показало, що загальний час роботи Label Propagation менший, ніж у Louvain. А саме, сумарний час роботи алгоритму (який визначався сумою часу завантаження даних, часу роботи алгоритму та часу запису результатів у граф) на розглянутому прикладі для Louvain становив 145 мс, а для Label Propagation – 29 мс. Також було виявлено суттєву різницю при розбитті на кластери одного

і того самого графу, Louvain розбив його на 11 кластерів, а Label Propagation на 3 кластери. Тому можна зробити висновок, що вибір конкретного алгоритму буде залежати від властивостей вхідних даних, і його можна зробити, тестуючи вірність розбиття на кластери на навчаючих вибірках.

В подальших дослідженнях у даній роботі було вирішено використовувати метод Label Propagation, так як він показав більшу швидкість роботи та розбив дані на меншу кількість кластерів, а в задачі пошуку ботів усі профілі користувачів треба розділити всього на два кластери – боти та аутентичні користувачі, тобто, немає потреби створювати багато невеликих кластерів на основі незначних розбіжностей в даних профілів.

6.1.2. Розробка методу виявлення та нейтралізації мережі ботів у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів

У попередньому розділі було запропоновано метод визначення множини ймовірних цілей інформаційної атаки мережі ботів за допомогою аналізу трендів рейтингів об’єктів рекомендаційної системи. Позначимо таку множину можливих цілей атаки G.

Після того, як виявлена атака на рекомендаційну систему та сформована множина ймовірних цілей ботів G, логічним буде дослідити профілі всіх користувачів, які вплинули на зміну трендів рейтингів об’єктів з цієї множини G, що і пропонується зробити у даній роботі для пошуку серед них профілів ботів.

Для вирішення задачі ідентифікації профілів ботів було розроблено метод заснований на графовій кластеризації та аналізі дій користувачів, з використанням коефіцієнтів «недовіри», що складається з наступних етапів:

Етап 1. Формуємо множину підозрілих профілів користувачів S, в яку поміщаємо профілі, що ставили цільові оцінки r_t об’єктам з множини G.

Етап 2. Привласнюємо кожному користувачу з множини G мітку *:suspicious* та коефіцієнт недовіри, розрахований за наступною формулою:

$$k_{d,i} = \sum_{j \in G} \frac{E_{r_{t,i,j}}}{n_g}, \tag{6.3}$$

де $E_{r_{t,i,j}}$ – наявність цільової оцінки r_t від користувача i об'єкту j, що належить множині ймовірних цілей атаки G, приймає значення 1, якщо цільова оцінка є та 0 – при відсутності такої оцінки від користувача i об'єкту j; n_g – кількість об'єктів у множині можливих цілей атаки G.

Етап 3. Для кожної пари користувачів i_1 та i_2 з множини S, де $k_{d,i1} \geq q$ та $k_{d,i2} \geq q$, створюємо ребро між ними з міткою *:BotNet.*

Етап 4. Виконуємо графову кластеризацію для підграфу, що містить вершини з мітками *:User* і *:suspicious* та ребра з міткою *:BotNet.* Результати роботи такої кластеризації будуть наступними – усі боти потраплять до одного великого кластеру, якщо бот-мережа одна, або до декількох великих кластерів – якщо бот-мереж декілька; аутентичні користувачі потраплять у різні кластери, кожний з таких кластерів буде містити одного користувача або невелику кількість користувачів. Можливі й випадки, коли до кластеру з ботами потрапить деяка кількість аутентичних користувачів або деяка група схожих між собою та активних аутентичних користувачів утворить окремий кластер у разі, якщо їх дії зсунули рейтинги деяких об'єктів.

Етап 5. Визначаємо найбільші кластери, що складаються з $(N_{cr} - e)$ користувачів, де N_{cr} – мінімальна кількість користувачів, що може вплинути на результати роботи рекомендаційної системи (залежить від параметрів конкретної системи), e – приблизне значення похибки при розділенні профілів користувачів на кластери. Такий кластер (або кластери) вважаємо можливою бот-мережею (бот-мережами). У користувачів, що не потрапляють до даних кластерів прибираємо ребра з міткою *:BotNet*. Користувачів, що потрапили до підграфу *BotNet* треба додатково перевірити, проаналізувавши статистичні характеристики їх профілів, наприклад, за допомогою

запропонованого у попередньому розділі способу з використанням нейронних мереж [56]. Також для додаткової перевірки профілів з підграфу *BotNet* можна здійснити пошук у них певних характерних для ботів ознак, наприклад, погано заповнених анкетних даних або надзвичайно високої активності (характерні особливості ботів залежать від конкретної системи і можуть ставати відомими в процесі збору статистичних даних під час її роботи). Одними з загальних ознак ботів для багатьох систем можуть бути: відмінність значення дисперсії оцінок та дисперсії часових інтервалів між виставленнями оцінок у профілях ботів від середньостатистичних значень відповідних дисперсій у профілях користувачів системи. Для конкретної системи ознаками ботів можуть бути: особливості реєстрації профілю, особливості наповнення профілю особистою інформацією, стиль написання та зміст коментарів, список друзів користувача тощо. Після перевірки статистичних даних окремих профілів користувачів, які потрапили у підграф *BotNet*, треба його скоректувати, видаливши з нього користувачів, розпізнаних за статистичними даними як аутентичні. Якщо є кластер, у якому всі користувачі визнані автентичними – слід перестати вважати його бот-мережею.

Етап 6. Коректуємо множину G після аналізу оцінок користувачів з підграфу *BotNet*. Слід перевірити, яким об'єктам користувачі з бот-мережі скоординовано виставляли цільові оцінки. Видаляємо з G об'єкти, які не одержували взагалі або одержали незначний процент цільових оцінок від користувачів ідентифікованих як боти. Додаємо до множини G об'єкти, які одержали цільові оцінки від усіх ботів (або великого проценту ботів).

Було проведено серію експериментів для тестування розробленого методу. Набори даних для експериментів генерувалися у розробленій програмній імітаційній моделі рекомендаційної системи [53]. Формат та статистичні особливості даних генерувалися максимально наближеними до відповідних характеристик відкритого набору даних MovieLens Datasets [30]. Інформаційні атаки моделювалися за допомогою популярної моделі атаки

[28, 40, 76]. У якості алгоритму графової кластеризації використовувався алгоритм Label Propagation.

У проведеній серії експериментів було згенеровано 10% ботів, усі інші користувачі системи – аутентичні. У різних експериментах у ботів була різна кількість цілей для атаки: 1, 5, 10, 15, 20, 25, 30, 35, 40 та 45 цілей. Також серед аутентичних користувачів було згенеровано 20% профілів з високим рівнем активності у рекомендаційній системі, які виставляли значно більше оцінок, ніж середньостатистичні користувачі. Об'єкти системи були згенеровані таким чином, щоб відноситися за своїми властивостями до одного з 19 кластерів.

Алгоритм пошуку ботів для досліджуваної рекомендаційної системи мав наступні параметри:

– $q = 0.05$ – тобто, користувач виставив цільові оцінки не менше 5% об'єктів, які визначені підсистемою інформаційної безпеки як ймовірні цілі атаки. Порогове значення невелике, тому що невідомо наскільки вірно розпізнані цілі атаки, можливо там багато об'єктів, у яких змінилися рейтинги природнім чином.

– Відсіювання профілів користувачів з підграфу *BotNet* здійснюється на основі значень дисперсії оцінок та дисперсії часових інтервалів між виставленням оцінок. Аутентичними користувачами, які помилково потрапили до підграфу *BotNet*, вважаються такі, дані з профілів яких відповідають наступному правилу:

$$|D_r - D_{r,avr}| < 0.15 \text{ AND } (D_{t,r} > 72 \text{ AND } D_{t,r} < 600),$$

де D_r – дисперсія оцінок у профілі користувача, $D_{r,avr}$ – усічене середнє дисперсії оцінок у профілі користувачів системи (відсікалося 30% крайніх значень, тобто, гранично можливий процент ботів у розглядуваній системі), $D_{t,r}$ – дисперсія часових інтервалів між виставленнями оцінок. Тобто, дисперсія оцінок у профілях аутентичних користувачів несуттєво відрізняється від середньостатистичної дисперсії (атакуючий систему це значення знати не може, він може лише приблизно його оцінити, тому не

може вірно відтворити дану характеристику при створенні профілів ботів). А перевірка дисперсії часових інтервалів між виставленнями оцінок має наступний сенс – слід вважати підозрілими користувачів, які роблять занадто однакові або занадто різні інтервали між виставленнями оцінок, в той же час дисперсія часових проміжків між виставленнями оцінок у автентичних користувачів знаходяться, як правило, в певному діапазоні значень.

Для оцінки якості роботи розробленого методу було обрано наступні метрики:

– *Точність* (*Precision*) – міра, що характеризує, скільки позитивних спрацьовувань підсистеми інформаційної безпеки відносно ідентифікації профілів ботів є правильними. Вона обчислювалася за формулою:

$$precision = \frac{tp}{tp + fp}, \tag{6.4}$$

де *tp* – правильно виявлений бот; *fp* – аутентичний користувач невірно ідентифікований як бот.

– *Повнота* (*Recall, Sensitivity*) – міра, що характеризує можливість підсистеми інформаційної безпеки правильно ідентифікувати профілі ботів, при цьому не враховуються невірні позитивні спрацьовування. Визначалася за формулою:

$$recall = \frac{tp}{tp + fn}, \tag{6.5}$$

де *tp* – правильно виявлений бот; *fn* – бот, який невірно ідентифікований як аутентичний користувач.

– *RMSE* – середньоквадратична помилка:

$$RMSE = \sum_{i=1}^{n} \frac{\left(\hat{y}_i - y_i\right)}{n}, \tag{6.6}$$

В таблицях 6.4-6.5 наведено результати серії експериментів для перевірки точності роботи розробленого методу ідентифікації профілів ботів у рекомендаційній системі. В них використані наступні позначення моделей атаки: RA – випадкова атака, AA – середня атака, PA – популярна атака.

Таблиця 6.4. Результати тестування точності роботи розробленого методу ідентифікації профілів ботів у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів для різних моделей атак та різної кількості ботів і цілей атаки

№ експ.	Частка ботів серед профілів системи, %	Кількість цілей у кожного бота, шт.	Тип атаки	Кількість вірно розпізнаних цілей, %	Кількість невірно розпізнаних як цілі, %	Точність розпізнавання ботів	Повнота розпізнавання ботів	Частка помилок першого роду, %	Частка помилок другого роду, %	RMSE для розпізнавання ботів
1	5	1	RA	100.000	24.1206	0.666666	0.800000	2.1052	20.0000	0.173205
2	5	5	RA	40.0000	22.0513	0.571428	0.800000	3.1578	20.0000	0.200000
3	5	10	RA	60.0000	21.0526	0.800000	0.800000	1.0526	20.0000	0.141421
4	10	1	RA	0.00000	22.1106	0.375000	0.600000	5.2631	40.0000	0.264575
5	10	5	RA	60.0000	24.1025	0.538461	0.700000	6.6666	30.0000	0.300000
6	10	10	RA	90.0000	21.5789	0.888888	0.800000	1.1111	20.0000	0.173205
7	20	1	RA	100.000	30.6532	0.692307	0.900000	4.4444	10.0000	0.223606
8	20	5	RA	100.000	23.5897	0.700000	0.700000	7.5000	30.0000	0.346410
9	20	10	RA	80.0000	18.9473	0.850000	0.850000	3.7500	15.0000	0.244948
10	30	1	RA	100.000	33.6683	0.823529	0.700000	3.7500	30.0000	0.300000
11	30	5	RA	80.0000	29.7435	0.896551	0.866666	4.2857	13.3333	0.264575
12	30	10	RA	80.0000	23.6842	0.884615	0.766666	4.2857	23.3333	0.316227
Середні значення:				**74.1666**	**24.6085**	**0.723953**	**0.773611**	**3.9476**	**22.6388**	**0.245681**
13	5	1	AA	100.000	18.0905	0.625000	1.000000	0.0000	3.2608	0.173205
14	5	5	AA	100.000	21.0257	0.555555	1.000000	0.0000	4.3956	0.200000
15	5	10	AA	60.0000	21.5789	1.000000	1.000000	0.0000	0.0000	0.000000
16	10	1	AA	100.000	24.6231	0.588235	1.000000	0.0000	7.7777	0.264575
17	10	5	AA	60.0000	25.6410	0.900000	0.900000	1.1111	10.000	0.141421
18	10	10	AA	70.0000	25.2631	0.833333	1.000000	2.2222	0.0000	0.141421
19	20	1	AA	100.000	25.1256	0.740740	1.000000	8.7500	0.0000	0.264575
20	20	5	AA	100.000	22.0512	0.850000	0.850000	3.7500	15.000	0.244948
21	20	10	AA	60.0000	13.6842	0.900000	0.900000	2.5000	10.000	0.200000
22	30	1	AA	60.0000	14.5728	0.909090	1.000000	4.2857	0.0000	0.173205
23	30	5	AA	100.000	17.9487	0.928571	0.866666	2.8571	13.3333	0.244948
24	30	10	AA	60.0000	11.5789	0.925925	0.833333	2.8571	16.6666	0.264575
Середні значення:				**80.8333**	**20.0986**	**0.813037**	**0.945833**	**2.3611**	**6.7028**	**0.192739**
25	5	1	PA	75.8793	24.1206	0.714285	1.000000	2.1052	0.000000	0.141421
26	5	5	PA	40.0000	17.9488	0.625000	1.000000	3.1578	0.000000	0.173205
27	5	10	PA	60.0000	22.6315	0.416666	1.000000	7.3684	0.000000	0.264575
28	10	1	PA	100.000	22.1105	0.666666	1.000000	5.5555	0.000000	0.223606
29	10	5	PA	80.0000	30.2564	0.900000	0.900000	1.1111	0.10.0000	0.141421
30	10	10	PA	90.0000	19.4736	0.888888	0.800000	1.1111	0.20.0000	0.173205
31	20	1	PA	0.00000	30.1507	0.000000	0.000000	6.2500	1.00.0000	0.500000
32	20	5	PA	100.000	14.3589	0.800000	0.600000	3.7500	0.40.0000	0.331662
33	20	10	PA	100.000	17.8947	0.869565	1.000000	3.7500	0.000000	0.173205
34	30	1	PA	100.000	28.6432	0.909090	1.000000	4.2857	0.000000	0.173205
35	30	5	PA	100.000	25.6410	0.900000	0.600000	2.8571	0.40.0000	0.374165
36	30	10	PA	100.000	17.3684	0.892857	0.833333	4.2857	0.16.6666	0.282842
Середні значення:				**78.8232**	**22.5498**	**0.715251**	**0.811111**	**3.7990**	**0.188889**	**0.246043**

Як показала серія експериментів з табл. 6.4, точність розпізнавання ботів розробленим методом в середньому становить 0.72 для випадкової атаки, 0.81 для середньої атаки, а також 0.71 для популярної атаки.

Також було більш детально досліджено точність розпізнавання ботів розробленим методом для популярної моделі атаки (табл. 6.5).

Таблиця 6.5. Результати тестування точності роботи розробленого методу ідентифікації профілів ботів у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів для популярної моделі атаки та різної кількості цілей атаки

№ експ.	Частка ботів серед профілів системи, %	Кількість цілей у кожного бота, шт.	Тип атаки	Точність (Precision) розпізнавання ботів	Повнота (Recall) розпізнавання ботів	RMSE для класифікації користувачів на ботів та аутентичних
1.	10	1	PA	0.5882352941176471	1.0000000000000000	0.26457513110645910
2.	10	1	PA	0.0777777777777777	1.0000000000000000	0.26457513110645910
3.	10	1	PA	0.2500000000000000	0.1000000000000000	0.34641016151377540
4.	10	1	PA	0.9090909090909091	1.0000000000000000	0.10000000000000000
5.	10	1	PA	0.8333333333333334	1.0000000000000000	0.14142135623730950
6.	10	1	PA	0.9090909090909091	1.0000000000000000	0.10000000000000000
7.	10	1	PA	1.0000000000000000	1.0000000000000000	0.00000000000000000
8.	10	1	PA	0.6153846153846154	0.8000000000000000	0.26457513110645910
9.	10	1	PA	0.0000000000000000	0.0000000000000000	0.33166247903553990
10.	10	1	PA	0.6666666666666666	1.0000000000000000	0.22360679774997890
11.	10	1	PA	0.0000000000000000	0.0000000000000000	0.48989794855663560
12.	10	1	PA	0.7500000000000000	0.9000000000000000	0.20000000000000000
13.	10	1	PA	0.0000000000000000	0.0000000000000000	0.36055512754639890
14.	10	1	PA	0.6666666666666666	1.0000000000000000	0.22360679774997890
15.	10	1	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
16.	10	1	PA	0.7777777777777778	0.7000000000000000	0.22360679774997890
17.	10	1	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
18.	10	1	PA	0.7692307692307693	1.0000000000000000	0.17320508075688770
19.	10	1	PA	0.3333333333333333	1.0000000000000000	0.44721359549995790
20.	10	1	PA	0.4761904761904761	1.0000000000000000	0.33166247903553990
Середнє усічене (20%):				**0.5713911910000000**	**0.8437500000000000**	**0.24059140400000000**
21.	10	5	PA	0.5833333333333334	0.7000000000000000	0.28284271247461900
22.	10	5	PA	0.9090909090909091	1.0000000000000000	1.00000000000000000
23.	10	5	PA	0.7692307692307693	1.0000000000000000	0.17320508075688773
24.	10	5	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
25.	10	5	PA	0.4545454545454545	1.0000000000000000	0.34641016151377546
26.	10	5	PA	0.7000000000000000	0.7000000000000000	0.24494897427831780
27.	10	5	PA	0.7692307692307693	1.0000000000000000	0.17320508075688773
28.	10	5	PA	0.6250000000000000	1.0000000000000000	0.24494897427831780
29.	10	5	PA	0.6666666666666666	1.0000000000000000	0.22360679774997896
30.	10	5	PA	0.6250000000000000	1.0000000000000000	0.24494897427831780
31.	10	5	PA	0.7500000000000000	0.9000000000000000	0.20000000000000000
32.	10	5	PA	0.9090909090909091	1.0000000000000000	0.10000000000000000
33.	10	5	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
34.	10	5	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
35.	10	5	PA	0.6666666666666666	0.8000000000000000	0.24494897427831780
36.	10	5	PA	0.8750000000000000	0.7000000000000000	0.20000000000000000
37.	10	5	PA	1.0000000000000000	0.9000000000000000	0.10000000000000000
38.	10	5	PA	0.8333333333333334	1.0000000000000000	0.14142135623730950
39.	10	5	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
40.	10	5	PA	0.5833333333333334	0.7000000000000000	0.28284271247461900
Середнє усічене (20%):				**0.7500702420000000**	**0.9250000000000000**	**0.20873514700000000**
41.	10	10	PA	0.6666666666666666	0.8000000000000000	0.24494897427831780
42.	10	10	PA	0.7692307692307693	1.0000000000000000	0.17320508075688773
43.	10	10	PA	0.6428571428571429	0.9000000000000000	0.24494897427831780
44.	10	10	PA	0.8888888888888888	0.8000000000000000	0.17320508075688773
45.	10	10	PA	0.6428571428571429	0.9000000000000000	0.26457513110645910

№ експ.	Частка ботів серед профілів системи, %	Кількість цілей у кожного бота, шт.	Тип атаки	Точність (Precision) розпізнавання ботів	Повнота (Recall) розпізнавання ботів	RMSE для класифікації користувачів на ботів та аутентичних
46.	10	10	PA	0.8181818181818182	0.9000000000000000	0.17320508075688773
47.	10	10	PA	0.9090909090909091	1.0000000000000000	1.00000000000000000
48.	10	10	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
49.	10	10	PA	0.8000000000000000	0.8000000000000000	0.20000000000000000
50.	10	10	PA	0.6666666666666666	1.0000000000000000	0.22360679774997896
51.	10	10	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
52.	10	10	PA	0.5625000000000000	0.9000000000000000	0.28284271247461900
53.	10	10	PA	0.6250000000000000	1.0000000000000000	0.24494897427831780
54.	10	10	PA	0.8333333333333334	1.0000000000000000	0.14142135623730950
55.	10	10	PA	0.7692307692307693	1.0000000000000000	0.17320508075688773
56.	10	10	PA	0.7777777777777778	0.7000000000000000	0.22360679774997896
57.	10	10	PA	0.6666666666666666	1.0000000000000000	0.22360679774997896
58.	10	10	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
59.	10	10	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
60.	10	10	PA	0.8333333333333334	1.0000000000000000	0.14142135623730950
Середнє усічене (20%):				**0.7516889970000000**	**0.9375000000000000**	**0.20800568300000000**
61.	10	15	PA	0.9090909090909091	1.0000000000000000	0.10000000000000000
62.	10	15	PA	0.6666666666666666	1.0000000000000000	0.22360679774997896
63.	10	15	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
64.	10	15	PA	0.8181818181818182	0.9000000000000000	0.17320508075688773
65.	10	15	PA	0.6250000000000000	1.0000000000000000	0.24494897427831780
66.	10	15	PA	0.7272727272727273	0.8000000000000000	0.22360679774997896
67.	10	15	PA	0.7692307692307693	1.0000000000000000	0.17320508075688773
68.	10	15	PA	0.8181818181818182	0.9000000000000000	0.17320508075688773
69.	10	15	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
70.	10	15	PA	0.6666666666666666	1.0000000000000000	0.22360679774997896
71.	10	15	PA	0.7692307692307693	1.0000000000000000	0.17320508075688773
72.	10	15	PA	0.6363636363636364	0.7000000000000000	0.26457513110645910
73.	10	15	PA	0.9090909090909091	1.0000000000000000	0.10000000000000000
74.	10	15	PA	0.6000000000000000	0.9000000000000000	0.26457513110645910
75.	10	15	PA	0.7272727272727273	0.8000000000000000	0.22360679774997896
76.	10	15	PA	1.0000000000000000	1.0000000000000000	0.00000000000000000
77.	10	15	PA	0.4666666666666667	0.7000000000000000	0.33166247903553997
78.	10	15	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
79.	10	15	PA	0.8333333333333334	1.0000000000000000	0.14142135623730950
80.	10	15	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
Середнє усічене (20%):				**0.7573135200000000**	**0.9500000000000000**	**0.19443297400000000**
81.	10	20	PA	0.6666666666666666	1.0000000000000000	0.22360679774997896
82.	10	20	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
83.	10	20	PA	0.5882352941176471	1.0000000000000000	0.26457513110645910
84.	10	20	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
85.	10	20	PA	0.8333333333333334	1.0000000000000000	0.14142135623730950
86.	10	20	PA	0.8750000000000000	0.7000000000000000	0.20000000000000000
87.	10	20	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
88.	10	20	PA	0.8888888888888888	0.8000000000000000	0.17320508075688773
89.	10	20	PA	0.9090909090909091	1.0000000000000000	1.00000000000000000
90.	10	20	PA	0.7500000000000000	0.9000000000000000	0.20000000000000000
91.	10	20	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
92.	10	20	PA	0.7272727272727273	0.8000000000000000	0.22360679774997896
93.	10	20	PA	0.7500000000000000	0.9000000000000000	0.20000000000000000
94.	10	20	PA	0.9090909090909091	1.0000000000000000	0.10000000000000000
95.	10	20	PA	0.6428571428571429	0.9000000000000000	0.24494897427831780
96.	10	20	PA	0.8181818181818182	0.9000000000000000	0.17320508075688773
97.	10	20	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
98.	10	20	PA	0.8181818181818182	0.9000000000000000	0.17320508075688773

№ експ.	Частка ботів серед профілів системи, %	Кількість цілей у кожного бота, шт.	Тип атаки	Точність (Precision) розпізнавання ботів	Повнота (Recall) розпізнавання ботів	RMSE для класифікації користувачів на ботів та аутентичних
99.	10	20	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
100.	10	20	PA	0.6363636363636364	0.7000000000000000	0.26457513110645910
Середнє усічене (20%):				**0.7594442020000000**	**0.9312500000000000**	**0.19176558000000000**
101.	10	25	PA	0.7692307692307693	1.0000000000000000	0.17320508075688773
102.	10	25	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
103.	10	25	PA	1.0000000000000000	0.9000000000000000	0.10000000000000000
104.	10	25	PA	0.6250000000000000	1.0000000000000000	0.24494897427831780
105.	10	25	PA	0.9090909090909091	1.0000000000000000	1.00000000000000000
106.	10	25	PA	1.0000000000000000	1.0000000000000000	0.00000000000000000
107.	10	25	PA	0.7500000000000000	0.9000000000000000	0.20000000000000000
108.	10	25	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
109.	10	25	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
110.	10	25	PA	0.6666666666666666	0.8000000000000000	0.24494897427831780
111.	10	25	PA	0.9090909090909091	1.0000000000000000	0.10000000000000000
112.	10	25	PA	0.8181818181818182	0.9000000000000000	0.17320508075688773
113.	10	25	PA	0.8000000000000000	0.8000000000000000	0.20000000000000000
114.	10	25	PA	0.8333333333333334	1.0000000000000000	0.14142135623730950
115.	10	25	PA	0.0333333333333333	0.1000000000000000	0.20000000000000000
116.	10	25	PA	0.8181818181818182	0.9000000000000000	0.17320508075688773
117.	10	25	PA	0.6666666666666666	1.0000000000000000	0.22360679774997896
118.	10	25	PA	0.8888888888888888	0.8000000000000000	0.17320508075688773
119.	10	25	PA	0.7272727272727273	0.8000000000000000	0.22360679774997896
120.	10	25	PA	0.5000000000000000	1.0000000000000000	0.31622776601683794
Середнє усічене (20%):				**0.7805123700000000**	**0.9250000000000000**	**0.18977383600000000**
121.	10	30	PA	0.6428571428571429	0.9000000000000000	0.24494897427831780
122.	10	30	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
123.	10	30	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
124.	10	30	PA	0.8888888888888888	0.8000000000000000	0.17320508075688773
125.	10	30	PA	0.6428571428571429	0.9000000000000000	0.24494897427831780
126.	10	30	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
127.	10	30	PA	0.7142857142857143	1.0000000000000000	0.20000000000000000
128.	10	30	PA	0.7692307692307693	1.0000000000000000	0.17320508075688773
129.	10	30	PA	0.6428571428571429	0.9000000000000000	0.24494897427831780
130.	10	30	PA	1.0000000000000000	0.9000000000000000	0.10000000000000000
131.	10	30	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
132.	10	30	PA	1.0000000000000000	0.9000000000000000	0.10000000000000000
133.	10	30	PA	0.9090909090909091	1.0000000000000000	0.10000000000000000
134.	10	30	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
135.	10	30	PA	1.0000000000000000	0.8000000000000000	0.14142135623730950
136.	10	30	PA	0.8888888888888888	0.8000000000000000	0.17320508075688773
137.	10	30	PA	0.5000000000000000	0.3000000000000000	0.31622776601683794
138.	10	30	PA	0.6363636363636364	0.7000000000000000	0.26457513110645910
139.	10	30	PA	0.7777777777777778	0.7000000000000000	0.22360679774997896
140.	10	30	PA	0.7272727272727273	0.8000000000000000	0.22360679774997896
Середнє усічене (20%):				**0.7814317630000000**	**0.8750000000000000**	**0.19207208900000000**
141.	10	35	PA	0.8333333333333334	1.0000000000000000	0.14142135623730950
142.	10	35	PA	0.7142857142857143	0.5000000000000000	0.26457513110645910
143.	10	35	PA	0.6000000000000000	0.9000000000000000	0.26457513110645910
144.	10	35	PA	0.6250000000000000	0.5000000000000000	0.28284271247461900
145.	10	35	PA	0.8000000000000000	0.8000000000000000	0.20000000000000000
146.	10	35	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
147.	10	35	PA	0.7500000000000000	0.6000000000000000	0.24494897427831780
148.	10	35	PA	0.5714285714285714	0.4000000000000000	0.30000000000000000
149.	10	35	PA	0.5000000000000000	0.5000000000000000	0.31622776601683794
150.	10	35	PA	0.3750000000000000	0.6000000000000000	0.37416573867739417

№ експ.	Частка ботів серед профілів системи, %	Кількість цілей у кожного бота, шт.	Тип атаки	Точність (Precision) розпізнавання ботів	Повнота (Recall) розпізнавання ботів	RMSE для класифікації користувачів на ботів та аутентичних
151.	10	35	PA	0.7272727272727273	0.8000000000000000	0.22360679774997896
152.	10	35	PA	0.4444444444444444	0.4000000000000000	0.33166247903553997
153.	10	35	PA	0.8000000000000000	0.8000000000000000	0.20000000000000000
154.	10	35	PA	0.7500000000000000	0.9000000000000000	0.20000000000000000
155.	10	35	PA	0.9000000000000000	0.9000000000000000	0.14142135623730950
156.	10	35	PA	0.7272727272727273	0.8000000000000000	0.22360679774997896
157.	10	35	PA	0.5833333333333334	0.7000000000000000	0.28284271247461900
158.	10	35	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
159.	10	35	PA	0.5833333333333334	0.7000000000000000	0.28284271247461900
160.	10	35	PA	0.6923076923076923	0.9000000000000000	0.22360679774997896
Середнє усічене (20%):				**0.6843671950000000**	**0.7375000000000000**	**0.24216898000000000**
161.	10	40	PA	0.7500000000000000	0.6000000000000000	0.24494897427831780
162.	10	40	PA	0.7142857142857143	0.5000000000000000	0.26457513110645910
163.	10	40	PA	0.7500000000000000	0.3000000000000000	0.28284271247461900
164.	10	40	PA	0.6250000000000000	0.5000000000000000	0.28284271247461900
165.	10	40	PA	0.7500000000000000	0.6000000000000000	0.24494897427831780
166.	10	40	PA	0.6666666666666666	0.6000000000000000	0.26457513110645910
167.	10	40	PA	0.6000000000000000	0.6000000000000000	0.28284271247461900
168.	10	40	PA	0.8181818181818182	0.9000000000000000	0.17320508075688773
169.	10	40	PA	0.5000000000000000	0.4000000000000000	0.31622776601683794
170.	10	40	PA	0.6666666666666666	0.4000000000000000	0.28284271247461900
171.	10	40	PA	0.5555555555555556	0.5000000000000000	0.30000000000000000
172.	10	40	PA	0.7142857142857143	0.5000000000000000	0.26457513110645910
173.	10	40	PA	0.7500000000000000	0.6000000000000000	0.24494897427831780
174.	10	40	PA	0.4666666666666667	0.7000000000000000	0.33166247903553997
175.	10	40	PA	1.0000000000000000	0.9000000000000000	0.10000000000000000
176.	10	40	PA	0.7272727272727273	0.8000000000000000	0.22360679774997896
177.	10	40	PA	0.4444444444444444	0.4000000000000000	0.33166247903553997
178.	10	40	PA	0.6000000000000000	0.6000000000000000	0.28284271247461900
179.	10	40	PA	0.6428571428571429	0.9000000000000000	0.24494897427831780
180.	10	40	PA	0.8333333333333334	0.5000000000000000	0.24494897427831780
Середнє усічене (20%):				**0.6769232500000000**	**0.5812500000000000**	**0.26703239900000000**
181.	10	45	PA	0.5555555555555556	0.5000000000000000	0.30000000000000000
182.	10	45	PA	0.4166666666666667	0.5000000000000000	0.34641016151377546
183.	10	45	PA	0.5714285714285714	0.8000000000000000	0.28284271247461900
184.	10	45	PA	0.8888888888888888	0.8000000000000000	0.17320508075688773
185.	10	45	PA	0.6666666666666666	0.4000000000000000	0.28284271247461900
186.	10	45	PA	0.6000000000000000	0.6000000000000000	0.28284271247461900
187.	10	45	PA	0.5000000000000000	0.4000000000000000	0.31622776601683794
188.	10	45	PA	0.6250000000000000	0.5000000000000000	0.28284271247461900
189.	10	45	PA	0.7000000000000000	0.7000000000000000	0.24494897427831780
190.	10	45	PA	0.5833333333333334	0.7000000000000000	0.28284271247461900
191.	10	45	PA	1.0000000000000000	0.7000000000000000	0.17320508075688773
192.	10	45	PA	0.5000000000000000	0.5000000000000000	0.31622776601683794
193.	10	45	PA	0.5714285714285714	0.4000000000000000	0.30000000000000000
194.	10	45	PA	0.6363636363636364	0.7000000000000000	0.26457513110645910
195.	10	45	PA	0.8571428571428571	0.6000000000000000	0.22360679774997896
196.	10	45	PA	0.5000000000000000	0.2000000000000000	0.31622776601683794
197.	10	45	PA	0.7777777777777778	0.7000000000000000	0.22360679774997896
198.	10	45	PA	0.6000000000000000	0.6000000000000000	0.28284271247461900
199.	10	45	PA	0.8571428571428571	0.6000000000000000	0.22360679774997896
200.	10	45	PA	0.6666666666666666	0.4000000000000000	0.28284271247461900
Середнє усічене (20%):				**0.6417816560000000**	**0.5687500000000000**	**0.27454368900000000**

У таблиці 6.6. наведені для наглядності середні значення з таблиці 6.5.

Таблиця 6.6. Середні значення точності роботи розробленого методу для різної кількості цілей інформаційної атаки у ботів

Кількість цілей у кожного бота	**Точність (Precision) розпізнавання ботів**	**Повнота (Recall) розпізнавання ботів**	**RMSE класифікації користувачів на ботів та аутентичних**
1	0.571391191	0.843750000	0.240591404
5	0.750070242	0.925000000	0.208735147
10	0.751688997	0.937500000	0.208005683
15	0.757313520	0.950000000	0.194432974
20	0.759444202	0.931250000	0.191765580
25	0.780512370	0.925000000	0.189773836
30	0.781431763	0.875000000	0.192072089
35	0.684367195	0.737500000	0.242168980
40	0.676923250	0.581250000	0.267032399
45	0.641781656	0.568750000	0.274543689
Сер. знач.:	**0.715492439**	**0.827500000**	**0.220912178**

Як показали проведені експерименти, точність розпізнавання ботів розробленим методом для популярної атаки в середньому становить 0.71, повнота 0.82, а RMSE – 0.22. Найгірші результати одержувалися, коли ціль для атаки ботів була одна, тоді точність розробленого методу падала до 0.57 у середньому. Найвища точність спостерігалася, коли цілей атак було 25-30 шт., в такому разі вона сягала значення 0.78.

6.1.3. Дослідження показників стійкості розробленого гібридного методу колаборативної фільтрації з запропонованою підсистемою інформаційної безпеки до зовнішніх дестабілізуючих факторів

Було проведено серію експериментів для визначення стійкості розробленого гібридного методу колаборативної фільтрації з використанням

запропонованої підсистеми інформаційної безпеки в умовах дії зовнішніх дестабілізуючих факторів у вигляді інформаційних атак ін'єкцією профілів.

Розроблений гібрид містить у собі наступні методи:

1) існуючий метод колаборативної фільтрації на основі моделі сусідства;

2) розроблений метод колаборативної фільтрації з використанням продукційних правил для визначення відсутніх коефіцієнтів подоби між користувачами;

3) розроблений метод експертно-орієнтованої колаборативної фільтрації з врахуванням показників активності користувачів.

Розроблена підсистема інформаційної безпеки рекомендаційної системи складається з наступних елементів:

1) розроблений метод виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів у рейтингах об'єктів;

2) розроблений метод виявлення бот-мереж у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів;

3) розроблений спосіб ідентифікації профілів ботів на основі нейронних мереж у рекомендаційних системах.

Набори даних для експериментів генерувалися у розробленій програмній імітаційній моделі рекомендаційної системи [53]. Формат та статистичні особливості даних генерувалися максимально наближеними до набору даних MovieLens Datasets [30]. Інформаційні атаки моделювалися за допомогою випадкової, середньої та популярної моделей атак [28, 40, 76].

Показники стійкості рекомендаційної системи розраховувалися за формулами (3.38)-(3.41). Чим менше значення суми зсувів у роботі системи при формуванні списків рекомендацій (консолідованого показника стійкості) тим вища стійкість системи до дестабілізуючих факторів.

Результати проведеної серії експериментів наведені у таблицях 6.7-6.15. У даних таблицях були використані наступні скорочення:

– **МС** – відома колаборативна фільтрація на основі моделі сусідства;

– **ФМ** – відома колаборативна фільтрація на основі факторизації матриць;

– **Г+** – розроблений гібридний метод колаборативної фільтрації.

– **ІМЗ** – відомий метод захисту рекомендаційної системи від інформаційних атак;

– **РМЗ** – розроблена підсистема інформаційної безпеки рекомендаційної системи.

Таблиця 6.7. Показники стійкості існуючого методу виявлення ботів при застосуванні до існуючих та розробленого методів колаборативної фільтрації при випадковій атаці

№ експерименту	Середній зсув прогнозувань оцінок для користувачів			Середній зсув кількості елементів у списках рекомендацій користувача			Середній зсув прогнозувань оцінок для об'єктів			Середній зсув кількості потраплянь у списки рекомендацій для об'єктів		
	МС	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**
1	0.0010	0.0052	0.0011	0.0601	0.0025	0.1160	0.0004	0.0026	0.0005	0.0141	0.0003	0.0392
2	0.0004	0.0046	0.0004	0.0425	0.0016	0.0578	0.0002	0.0023	0.0002	0.0105	0.0027	0.0193
3	0.0009	0.0053	0.0010	0.0703	0.0000	0.1351	0.0004	0.0027	0.0005	0.0024	0.0000	0.0123
4	0.0013	0.0044	0.0013	0.1114	0.0000	0.1332	0.0006	0.0023	0.0006	0.0205	0.0005	0.0287
5	0.0013	0.0050	0.0012	0.0877	0.0032	0.1044	0.0006	0.0026	0.0006	0.0265	0.0014	0.0314
6	0.0008	0.0043	0.0009	0.0712	0.0107	0.1090	0.0004	0.0022	0.0004	0.0015	0.0189	0.0086
7	0.0006	0.0042	0.0006	0.0596	0.0069	0.0669	0.0003	0.0021	0.0003	0.0115	0.0054	0.0152
8	0.0009	0.0048	0.0009	0.0890	0.0000	0.1185	0.0004	0.0024	0.0004	0.0165	0.0000	0.0246
9	0.0011	0.0048	0.0010	0.1112	0.0000	0.1099	0.0005	0.0024	0.0005	0.0271	0.0000	0.0213
10	0.0008	0.0047	0.0007	0.0629	0.0000	0.0800	0.0003	0.0024	0.0003	0.0372	0.0044	0.0639
С. з.:	**0.00091**	**0.00473**	**0.00091**	**0.07659**	**0.00249**	**0.10308**	**0.00041**	**0.0024**	**0.00043**	**0.01678**	**0.00336**	**0.02645**

Таблиця 6.8. Показники стійкості розробленого методу виявлення ботів при застосуванні до існуючих та розробленого методів колаборативної фільтрації при випадковій атаці

№ експерименту	Середній зсув прогнозувань оцінок для користувачів			Середній зсув кількості елементів у списках рекомендацій користувача			Середній зсув прогнозувань оцінок для об'єктів			Середній зсув кількості потраплянь у списки рекомендацій для об'єктів		
	МС	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**
1	0.0000	0.0052	0.0000	0.0000	0.0000	0.0111	0.0000	0.0026	0.0000	0.0000	0.0000	0.0026
2	0.0004	0.0044	0.0004	0.0352	0.0013	0.0473	0.0002	0.0023	0.0002	0.0046	0.0013	0.0110
3	0.0005	0.0053	0.0007	0.0511	0.0000	0.1023	0.0003	0.0027	0.0003	0.0000	0.0000	0.0000
4	0.0004	0.0042	0.0004	0.0288	0.0000	0.0426	0.0002	0.0022	0.0002	0.0043	0.0000	0.0066
5	0.0000	0.0049	0.0000	0.0000	0.0057	0.0114	0.0000	0.0025	0.0000	0.0000	0.0034	0.0027
6	0.0005	0.0044	0.0005	0.0327	0.0068	0.0005	0.0002	0.0022	0.0002	0.0000	0.0115	0.0039
7	0.0004	0.0043	0.0004	0.0480	0.0000	0.0553	0.0002	0.0022	0.0002	0.0063	0.0000	0.0083
8	0.0002	0.0046	0.0001	0.0190	0.0000	0.0208	0.0001	0.0024	0.0001	0.0029	0.0000	0.0047
9	0.0005	0.0045	0.0005	0.0486	0.0000	0.0509	0.0002	0.0022	0.0002	0.0101	0.0000	0.0092
10	0.0008	0.0684	0.0045	0.0024	0.0007	0.0753	0.0003	0.0023	0.0003	0.0344	0.0025	0.0563
С. з.:	**0.00037**	**0.01102**	**0.00075**	**0.02658**	**0.00145**	**0.04175**	**0.00017**	**0.00236**	**0.00017**	**0.00626**	**0.00187**	**0.01053**

Як видно з таблиць 6.7 та 6.8, розроблена підсистема інформаційної безпеки дозволяє знизити вплив випадкової атаки ботів на формування списків рекомендацій (що проявляється у зменшенні значень переважної більшості вимірюваних зсувів), а отже, підвищити стійкість рекомендаційної системи до зовнішніх дестабілізуючих факторів. Для того, щоб більш наглядно показати цей факт, у таблиці 6.9 наведено консолідовані показники стійкості для таблиць 6.7 та 6.8, розраховані за формулою (3.43).

Таблиця 6.9. Консолідовані показники стійкості для існуючого та розробленого методів виявлення ботів при випадковій атаці

№ експ.	**Стійкість. Консолідований показник**					
	МС		**ФМ**		**Г+**	
	ІМЗ	**РМЗ**	**ІМЗ**	**РМЗ**	**ІМЗ**	**РМЗ**
1	0.08551	0.00000	0.05895	0.05720	0.21870	0.01411
2	0.06115	0.03092	0.06426	0.05703	0.10668	0.06413
3	0.02793	0.01161	0.05930	0.05930	0.08601	0.01693
4	0.12694	0.02878	0.05290	0.04820	0.17012	0.04166
5	0.15457	0.00000	0.06432	0.07247	0.18064	0.01464
6	0.02342	0.00777	0.14387	0.10658	0.06280	0.02405
7	0.07006	0.04070	0.07389	0.04830	0.08929	0.05143
8	0.10030	0.01860	0.05280	0.05260	0.14375	0.02768
9	0.15772	0.05986	0.05280	0.04850	0.12849	0.05559
10	0.19909	0.17904	0.07470	0.12697	0.33420	0.29953
С.з.:	**0.100669**	**0.037728**	**0.069779**	**0.067715**	**0.152068**	**0.060975**

Таблиця 6.10. Показники стійкості існуючого методу виявлення ботів при застосуванні до існуючих та розробленого методів колаборативної фільтрації при середній атаці

№ експерименту	**Середній зсув прогнозувань оцінок для користувачів**			**Середній зсув кількості елементів у списках рекомендацій користувача**			**Середній зсув прогнозувань оцінок для об'єктів**			**Середній зсув кількості потраплянь у списки рекомендацій для об'єктів**		
	МС	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**
1	0.0005	0.0044	0.0005	0.0747	0.0004	0.0870	0.0004	0.0026	0.0005	0.0141	0.0003	0.0392
2	0.0007	0.0045	0.0006	0.0790	0.0014	0.1126	0.0002	0.0023	0.0002	0.0105	0.0027	0.0193
3	0.0010	0.0035	0.0011	0.1628	0.0089	0.1629	0.0004	0.0027	0.0005	0.0024	0.0000	0.0123
4	0.0004	0.0051	0.0004	0.0491	0.0000	0.0721	0.0006	0.0023	0.0006	0.0205	0.0005	0.0287
5	0.0009	0.0043	0.0009	0.1105	0.0000	0.1210	0.0006	0.0026	0.0006	0.0265	0.0014	0.0314
6	0.0005	0.0045	0.0004	0.0491	0.0029	0.0611	0.0004	0.0022	0.0004	0.0015	0.0189	0.0086
7	0.0013	0.0045	0.0013	0.1055	0.0000	0.1216	0.0003	0.0021	0.0003	0.0115	0.0054	0.0152
8	0.0009	0.0036	0.0009	0.0935	0.0014	0.1030	0.0004	0.0024	0.0004	0.0165	0.0000	0.0246
9	0.0013	0.0043	0.0012	0.1269	0.0000	0.1395	0.0005	0.0024	0.0005	0.0271	0.0000	0.0213
10	0.0013	0.0041	0.0013	0.1187	0.0079	0.1352	0.0003	0.0024	0.0003	0.0372	0.0044	0.0639
С. з.:	**0.00088**	**0.00428**	**0.00086**	**0.09698**	**0.00229**	**0.1116**	**0.00041**	**0.0024**	**0.00043**	**0.01678**	**0.00336**	**0.02645**

Таблиця 6.11. Показники стійкості розробленого методу виявлення ботів при застосуванні до існуючих та розробленого методів колаборативної фільтрації при середній атаці

№ експерименту	Середній зсув прогнозувань оцінок для користувачів			Середній зсув кількості елементів у списках рекомендацій користувача			Середній зсув прогнозувань оцінок для об'єктів			Середній зсув кількості потраплянь у списки рекомендацій для об'єктів		
	МС	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**
1	0.0000	0.0040	0.0000	0.0000	0.0099	0.0062	0.0000	0.0026	0.0000	0.0000	0.0000	0.0026
2	0.0000	0.0045	0.0000	0.0000	0.0002	0.0054	0.0002	0.0023	0.0002	0.0046	0.0013	0.0110
3	0.0003	0.0036	0.0003	0.0576	0.0060	0.0586	0.0003	0.0027	0.0003	0.0000	0.0000	0.0000
4	0.0001	0.0052	0.0000	0.0074	0.0000	0.0080	0.0002	0.0022	0.0002	0.0043	0.0000	0.0066
5	0.0001	0.0042	0.0001	0.0163	0.0000	0.0230	0.0000	0.0025	0.0000	0.0000	0.0034	0.0027
6	0.0001	0.0047	0.0001	0.0097	0.0000	0.0097	0.0002	0.0022	0.0002	0.0000	0.0115	0.0039
7	0.0005	0.0047	0.0005	0.0306	0.0000	0.0326	0.0002	0.0022	0.0002	0.0063	0.0021	0.0083
8	0.0000	0.0038	0.0000	0.0000	0.0000	0.0042	0.0001	0.0024	0.0001	0.0029	0.0028	0.0047
9	0.0000	0.0044	0.0000	0.0000	0.0000	0.0052	0.0002	0.0022	0.0002	0.0101	0.0010	0.0092
10	0.0003	0.0041	0.0003	0.0271	0.0052	0.0291	0.0003	0.0023	0.0003	0.0344	0.0025	0.0563
С. з.:	**0.00014**	**0.00432**	**0.00013**	**0.01487**	**0.00213**	**0.0182**	**0.00017**	**0.00236**	**0.00017**	**0.00626**	**0.00246**	**0.01053**

Як видно з таблиць 6.10 та 6.11, розроблена підсистема інформаційної безпеки дозволяє знизити вплив середньої атаки ботів на формування списків рекомендацій (що проявляється у зменшенні значень переважної більшості вимірюваних зсувів), а отже, підвищити стійкість рекомендаційної системи до зовнішніх дестабілізуючих факторів. Для того, щоб більш наглядно показати цей факт, у таблиці 6.12 наведено консолідовані показники стійкості.

Таблиця 6.12. Консолідовані показники стійкості для існуючого та розробленого методів виявлення ботів при середній атаці

№ експ.	Стійкість. Консолідований показник					
	МС		**ФМ**		**Г+**	
	ІМЗ	**РМЗ**	**ІМЗ**	**РМЗ**	**ІМЗ**	**РМЗ**
1	0.08647	0.00000	0.05794	0.05699	0.21520	0.01362
2	0.06510	0.02700	0.06414	0.05702	0.11236	0.05954
3	0.03728	0.01206	0.05839	0.05820	0.08889	0.01216
4	0.11981	0.02634	0.05360	0.04920	0.16311	0.03780
5	0.15645	0.00173	0.06330	0.07120	0.18200	0.01590
6	0.02091	0.00507	0.14329	0.10620	0.05751	0.02457
7	0.07535	0.03906	0.07350	0.05920	0.09546	0.04926
8	0.10075	0.01650	0.05174	0.06580	0.14220	0.02592
9	0.15949	0.05450	0.05230	0.05340	0.13165	0.05052
10	0.20517	0.18101	0.07489	0.06312	0.34032	0.29071
С.з.:	**0.102678**	**0.036327**	**0.069309**	**0.064033**	**0.152870**	**0.058000**

Таблиця 6.13. Показники стійкості існуючого методу виявлення ботів при застосуванні до існуючих та розробленого методів колаборативної фільтрації при популярній атаці

№ експерименту	Середній зсув прогнозувань оцінок для користувачів			Середній зсув кількості елементів у списках рекомендацій користувача			Середній зсув прогнозувань оцінок для об'єктів			Середній зсув кількості потраплянь у списки рекомендацій для об'єктів		
	МС	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**
1	0.0012	0.0048	0.0012	0.1006	0.0000	0.1208	0.0004	0.0026	0.0003	0.0141	0.0003	0.0089
2	0.0006	0.0047	0.0006	0.0423	0.0090	0.0663	0.0002	0.0023	0.0003	0.0105	0.0027	0.0089
3	0.0010	0.0042	0.0010	0.0856	0.0087	0.1274	0.0004	0.0027	0.0003	0.0024	0.0000	0.0089
4	0.0008	0.0048	0.0009	0.0637	0.0003	0.0994	0.0006	0.0023	0.0003	0.0205	0.0005	0.0089
5	0.0005	0.0051	0.0005	0.0429	0.0033	0.0639	0.0006	0.0026	0.0003	0.0265	0.0014	0.0089
6	0.0008	0.0038	0.0008	0.0757	0.0114	0.0844	0.0004	0.0022	0.0003	0.0015	0.0189	0.0089
7	0.0010	0.0049	0.0011	0.0671	0.0000	0.1115	0.0003	0.0021	0.0003	0.0115	0.0054	0.0089
8	0.0010	0.0043	0.0010	0.1012	0.0000	0.1042	0.0004	0.0024	0.0003	0.0165	0.0000	0.0089
9	0.0004	0.0044	0.0004	0.0504	0.0004	0.0786	0.0005	0.0024	0.0003	0.0271	0.0000	0.0089
10	0.0005	0.0043	0.0006	0.0602	0.0050	0.0886	0.0003	0.0024	0.0003	0.0372	0.0044	0.0089
С. з.:	**0.00078**	**0.00453**	**0.00081**	**0.06897**	**0.00381**	**0.09451**	**0.00041**	**0.0024**	**0.0003**	**0.01678**	**0.00336**	**0.0089**

Таблиця 6.14. Показники стійкості розробленого методу виявлення ботів при застосуванні до існуючих та розробленого методів колаборативної фільтрації при популярній атаці

№ експерименту	Середній зсув прогнозувань оцінок для користувачів			Середній зсув кількості елементів у списках рекомендацій користувача			Середній зсув прогнозувань оцінок для об'єктів			Середній зсув кількості потраплянь у списки рекомендацій для об'єктів		
	МС	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**	**МС**	**ФМ**	**Г+**
1	0.0005	0.0053	0.0005	0.0435	0.0000	0.0536	0.0000	0.0026	0.0000	0.0000	0.0000	0.0026
2	0.0001	0.0048	0.0001	0.0134	0.0065	0.0261	0.0002	0.0023	0.0002	0.0046	0.0013	0.0110
3	0.0006	0.0043	0.0006	0.0519	0.0072	0.0721	0.0003	0.0027	0.0003	0.0000	0.0000	0.0000
4	0.0005	0.0050	0.0005	0.0303	0.0000	0.0503	0.0002	0.0022	0.0002	0.0043	0.0000	0.0066
5	0.0003	0.0050	0.0003	0.0318	0.0000	0.0461	0.0000	0.0025	0.0000	0.0000	0.0034	0.0027
6	0.0004	0.0041	0.0004	0.0397	0.0006	0.0447	0.0002	0.0022	0.0002	0.0000	0.0115	0.0039
7	0.0007	0.0051	0.0008	0.0476	0.0000	0.0691	0.0002	0.0022	0.0002	0.0063	0.0000	0.0083
8	0.0006	0.0046	0.0006	0.0680	0.0000	0.0709	0.0001	0.0024	0.0001	0.0029	0.0000	0.0047
9	0.0001	0.0046	0.0001	0.0174	0.0000	0.0355	0.0002	0.0022	0.0002	0.0101	0.0000	0.0092
10	0.0006	0.0046	0.0006	0.0622	0.0000	0.0761	0.0003	0.0023	0.0003	0.0344	0.0025	0.0563
С. з.:	**0.00044**	**0.00474**	**0.00045**	**0.04058**	**0.00143**	**0.05445**	**0.00017**	**0.00236**	**0.00017**	**0.00626**	**0.00187**	**0.01053**

Як видно з таблиць 6.13 та 6.14, розроблена підсистема інформаційної безпеки дозволяє знизити вплив популярної атаки ботів на формування списків рекомендацій (що проявляється у зменшенні значень переважної більшості вимірюваних зсувів), а отже, підвищити стійкість рекомендаційної системи до

зовнішніх дестабілізуючих факторів. Для того, щоб більш наглядно показати цей факт, у таблиці 6.15 наведено консолідовані показники стійкості.

Таблиця 6.15. Консолідовані показники стійкості для існуючого та розробленого методів виявлення ботів при популярній атаці

№ експ.	Стійкість. Консолідований показник					
	МС		ФМ		Г+	
	ІМЗ	РМЗ	ІМЗ	РМЗ	ІМЗ	РМЗ
1	0.08976	0.00485	0.05830	0.05730	0.06378	0.01886
2	0.06133	0.02844	0.06510	0.05795	0.05773	0.06171
3	0.02956	0.01179	0.05907	0.05902	0.06424	0.01381
4	0.12167	0.02903	0.05333	0.04900	0.06134	0.04253
5	0.14929	0.00348	0.06443	0.07200	0.05739	0.01841
6	0.02387	0.00837	0.14344	0.10566	0.05974	0.02837
7	0.07121	0.04096	0.07390	0.04910	0.06275	0.05321
8	0.10162	0.02390	0.05230	0.05260	0.06192	0.03319
9	0.15094	0.05634	0.05244	0.04860	0.05876	0.05365
10	0.19852	0.18482	0.07480	0.06310	0.05996	0.29571
С.з.:	**0.099777**	**0.039198**	**0.069711**	**0.061433**	**0.060761**	**0.061945**

Таким чином, результати експериментів показали, що розроблена підсистема інформаційної безпеки дозволяє забезпечити вищу стійкість рекомендаційної системи до зовнішніх дестабілізуючих факторів на відміну від існуючих методів. Вона в середньому показує в 2.5 рази кращі результати значення стійкості для випадкової та середньої атаки та в 1.7 разів кращі результати для популярної атаки, якщо застосовувалася до методу колаборативної фільтрації на основі сусідства та до розробленого гібридного методу. Для методу на основі матричної факторизації вона показує практично такі ж результати, як у існуючого методу на основі кластеризації профілів користувачів за їх статистичними даними.

6.2. Обґрунтування достовірності одержаних результатів наукових досліджень

Проведемо обґрунтування достовірності отриманих у 3-6 розділах даної роботи результатів.

За результатами серії експериментів, частина яких наведена у табл. 6.5-

6.6, отримані гістограми частоти безпомилкового розпізнавання профілів ботів, частоти повного розпізнавання профілів ботів, а також RMSE класифікації користувачів на ботів та аутентичних. Гістограми представлені на рис. 6.3, рис. 6.4 та рис. 6.5 відповідно.

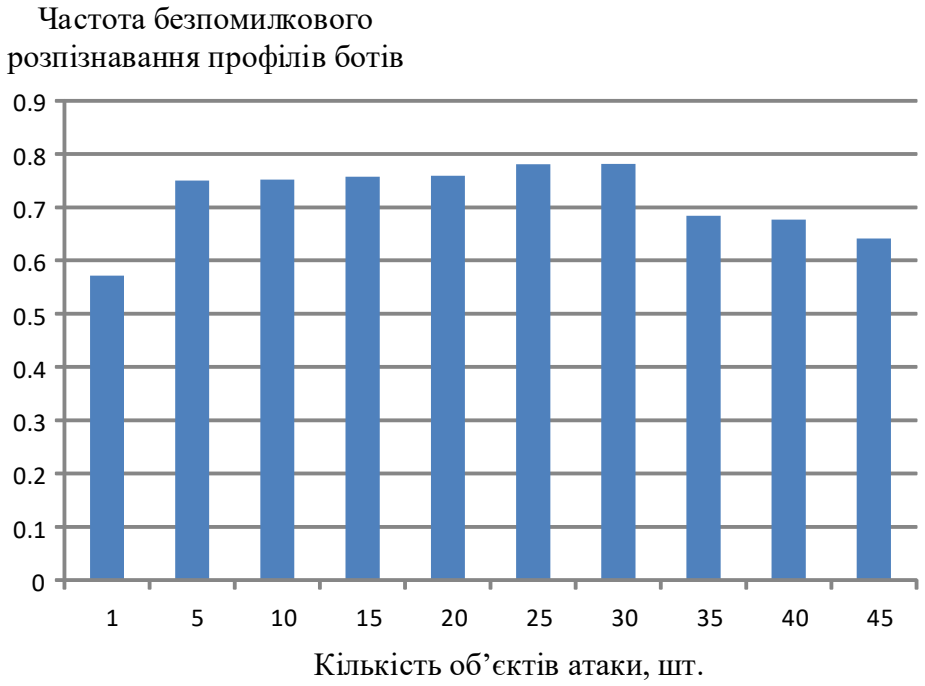


Рис. 6.3. Гістограма частот безпомилкового розпізнавання профілів ботів

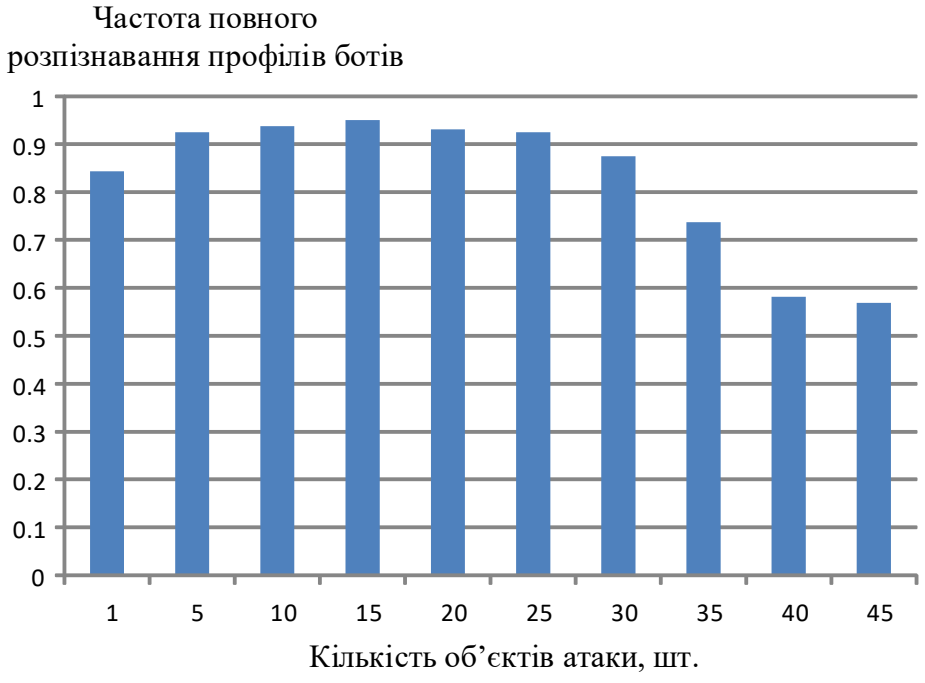


Рис. 6.4. Гістограма частот повного розпізнавання профілів ботів

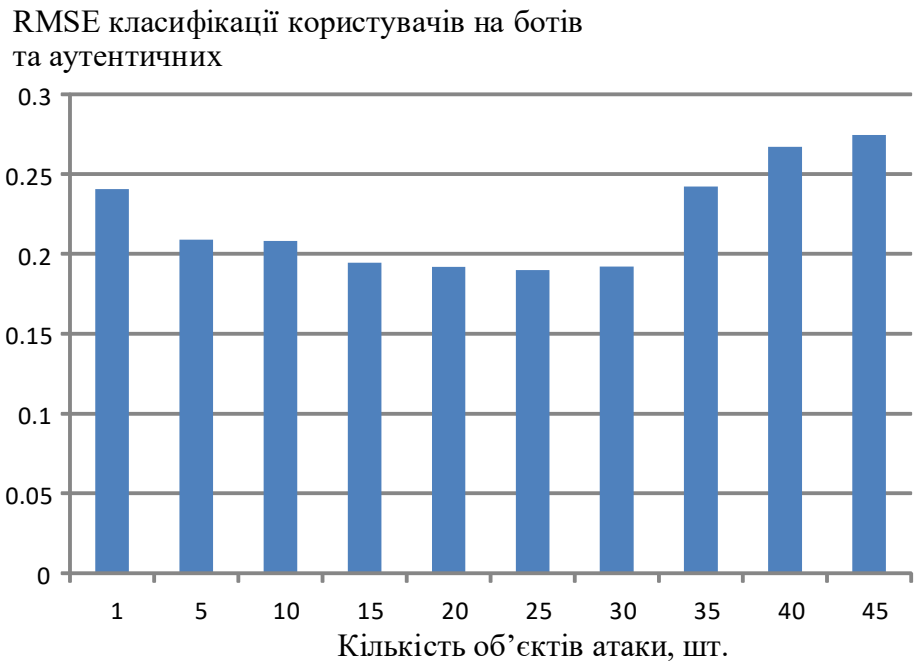


Рис. 6.5. Гістограма RMSE класифікації користувачів на ботів та аутентичних

Висунута гіпотеза про нормальний розподіл цих випадкових величин була перевірена за критерієм згоди χ^2 Пірсона [115, 117, 214]:

$$\chi^2 = N^* \sum_{i=1}^{k} \left(P_i^* - P_i \right)^2 / P_i \,, \tag{6.7}$$

де k – кількість розрядів (інтервалів) статистичного ряду; P_i^* та P_i – «статистична» та теоретична ймовірності «попадання» заданого показника до i-й розряду.

Проведена перевірка довела правдоподібність гіпотези про те, що величини частоти безпомилкового розпізнавання профілів ботів та частоти повного розпізнавання профілів ботів розподілені за нормальним законом.

Отримано оцінки $\hat{P}^{(i)}_{безпом}$ математичного сподівання та $\hat{D}_{P^{(i)}_{безпом}}$ дисперсії ($\hat{\sigma}_{P^{(i)}_{безпом}}$ середньоквадратичного відхилення) випадкової величини $P^{(i)}_{безпом}$, що характеризує частоту безпомилкового розпізнавання профілів ботів, а також оцінки $\hat{P}^{(i)}_{повн}$ математичного сподівання та $\hat{D}_{P^{(i)}_{повн}}$ дисперсії

($\hat{\sigma}_{P^{(i)}_{повн}}$ середньоквадратичного відхилення) випадкової величини $P^{(i)}_{повн}$, що характеризує частоту повного розпізнавання профілів ботів:

$$\hat{P}^{(i)}_{безпом} = \frac{\sum_{j=1}^{k} P^{(i,j)}_{безпом}}{N^*}; \quad \hat{D}_{P^{(i)}_{безпом}} = \frac{\sum_{j=1}^{k} \left(\hat{P}^{(i)}_{безпом} - P^{(i,j)}_{безпом}\right)^2}{N^* - 1}; \quad \hat{\sigma}_{P^{(i)}_{безпом}} = \sqrt{\hat{D}_{P^{(i)}_{безпом}}}.$$

$$\hat{P}^{(i)}_{повн} = \frac{\sum_{j=1}^{k} P^{(i,j)}_{повн}}{N^*}; \quad \hat{D}_{P^{(i)}_{повн}} = \frac{\sum_{j=1}^{k} \left(\hat{P}^{(i)}_{повн} - P^{(i,j)}_{повн}\right)^2}{N^* - 1}; \quad \hat{\sigma}_{P^{(i)}_{повн}} = \sqrt{\hat{D}_{P^{(i)}_{повн}}}.$$

Скориставшись відомим виразом для розрахунку довірчої ймовірності відхилення відносної частоти від постійної ймовірності в незалежних випробуваннях [114, 214] визначимо довірчу ймовірність того, що отримані в результаті експерименту значення характеристики частоти безпомилкового розпізнавання профілів ботів $P^{(i)}_{безпом}$ «не відхилиться» від математичного сподівання $\hat{P}^{(i)}_{безпом}$, а також значення характеристики та частоти повного розпізнавання $P^{(i)}_{повн}$ профілів ботів «не відхилиться» від математичного сподівання $\hat{P}^{(i)}_{повн}$ відповідно більше, ніж на 0.05:

$$P\left(\left|\hat{P}^{(i)}_{безпом} - P^{(i)}_{безпом}\right| < 0.05\right) = 2\Phi\left(0.05 \Big/ \hat{\sigma}_{P^{(i)}_{безпом}}\right),$$

$$P\left(\left|\hat{P}^{(i)}_{повн} - P^{(i)}_{повн}\right| < 0.05\right) = 2\Phi\left(0.05 \Big/ \hat{\sigma}_{P^{(i)}_{повн}}\right),$$

де Φ – функція Лапласа виду $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-t^2/2}\, dt$ [117, 214].

Результати проведених експериментів показали, що для всіх досліджуваних видів даних довірча ймовірність того, що значення статистичної величини $P^{(i)}_{безпом}$ не відхиляється від математичного сподівання $\hat{P}^{(i)}_{безпом}$ більше, ніж на 0.05, дорівнює: $P \approx 0.94$.

Також результати проведених експериментів показали, що для всіх досліджуваних видів даних довірча ймовірність того, що значення

статистичної величини $P_{повн}^{(i)}$ не відхиляється від математичного сподівання $\hat{P}_{повн}^{(i)}$ більше, ніж на 0.05, дорівнює: $P \approx 0.95$.

За результатами серії експериментів, частина яких наведена у табл. 3.6-3.7, отримані гістограми частот середнього зсуву прогнозувань оцінок для користувачів, середнього зсуву кількості елементів у списках рекомендацій користувача, середнього зсуву прогнозувань оцінок для об'єктів, та середнього зсуву кількості потраплянь у списки рекомендацій для об'єктів. Гістограми представлені на рис. 6.6 (а-г) відповідно.

а)

б)

в)

г)

Рис. 6.6. Гістограми частот середнього зсуву у визначеному діапазоні

Проведена перевірка довела правдоподібність гіпотези про те, що величини частот середнього зсуву у визначеному діапазоні розподілені за

нормальним законом.

Аналогічно попередньому прикладу, скориставшись відомим виразом для розрахунку довірчої ймовірності відхилення відносної частоти від постійної ймовірності в незалежних випробуваннях [114, 214] визначимо довірчу ймовірність того, що отримані в результаті експерименту значення характеристики частот середнього зсуву прогнозувань оцінок для користувачів $P_{знок}^{(i)}$, середнього зсуву кількості елементів у списках рекомендацій користувача $P_{зкес}^{(i)}$, середнього зсуву прогнозувань оцінок для об'єктів $P_{зноо}^{(i)}$, та середнього зсуву кількості потраплянь у списки рекомендацій для об'єктів $P_{зпкс}^{(i)}$ «не відхилиться» від математичних сподівань $\hat{P}_{знок}^{(i)}$, $\hat{P}_{зкес}^{(i)}$, $\hat{P}_{зноо}^{(i)}$, $\hat{P}_{зпкс}^{(i)}$ відповідно більше, ніж на 0.05:

$$P\left(\left|\hat{P}_{знок}^{(i)} - P_{знок}^{(i)}\right| < 0.05\right) = 2\Phi\left(0.05 \Big/ \hat{\sigma}_{P_{знок}^{(i)}}\right),$$

$$P\left(\left|\hat{P}_{зкес}^{(i)} - P_{зкес}^{(i)}\right| < 0.05\right) = 2\Phi\left(0.05 \Big/ \hat{\sigma}_{P_{зкес}^{(i)}}\right),$$

$$P\left(\left|\hat{P}_{зноо}^{(i)} - P_{зноо}^{(i)}\right| < 0.05\right) = 2\Phi\left(0.05 \Big/ \hat{\sigma}_{P_{зноо}^{(i)}}\right),$$

$$P\left(\left|\hat{P}_{зпкс}^{(i)} - P_{зпкс}^{(i)}\right| < 0.05\right) = 2\Phi\left(0.05 \Big/ \hat{\sigma}_{P_{зпкс}^{(i)}}\right).$$

Результати проведених експериментів показали, що для всіх досліджуваних видів даних довірча ймовірність того, що значення статистичної величини частоти середнього зсуву прогнозувань оцінок для користувачів $P_{знок}^{(i)}$ не відхиляється від математичного сподівання $\hat{P}_{знок}^{(i)}$ більше, ніж на 0.05, дорівнює: $P \approx 0.96$. Довірча ймовірність того, що значення статистичної величини $P_{зкес}^{(i)}$ не відхиляється від математичного сподівання $\hat{P}_{зкес}^{(i)}$ більше, ніж на 0.05, дорівнює: $P \approx 0.95$. Довірча ймовірність того, що значення статистичної величини $P_{зноо}^{(i)}$ не відхиляється від математичного сподівання $\hat{P}_{зноо}^{(i)}$

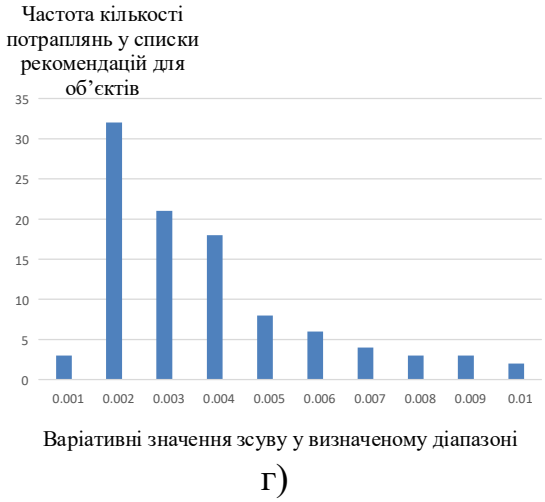
більше, ніж на 0.05, дорівнює: $P \approx 0.94$. Довірча ймовірність того, що значення статистичної величини $P_{зпкс}^{(i)}$ не відхиляється від математичного сподівання $\hat{P}_{зпкс}^{(i)}$ більше, ніж на 0.05, дорівнює: $P \approx 0.95$.

За результатами серії експериментів, частина яких наведена у табл. 6.7-6.9, отримані гістограми частот середнього зсуву прогнозувань для користувачів $P_{зпк}^{(i)}$, середнього зсуву кількості елементів у списках рекомендацій користувача $P_{зке}^{(i)}$, середнього зсуву прогнозувань для об'єктів $P_{зпо}^{(i)}$, та середнього зсуву кількості потраплянь у списки рекомендацій для об'єктів $P_{зкп}^{(i)}$ при випадковій атаці. Гістограми представлені на рис. 6.7 (а-г) відповідно.

а)

б)

в)



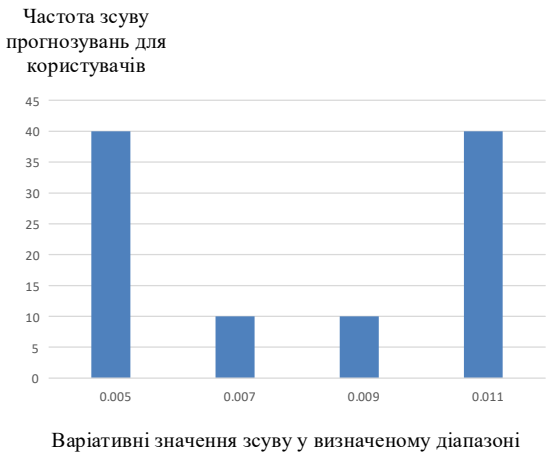
г)

Рис. 6.7. Гістограми частот зсуву при випадковій атаці у визначеному діапазоні

Результати проведених експериментів показали, що для всіх досліджуваних видів даних довірча ймовірність того, що значення статистичних величин частот $P^{(i)}_{зпк}$, $P^{(i)}_{зке}$, $P^{(i)}_{зпо}$, $P^{(i)}_{зкп}$ не відхиляється від математичного сподівання $\hat{P}^{(i)}_{зпк}$, $\hat{P}^{(i)}_{зке}$, $\hat{P}^{(i)}_{зпо}$, $\hat{P}^{(i)}_{зкп}$ більше, ніж на 0.05, дорівнює: $P \approx 0.96$.

За результатами серії експериментів, частина яких наведена у табл. 6.10-6.12, отримані гістограми частот середнього зсуву прогнозувань для користувачів $P^{(i)}_{зпкс}$, середнього зсуву кількості елементів у списках рекомендацій $P^{(i)}_{зкес}$, середнього зсуву прогнозувань для об'єктів $P^{(i)}_{зпос}$, та середнього зсуву кількості потраплянь у списки рекомендацій для об'єктів $P^{(i)}_{зкпс}$ при середній атаці. Гістограми представлені на рис. 6.8 (а-г) відповідно.

а)



б)

в)

г)

Рис. 6.8. Гістограми частот зсуву при середній атаці у визначеному діапазоні

Результати проведених експериментів показали, що для всіх досліджуваних видів даних довірча ймовірність того, що значення статистичних величин частот $P_{зпкс}^{(i)}$, $P_{зкес}^{(i)}$, $P_{зпос}^{(i)}$, $P_{зкпс}^{(i)}$ не відхиляється від математичного сподівання $\hat{P}_{зпкс}^{(i)}$, $\hat{P}_{зкес}^{(i)}$, $\hat{P}_{зпос}^{(i)}$, $\hat{P}_{зкпс}^{(i)}$ більше, ніж на 0.05, дорівнює: $P \approx 0.94$.

Це підтверджує достовірність розроблених в 3-6 розділах моделей та методів синтезу рекомендаційних систем і одержаних результатів наукових досліджень.

Висновки до розділу 6

У даному розділі запропоновано метод виявлення та нейтралізації мережі ботів у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів. Проведені дослідження точності його роботи.

Як показали проведені експерименти, точність розпізнавання ботів розробленим методом в середньому становить 0.72 для випадкової атаки, 0.81 для середньої атаки, а також 0.71 для популярної атаки. Найгірші результати одержувалися, коли ціль для атаки ботів була одна, тоді точність розробленого методу, наприклад, для популярної атаки падала у середньому до 0.57. Найвища точність спостерігалася, коли цілей атаки було 25-30 шт., в такому разі вона, наприклад, для популярної атаки, сягала значення 0.78. При вдалій атаці ботів, що зсунула рейтинги цільових об’єктів, наявність бот-мережі та значний процент ботів з неї завжди було виявлено. Усі профілі користувачів, ідентифіковані як боти, доречно перевірити за допомогою існуючих методів ідентифікації ботів на основі статистичного аналізу даних окремого профіля. Що дозволить уточнити результати та видалити з даної сукупності профілів користувачів профілі, помилково ідентифіковані як боти. Для уточнення результатів розробленого методу можна використати також запропонований спосіб ідентифікації профілів ботів на основі нейронних мереж.

Розроблена підсистема інформаційної безпеки дозволяє забезпечити вищу стійкість рекомендаційної системи до зовнішніх дестабілізуючих факторів на відміну від існуючих методів. Вона в середньому показує в 2.5 рази кращі результати значення стійкості для випадкової та середньої атаки та в 1.7 разів кращі результати для популярної атаки, якщо застосовувалася до методу колаборативної фільтрації на основі сусідства та до розробленого гібридного методу. Для методу на основі матричної факторизації вона показує практично такі ж результати як у існуючого методу на основі кластеризації профілів користувачів за їх статистичними даними.

Також у даному розділі здійснене обґрунтування достовірності одержаних результатів наукових досліджень.

ОСНОВНІ ВИСНОВКИ

В дисертаційній роботі вирішена науково-практична проблема підвищення точності пропозицій рекомендаційних систем в умовах дестабілізуючих факторів у комп'ютерних мережах на основі розробки моделей та методів синтезу підсистеми забезпечення стійкості.

Проведено дослідження та порівняльний аналіз моделей та методів рекомендаційних систем соціальних мереж та контент-орієнтованих веб-сервісів, який показав, що переважна більшість існуючих моделей і методів вразливі до дії внутрішніх та зовнішніх дестабілізуючих факторів у комп'ютерних мережах. Показано, що забезпечення стійкості рекомендаційних систем до дії дестабілізуючих факторів є важливою умовою для підвищення точності їх роботи.

Основні наукові та практичні результати дисертаційної роботи:

1. Розроблено метод визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах з використанням математичного апарату марківських та напівмарківських процесів, що дає можливість встановлення зв'язку між набором щільності розподілу випадкових тривалостей перебування системи у цих станах і функціями опису динаміки ймовірностей станів для визначення ймовірностей перебування конкретної рекомендаційної системи в своїх можливих станах в довільний момент часу. На основі запропонованого методу було розроблено методику отримання аналітичних співвідношень для розрахунку ймовірностей перебування системи у своїх можливих станах в довільний момент часу.

2. Розроблено математичну модель стійкої рекомендаційної системи на основі запропонованого методу визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило здійснити оптимізацію загальних витрат на обслуговування системи в умовах

внутрішніх дестабілізуючих факторів. На основі розробленої моделі запропоновано спосіб визначення повних витрат підсистеми збору та перерахунку вхідних даних, а також спосіб визначення оптимальної частоти перерахунку вхідних даних, при яких система має мінімальну збитковість.

3. Удосконалено метод колаборативної фільтрації, який відрізняється від існуючих використанням продукційних правил для визначення подоби користувачів та використанням показників активності користувачів для формування рекомендацій, що дозволило підвищити стійкість системи у випадку недостатньої кількості вхідних даних та під час холодного старту. Розроблено та реалізовано відповідні алгоритми, використання яких дозволило в 1.6 разів підвищити стійкість системи в порівнянні з відомим методом колаборативної фільтрації на основі моделі сусідства та в 3.2 разів – в порівнянні з відомим методом колаборативної фільтрації на основі факторизації матриць.

4. Розроблено математичну модель підсистеми інформаційної безпеки стійкої рекомендаційної системи на основі запропонованого методу визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило визначити оптимальну частоту перевірки на наявність інформаційної атаки та профілів ботів. Розроблено методику отримання аналітичних співвідношень для розрахунку ймовірностей перебування підсистеми інформаційної безпеки в своїх можливих станах в довільний момент часу. На основі запропонованої математичної моделі розроблено спосіб визначення повних витрат, що зазнає рекомендаційна система внаслідок моніторингу власної інформаційної безпеки, нейтралізації діяльності бот-мереж та внаслідок інформаційних атак ін’єкцією профілів. Також розроблено спосіб визначення оптимальної частоти перевірки рекомендаційної системи на наявність інформаційної атаки та профілів ботів для оптимізації загальних витрат системи.

5. Розроблено метод імітаційного програмного моделювання користувачів та об’єктів рекомендаційної системи соціальної мережі або веб-

ресурсу на основі існуючих і розроблених методів моделювання структури складних мереж та методів моделювання поведінки користувачів, що дозволило генерувати вхідні дані для тестування якості роботи алгоритмів формування рекомендацій. Розроблено спосіб моделювання змін вподобань у часі користувачів системи, що дозволяє генерувати тестові набори даних, більш схожі за статистичними характеристиками на реальні. Спосіб засновано на математичній моделі нециклічних змін вподобань користувачів у часі з використанням експоненційного закону розподілу та закону радіоактивного розпаду. Розроблено та реалізовано відповідні алгоритми та програмну імітаційну модель користувачів, об'єктів, бот-мереж та інформаційних процесів у рекомендаційній системі. За допомогою розробленої програмної моделі сформовано набори даних для тестування методів рекомендаційних систем та підсистем їх інформаційної безпеки.

6. Розроблено метод виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів рейтингів об'єктів, що дозволило знизити кількість витрат на моніторинг безпеки системи за рахунок зняття необхідності пошуку ботів при відсутності ознак атаки. Розроблено та реалізовано відповідні алгоритми, які дозволили виявляти 76% об'єктів атак у рекомендаційній системі. Це дозволило при пошуку бот-мереж перевіряти не всі профілі системи, як у відомих методах, а тільки ті, які взаємодіяли з ймовірними цілями атаки.

7. Розроблено метод виявлення бот-мереж у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів, що дозволило виявляти бот-мережі та розрізняти їх за множинами об'єктів атаки. Також розроблено спосіб ідентифікації окремих профілів ботів на основі нейронних мереж для уточнення результатів запропонованого методу. Розроблено відповідні алгоритми та реалізовано підсистему інформаційної безпеки рекомендаційної системи. Розроблена підсистема інформаційної безпеки дозволяє забезпечити вищу стійкість системи до зовнішніх дестабілізуючих факторів на відміну від існуючих методів. Вона дозволяє підвищити стійкість

системи в середньому у 2.5 рази до випадкової та середньої атаки та в 1.7 разів – до популярної атаки, якщо застосовується разом з методом колаборативної фільтрації на основі моделі сусідства або запропонованим гібридним методом колаборативної фільтрації.

Проведена оцінка достовірності та ефективності запропонованих методів і моделей підвищення стійкості рекомендаційних систем.

Практичне значення отриманих результатів підтверджено відповідними актами впровадження. Результати дисертації впроваджені і використовуються у діяльності Компанії «Line Up», Державного підприємства «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості», Державного підприємства «Харківський науково-дослідний інститут технологій машинобудування», Національного наукового центру «Інститут судових експертиз ім. Засл. проф. М.С. Бокаріуса», а також використано у навчальному процесі Центральноукраїнського національного технічного університету та Національного технічного університету «Харківський політехнічний інститут».

Таким чином, сукупність отриманих у дисертаційній роботі наукових результатів і одержана в ході проведення експериментальних досліджень оцінка їх ефективності, дозволяють вважати сформульовану наукову проблему підвищення точності пропозицій рекомендаційних систем в умовах дестабілізуючих факторів у комп’ютерних мережах на основі розробки моделей та методів синтезу підсистеми забезпечення стійкості – вирішеною, а поставлену мету підвищення стійкості рекомендаційних систем соціальних мереж та веб-сервісів до внутрішніх та зовнішніх дестабілізуючих факторів у комп’ютерних мережах – досягнутою.

СПИСОК ЛІТЕРАТУРИ

1. Agrawal R., Srikant R. Fast Algorithms for mining association rules // Proc. Int. Conf. on Very Large Databases. – 1994. – P. 487-499.

2. Albert R., Barabási A.-L. Statistical mechanics of complex networks (англ.) // Reviews of Modern Physics, Vol. 74. – 2002. – P. 47-97. – [Electronic resource] – Access mode: doi:10.1103/RevModPhys.74.47

3. Anis A.A., Lloyd E.H. The expected value of the adjusted rescaled Hurst range of independent normal summands // Biometrica 63. – 1976. – P. 283-298.

4. Artemenko O., Kunanets O., Pasichnyk V. E-tourism recommender systems: a survey and development perspectives. Econtechmod an international quarterly journal, Vol. 6. No. 2. – 2017. – P. 91-95.

5. Barabási A.-L. Network science // Cambridge University Press. – 2018. – 475 p. – [Electronic resource] – Access mode: http://networksciencebook.com/

6. Barabási A.-L., Albert R. Emergence of scaling in random networks (англ.) // Science, Vol. 286, No. 5439. – 1999. – P. 509-512. – [Electronic resource] – Access mode: https://doi.org/10.1126/science.286.5439.509

7. Barabási L.-A., Albert R., Jeong H. Diameter of the world-wide web // Nature, Vol. 401. – 1999. – P. 130-131.

8. Barabási L.-A., Albert R., Jeong H. Scale-free characteristics of random networks: the topology of the world-wide web // Physica, A281. – 2000. – 69-77.

9. Bernardi L., Kamps J., Kiseleva J., Mueller M.J.I. The Continuous Cold Start Problem in e-Commerce Recommender Systems. – 2015. – 6 p. – [Electronic resource] – Access mode: https://arxiv.org/abs/1508.01177

10. Bollobás B., Borgs C., Chayes T., Riordan O.M. Directed scale-free graphs // ProceedingSODA '03 Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms. – 2003. – P. 132-139.

11. Bollobás B., Riordan O. Mathematical results on scale-free random graphs // Handbook of graphs and networks. – Weinheim: Wiley-VCH. – 2003. – P. 1-34.

12. Breese S., Heckerman D., Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. // In Proc. of the 14th Conference on Uncertainty in Artificial Intelligence, Volume 461, San Francisco, CA. – 1998. – P. 43-52.

13. Brin S., Motwani R., Ullman J.D., Tsur S. Dynamic itemset counting and implication rules for market basket data // In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'97). – 1997. – P. 255-264.

14. Brownlee J. Overfitting and underfitting with machine learning algorithms. – 2016. – [Electronic resource] – Access mode: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms

15. Burke R. Hybrid Web Recommender Systems // The Adaptive Web. Lecture Notes in Computer Science, Vol. 4321. Springer, Berlin, Heidelberg. – 2007. – C. 377-408.

16. C5.0: An Informal Tutorial. – 2019. – [Electronic resource] – Access mode: https://www.rulequest.com/see5-unix.html

17. Campos P.G., Díez F., Cantador I. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols // User Model User-Adap Inter 24. – 2014. – P. 67-119. – [Electronic resource] – Access mode: https://doi.org/10.1007/s11257-012-9136-x

18. Çano E., Morisio M. Hybrid recommender systems: A systematic literature review // Journal Intelligent Data Analysis, Vol. 21, No. 6. – 2017. – P. 1487-1524. – [Electronic resource] – Access mode: URL: https://arxiv.org/abs/1901.03888

19. Cao J., Hu H., Luo T., Wang J., Huang M., Wang K., Wu Z., Zhang X. Distributed Design and Implementation of SVD++ Algorithm for E-commerce Personalized Recommender System // Embedded System Technology. ESTC 2015. Communications in Computer and Information Science, Vol. 572. Springer, Singapore. – 2015. – P. 30-44. – [Electronic resource] – Access mode: http://doi-

org-443.webvpn.fjmu.edu.cn/10.1007/978-981-10-0421-6_4

20. Cao X.R. Optimization of average rewards of time nonhomogeneous Markov chains // IEEE Transactions on Automatic Control, Vol. 60(7). – 2015. – P. 1841-1856.

21. Castells P., Vargas S., Wang J. Novelty and Diversity Metrics for Recommender Systems: Choice, Discovery and Relevance. – 2011. – 8 p. – [Electronic resource] – Access mode: https://www.semanticscholar.org/paper/Novelty-and-Diversity-Metrics-for-Recommender-and-Castells-Vargas/4ec6bd672aaaa075b42a751099eb9317857e6e0c

22. Chirita P.A., Nejdl W., Zamfir C. Preventing shilling attacks in online recommender systems // In Proceedings of the ACM Workshop on Web Information and Data Management. – 2005. – P. 67-74.

23. Dimitrakos T.D., Kyriakidis E.G. A semi-Markov decision algorithm for the maintenance of a production system with buffer capacity and continuous repair times // International Journal of Production Economics, Vol. 111(2). – 2008. – P. 752-762.

24. Ding Y., Li X. Time weight collaborative filtering // In Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM ’05). Association for Computing Machinery, New York, NY, USA. – 2005. – P. 485-492. – [Electronic resource] – Access mode: https://doi.org/10.1145/1099554.1099689

25. Erdös P. and Rényi A. On the evolution of random graphs // Publication of the Mathematical Institute of the Hungarian Academy of Sciences, Vol. 5. – 1960. – P. 17-61.

26. Feinberg E.A., Yang F. Optimal pricing for a GI/M/k/N queue with several customer types and holding costs // Queueing Systems, 82(1-2). – 2016. – P. 103-120.

27. Funk S. Netflix Update: Try This at Home // The Evolution of Cybernetics – A Journal by Simon Funk. – 2006. – [Electronic resource] – Access

mode: https://sifter.org/~simon/journal/20061211.html

28. Gunes I., Kaleli C., Bilge A., Polat H. Shilling attacks against recommender systems: a comprehensive survey // Artificial Intelligence Review, Vol. 42. – 2014. – P. 767-799. – [Electronic resource] – Access mode: https://doi.org/10.1007/s10462-012-9364-9

29. Haidai B., Artiukh R., Malyeyeva O. Analysis and modelling the preferences of social networks users // Innovative technologies and scientific solutions for industries, No 1 (3). – 2018. – P. 5-12. – [Electronic resource] – Access mode: DOI: https://doi.org/10.30837/2522-9818.2018.3.005

30. Harper F.M., Konstan J.A. The MovieLens Datasets: History and Context // ACM Transactions on Interactive Intelligent Systems (TiiS). – 2015. – 19 p. – [Electronic resource] – Access mode: https://doi.org/10.1145/2827872

31. He J., Chu W.W. A Social Network-Based Recommender System (SNRS) // In: Memon N., Xu J., Hicks D., Chen H. (eds) Data Mining for Social Network Data. Annals of Information Systems, vol 12. Springer, Boston, MA. – 2010. – 32 p. – [Electronic resource] – Access mode: https://doi.org/10.1007/978-1-4419-6287-4_4

32. Hill K. Facebook recommended that this psychiatrist's patients friend each other // News web-site Splinternews.com. – 2016. – [Electronic resource] – Access mode: https://splinternews.com/facebook-recommended-that-this-psychiatrists-patients-f-1793861472

33. Hill K. Facebook says it did 'a test' last year using people's locations to make friend suggestions // News web-site Splinternews.com. – 2016. – [Electronic resource] – Access mode: https://splinternews.com/facebook-says-it-did-a-test-last-year-using-peoples-loc-1793857952

34. Huba G. The Cold Start Problem for Recommender Systems // Yuspify web-site of the machine-learning-based recommendations system for web-stores. – 2015. – [Electronic resource] – Access mode: https://www.yuspify.com/blog/cold-start-problem-recommender-systems/

35. Jia Y., Zhang C., Lu Q., Wang P. Users' brands preference based on

SVD++ in recommender systems // IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA). – 2014. – [Electronic resource] – Access mode: P. 1175-1178. https://doi.org/10.1109/wartia.2014.6976489

36. Jones M. Recommender systems, Part 1. Introduction to approaches and algorithms. Learn about the concepts that underlie web recommendation engines // Official Web-site of IBM company. – 2013. – [Electronic resource] – Access mode: https://www.ibm.com/developerworks/opensource/library/os-recommender1/index.html?S_TACT=105AGX99&S_CMP=CP

37. Jones M. Recommender systems, Part 2. Introducing open source engines. Explore software for building a recommendation capability // Official Web-site of IBM company. – 2013. – [Electronic resource] – Access mode: https://www.ibm.com/developerworks/library/os-recommender2/index.html?S_TACT=105AGX99&S_CMP=CP

38. Jøsang A., Ismail R., Boyd C. A survey of trust and reputation systems for online service provision // Decision Support Systems, Volume 43, Issue 2. – 2007. – P. 618-644. – [Electronic resource] – Access mode: https://doi.org/10.1016/j.dss.2005.05.019

39. Kaminskas M., Bridge D. Measuring Surprise in Recommender Systems // In: Workshop on recommender systems evaluation: dimensions and design (REDD 2014), October 10, 2014, Silicon Valley, USA. – 2014. – 6 p.

40. Kaur P., Goel S. Shilling attack models in recommender system // International Conference on Inventive Computation Technologies (ICICT), Coimbatore. – 2016. – P. 1-5. – [Electronic resource] – Access mode: https://ieeexplore.ieee.org/document/7824865/

41. Koren Y. Factorization meets the neighbor-hood // Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. – 2008. – P. 426-434. – [Electronic resource] – Access mode: https://doi.org/10.1145/1401890.1401944

42. Koren Ye. Collaborative filtering with temporal dynamics // Proceeding KDD '09 Proceedings of the 15th ACM SIGKDD international conference on

Knowledge discovery and data mining. – 2009. – P. 447-456.

43. Kotkov D., Konstan J.A., Zhao Q., Veijalainen J. Investigating Serendipity in Recommender Systems Based on Real User Feedback // In Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC, Pau, France. – 2018. – P. 1341-1350. – [Electronic resource] – Access mode: https://dl.acm.org/doi/10.1145/3167132.3167276

44. Krupnik I. Decomposition of a monic matrix polynomial into a product of linear factors // Linear Algebra Appl. – 1992. – P. 239-242.

45. Kumar R., Raghavan P., Rajagopalan S., Sivakumar D., Tomkins A., Upfal E. 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. – 2000. – P. 57-65. – [Electronic resource] – Access mode: https://doi.org/10.1109/SFCS.2000.892065

46. Kumari T., Punam B. A Comprehensive Study of Shilling Attacks in Recommender Systems // IJCSI International Journal of Computer Science Issues, Volume 14, Issue 4. – 2017. – [Electronic resource] – Access mode: https://www.ijcsi.org/articles/A-comprehensive-study-of-shilling-attacks-in-recommender-systems.php

47. Lam S.K., Riedl J. Shilling recommender systems for fun and profit // In Proceedings of the 13th International World Wide Web Conference. – 2004. – P. 393-402. – [Electronic resource] – Access mode: https://dl.acm.org/doi/10.1145/988672.988726

48. Li Q.L. Nonlinear Markov processes in big networks // Special Matrices, Vol. 4(1). – 2016. – P. 202-217.

49. Li Q.L., Lui J.C.S. Block-structured supermarket models // Discrete Event Dynamic Systems, Vol. 26(2). – 2016. – P. 147-182.

50. Lin Z., Chen H. Recommendation over time: a probabilistic model of time-aware recommender systems // Science China Information Sciences volume 62, Article number 212105. – 2019. – [Electronic resource] – Access mode: https://doi.org/10.1007/s11432-018-9915-8

51. Liu N.N., Zhao M., Xiang E.W., Yang Q. Online evolutionary collaborative filtering // In Proceedings of the fourth ACM conference on Recommender systems (RecSys '10), Association for Computing Machinery, New York, NY, USA. – 2010. – P. 95-102.

52. Meier A., Kaufmann M. SQL & NoSQL Databases // Springer Vieweg, Wiesbaden. – 2019. – 229 p. – [Electronic resource] – Access mode: https://doi.org/10.1007/978-3-658-24549-8

53. Meleshko Ye. Computer model of virtual social network with recommendation system // Innovative technologies and scientific solutions for industries. – 2019. – №2(8). – P. 80-85.

54. Meleshko Ye. Method of collaborative filtration based on associative networks of users similarity // Advanced information systems. – 2018. – T. 2, № 4. – P. 55-59.

55. Meleshko Ye. Method of generating recommendations lists with considering activity indexes of users in a recommendation system // Advanced information systems. – 2019. – T. 3, № 1. – P. 43-47.

56. Meleshko Ye., Drieiev O., Drieieva H. Method of identification bot profiles based on neural networks in recommendation systems // Advanced Information Systems. – 2020. – Vol. 4, No. 2 – P. 24-28.

57. Meleshko Ye., Drieiev O., Yakymenko M., Lysytsia D. Developing a model of the dynamics of states of a recommendation system under conditions of profile injection attacks // Eastern-European Journal of Enterprise Technologies (ISSN 1729-3774). 2020. – Vol. 4, No 4(106). – P. 14-24 (SCOPUS).

58. Meleshko Ye., Raskin L., Semenov S., Sira O. Methodology of probabilistic analysis of state dynamics of multi-dimensional semi-Markov dynamic systems // Eastern-European Journal of Enterprise Technologies (ISSN: 1729-3774). 2019. – Vol. 6, No 4(102). – P. 6-13 (SCOPUS).

59. Mitchell T.M. Machine Learning // McGraw-Hill Education Ltd. – 1997. – 414 p.

60. Mobasher B., Burke R., Bhaumik R., Williams C. Effective attack

models for shilling item-based collaborative filtering system // In Proceedings of the WebKDD Workshop, held in conjuction with ACM SIGKDD 2005, Chicago, Illinois. – 2005. – 8 p.

61. Mobasher B., Burke R., Bhaumik R.,Williams C. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness // ACM Transactions on Internet Technology, Vol. 7(4). – 2007. – 41 p. – [Electronic resource] – Access mode: https://doi.org/10.1145/1278366.1278372

62. Mobasher B., Burke R.D., Sandvig J.J. Model-based collaborative filtering as a defense against profile injection attacks // In Proceedings of the 21st national conference on Artificial intelligence, Vol. 2 (AAAI'06). AAAI Press. – 2006. – P. 1388-1393. – [Electronic resource] – Access mode: https://dl.acm.org/doi/10.5555/1597348.1597409

63. Mohammadi V., Rahmani A.M., Darwesh A.M., Sahafi A. Trust-based recommendation systems in Internet of Things: a systematic literature review // Human-centric Computing and Information Sciences. – 2019. – 61 p. – [Electronic resource] – Access mode: https://doi.org/10.1186/s13673-019-0183-8

64. Mohammed A.S., Meleshko Y., Balaji S.B., Semenov S. Collaborative filtering method with the use of production rules // Proceedings of ICCIKE, Amity University Dubai; United Arab Emirates. – 2019. – c. 387-391 (SCOPUS).

65. Narayanan V., Arora I., Bhatia A. Fast and accurate sentiment classification using an enhanced naive bayes model // Intelligent Data Engineering and Automated Learning (IDEAL), Lecture Notes in Computer Science, Vol. 8206, Springer, Berlin, Heidelberg. – 2013. – C. 194-201. – [Electronic resource] – Access mode: https://doi.org/10.1007/978-3-642-41278-3_24

66. Neo4j Documentation // Official website of the graph database Neo4j. – [Electronic resource] – Access mode: https://neo4j.com/docs/

67. Neumaier A. Solving ill-conditioned and singular lin-ear systems: A tutorial on regularization // SIAM Review, Vol. 40. – 1998. – P. 636-666.

68. Newman M., Barabási A.-L., Watts D. J. The Structure and dynamics of networks // Princeton University Press. – 2006. – P. 592.

69. O'Mahony M.P., Hurley N.J., Silvestre G.C.M. Promoting recommendations: An attack on collaborative filtering // DEXA, Lecture Notes in Computer Science, Vol. 2453. – 2002. – P. 494-503.

70. Okamura H., Miyata S., Dohi T. A Markov decision process approach to dynamic power management in a cluster system // IEEE Access, 3. – 2015. – P. 3039-3047.

71. Overfitting in machine learning: what it is and how to prevent it // Web-site EliteDataScience. – 2017. – [Electronic resource] – Access mode: https://elitedatascience.com/overfitting-in-machine-learning

72. Ozsoy M.G., Polat F. Trust based recommendation systems // Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. – 2013. – P. 1267-1274.

73. Park J.S., Chen M.-S., Philip S.Y. An Effective HashBased Algorithm for Mining Association Rules // Association for Computing Machinery, Vol. 24, No. 2. – 1995. – P. 175–186. – [Electronic resource] – Access mode: https://doi.org/10.1145/568271.223813

74. Patel D., Saxena S., Verma T. Sentiment Analysis using Maximum Entropy Algorithm in Big Data // International Journal of Innovative Research in Science, Engineering and Technology. – 2016. – №5. – P. 8355-8361.

75. Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J. Grouplens: An open architecture for collaborative filtering of netnews // Proceedings of the ACM Conference on Computer Supported Cooperative Work. – 1994. – P. 175-186.

76. Ricci F., Rokach L., Shapira B., Kantor P.B. (Editors) Recommender Systems Handbook // Boston: Springer. – 2011. – 842 p. – [Electronic resource] – Access mode: https://doi.org/10.1007/978-0-387-85820-3

77. Sanajian N., Abouee-Mehrizi H., Balcıoglu B. Scheduling policies in the M/G/1 make-to-stock queue // Journal of the Operational Research Society, Vol. 61(1). – 2010. – P. 115-123. – [Electronic resource] – Access mode: https://doi.org/10.1057/jors.2008.139

78. Sarwar B., Karypis G., Konstan J., Riedl J. Item-based collaborative filtering recommendation algorithms // Proceedings of the Tenth International World Wide Web Conference. – 2001. – P. 285-295.

79. Savasere A., Omiecinski E., Navathe S. An Efficient Algorithm for Mining Association Rules in Large Databases // In Proc. 21st Int'l Conf. Very Large Data Bases, Morgan Kaufmann, San Francisco. – 1995.

80. Schmid H. Probabilistic part-of-speech tagging using decision trees // In International Conference on New Methods in Language Processing, Manchester. – 1994. – P. 44-49.

81. Stekh Y., Artsibasov V. Adaptive clustering algorithm for recommender systems // Вісник НУ "Львівська політехніка": Комп'ютерні системи проектування. Теорія і практика, № 747. – 2012. – С. 75-78.

82. Stekh Y., Logvinenko A., Lobur M., Artsibasov V. Model and methods for building Web recommendation systems // Proc. of the VIth International Conference on Computer Science and Information Technologies (CSIT 2011). – Lviv, 2011. – P. 314-316.

83. Stone P., Hunt E. A computer approach to content analysis: Studies using the general inquirer system. // Spring Joint Computer Conference, AFIPS '63, New York: ACM. – 1963. – P. 241–256.

84. Su X. and Khoshgoftaar T.M. A Survey of Collaborative Filtering Techniques A Survey of Collaborative Filtering Techniques // Hindawi Publishing Corporation, Advances in Artificial Intelligence archive, USA. – 2009. – P. 1-19. – [Electronic resource] – Access mode: https://www.hindawi.com/journals/aai/2009/421425/

85. Taboada M., Brooke J., Tofiloski M., Voll K., Stede M. Lexicon-based methods for sentiment analysis // Computational linguistics, Volume 37, Issue 2. – 2011. – P. 267-307. – [Electronic resource] – Access mode: https://doi.org/10.1162/COLI_a_00049

86. TensorFlow tutorials – 2020. – [Electronic resource] – Access mode: https://www.tensorflow.org/tutorials/

87. Töscher A., Jahrer M., Bell R.M. The BigChaos So-lution to the Netflix Grand Prize // Netflix prize documentation. – 2009. – [Electronic resource] – Access mode: https://www.netflixprize.com/assets/ GrandPrize2009_BPC_BigChaos.pdf

88. Traag V.A. Algorithms and Dynamical Models for Communities and Reputation in Social Networks // Springer International Publishing. – 2014. – P. 229. – [Electronic resource] – Access mode: https://doi.org/10.1007/978-3-319-06391-1

89. Ulichev O., Meleshko Y., Khokh V. The computer simulation method of a social network structure for the research of dissemination processes of informational influences // Scientific and Practical Cyber Security Journal (SPCSJ) 4(3). – Georgia, Tbilisi, 2019. – P. 34-47.

90. Ulichev O., Meleshko Ye., Smirnov O., Khokh V., Goncharenko Iu. Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process // CEUR-WS (ISSN: 1613-0073), Vol 2588, Lviv, Ukraine. – 2019. – P. 215-227 (SCOPUS).

91. Ulichev O.S., Meleshko Ye.V., Sawicki D., Smailova S. Computer modeling of dissemination of informational influences in social networks with different strategies of information distributors // Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, Wilga, Poland (ISSN: 0277-786X). – 2019. – 111761T (SCOPUS).

92. Valois B.Jr.C., Oliveira M.A. Recommender systems in social networks // JISTEM J.Inf.Syst. Technol. Manag., Vol.8 No.3. – 2011. – P. 681-716. – [Electronic resource] – Access mode: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S1807-17752011000300009

93. Watts D.J., Strogatz S.H. Collective dynamics of “small-world” networks // Nature, Vol. 393(6684). – 1998. – P. 440-442. – [Electronic resource] – Access mode: https://www.nature.com/articles/30918

94. Wei S., Ye N., Zhang Q. Time-Aware Collaborative Filtering for Recommender Systems // Pattern Recognition, Communications in Computer and Information Science (CCPR 2012), Vol. 321, Springer, Berlin, Heidelberg. – 2012. – [Electronic resource] – Access mode: https://doi.org/10.1007/978-3-642-33506-8_81

95. Weisberg J. Bubble Trouble: Is Web personalization turning us into solipsistic twits? – 2011. – [Electronic resource] – Access mode: https://slate.com/news-and-politics/2011/06/eli-pariser-s-the-filter-bubble-is-web-personalization-turning-us-into-solipsistic-twits.html

96. What is Selenium? – [Electronic resource] – Access mode: http://docs.seleniumhq.org/

97. Wiki for project Gephi on GitHub // Web-site GitHub. – 2019. – [Electronic resource] – Access mode: https://github.com/gephi/gephi/wiki

98. Williams A. C., Mobasher B., Burke R. Defending recommender systems: detection of profile injection attacks // Service Oriented Computing and Applications. – 2007. – P. 157–170.

99. Yao Y.Y. Measuring retrieval effectiveness based on user preference of documents // Journal of the American Society for Information Science. – 1995. – №46. – P. 133-145.

100. Yuan Q., Cong G., Ma Z., Sun A., Magnenat-Thalmann N. Time-aware point-of-interest recommendation // In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '13). Association for Computing Machinery, New York, NY, USA. – 2013. – P. 363-372. – [Electronic resource] – Access mode: https://doi.org/10.1145/2484028.2484030

101. Zhang C., Liu J., Qu Y., Han T., Ge X., Zeng A. Enhancing the robustness of recommender systems against spammers // PLoS ONE 13(11): e0206458. – 2018. – [Electronic resource] – Access mode: https://doi.org/10.1371/journal.pone.0206458

102. Zhang Y.C., Séaghdha D.Ó., Quercia D., Jambor T. Auralist:

Introducing Serendipity into Music Recommendation // WSDM '12: Proceedings of the fifth ACM international conference on Web search and data mining. – 2012. – P. 13-22. – [Electronic resource] – Access mode: https://doi.org/10.1145/2124295.2124300

103. Zhou W., Wen J., Koh Y.S., Alam S., Dobbie G. Attack detection in recommender systems based on target item analysis // International Joint Conference on Neural Networks (IJCNN 2014), Beijing. – 2014. – P. 332-339. – [Electronic resource] – Access mode: https://ieeexplore.ieee.org/document/6889419

104. Zhou W., Wen J., Qu Q., Zeng J., Cheng T. Shilling attack detection for recommender systems based on credibility of group users and rating time series // PLoS ONE 13(5): e0196533. – 2018. – [Electronic resource] – Access mode: https://doi.org/10.1371/journal.pone.0196533

105. Амелькин С.А. Оценка эффективности рекомендательных систем // Труды 14-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL-2012), Переславль-Залесский, 15-18 октября 2012 г. – 2012. – С. 288-291.

106. Балдин К.В., Башлыков В.Н., Рукосуев А.В. Основы теории вероятностей и математической статистики: учебник. 4-е издание. – М.: Издательство "ФЛИНТА". – 2016. – 489 с.

107. Баруча-Рид А.Т. Элементы теории марковских процессов и их приложения: пер. с англ. – М.: Наука, 1969. – 320 с.

108. Батура Т.В. Модели и методы анализа компьютерных социальных сетей // Программные продукты и системы, №3. – 2013. – С. 130-137. – [Электронный ресурс] – Режим доступа: https://cyberleninka.ru/article/n/modeli-i-metody-analiza-kompyuternyh-sotsialnyh-setey

109. Берж К. Теория графов и ее приложения. – М.: ИЛ. – 1962. – 320 с.

110. Берновский М.М., Кузюрин Н.Н. Случайные графы, модели и генераторы безмасштабных графов // Труды Института системного

программирования РАН, Том 22. – 2012. – С. 419-432. – [Электронный ресурс] – Режим доступа: https://cyberleninka.ru/article/n/ sluchaynye-grafy-modeli-i-generatory-bezmasshtabnyh-grafov

111. Богуш В.М., Юдін О.К. Інформаційна безпека держави. – К.: "МК-Прес". – 2005. – 432 с.

112. Болотова Л.С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях. – Москва: «Финансы и Статистика». – 2012. – 663 с.

113. Булинский А.Н., Ширяев А.Н. Теория случайных процессов – М.: Физматгиз. – 2005. – 364 с.

114. Вентцель Е.С. Теория вероятностей (10-е изд.). – Москва: Высшая школа. – 2006. – 575 с.

115. Вуколов Э.А. Основы статистического анализа. Практикум по статистическим методам и исследованию операций с использованием пакетов STATISTICA и EXCEL (2-е изд., испр. и доп.). – Москва: Форум. – 2011 . – 463 с.

116. Глибовець М.М., Олецький О.В. Штучний інтелект: Підручник. – К.: Вид. дім "КМ Академія". – 2002. – 366 с.

117. Гмурман В.Е. Теория вероятностей и математическая статистика (12-е изд.). – Москва: Юрайт. – 2016 . – 479 с.

118. Горбулін В.П., Додонов О.Г., Ланде Д.В. Інформаційні операції та безпека суспільства: загрози, протидія, моделювання: монографія. – К.: Інтертехнологія. – 2009. – 164 с.

119. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Модели влияния в социальных сетях // Управление большими системами, №27. – 2009. – С. 205-281. – [Электронный ресурс] – Режим доступа: http://cyberleninka.ru/article/n/modeli-vliyaniya-v-sotsialnyh-setyah

120. Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети: модели информационного влияния, управления и противоборства. – М.: Издательство физико-математической литературы. – 2010. – 228 с.

121. Гусарова Н.Ф. Анализ социальных сетей. Основные понятия и метрики. – СПб: Университет ИТМО. – 2016. – 67 с.

122. Даниленко Д.О., Смірнов О.А., Мелешко Є.В. Дослідження методів виявлення вторгнень в телекомунікаційні системи та мережі // Системи озброєння і військова техніка. – 2012. – № 1. – С. 92-100.

123. Дистель Р. Теория графов. – Новосибирск: ИМ. – 2002. – 336 с.

124. Домарев В.В. Безопасность информационных технологий. Методология создания систем защиты. – К.: ТИД ДиаСофт, 2002. – 688 с.

125. Дрєєв О.М., Смірнов О.А., Мелешко Є.В., Коваленко О.В. Метод прогнозування завантаженості серверу телекомунікаційної мережі // Системи обробки інформації. – 2012. – Вип. 3(2). – С. 181-187.

126. Дынкин Е.Б. Марковские процессы. – М.: Физматгиз. – 1963. – 482 с.

127. Евин И.А. Введение с теорию сложных сетей // Компьютерные исследования и моделирование, Том 2, № 2. 2010. – С. 121-141.

128. Ильин В.А. Основы математического анализа. – М.: Наука. – 1965. – 572 с.

129. Калуш Ю.А., Логинов В.М. Показатель Хёрста и его скрытые свойства // Сибирский журнал индустриальной математики, Т. 5, Вып. 4. – 2002. – С. 29-37.

130. Кемени Дж., Снелл Дж. Конечные цепи Маркова. – М.: Наука. – 1970. – 198 с.

131. Климов А.Н. Ядерная физика и ядерные реакторы. – М.: Энергоатомиздат. – 1985. – 352 с.

132. Коломейченко М.И., Поляков И.В., Чеповский А.А., Чеповский А.М. Выделение сообществ в графе взаимодействующих объектов // Фундаментальная и прикладная математика, Т. 21, № 3. – 2016. – С. 131-139.

133. Кочкаров А.А, Сенникова Л.И., Кочкаров Р.А. Некоторые особенности применения динамических графов для конструирования алгоритмов взаимодействия подвижных абонентов // Известия ЮФУ,

Технические науки, №1(162). – 2015. – [Электронный ресурс] – Режим доступа: https://cyberleninka.ru/article/n/nekotorye-osobennosti-primeneniya-dinamicheskih-grafov-dlya-konstruirovaniya-algoritmov-vzaimodeystviya-podvizhnyh-abonentov

134. Краснов М.Л. Интегральные уравнения. – М.: Наука. – 1985. – 476 с.

135. Курбан О.В. Сучасні інформаційні війни в соціальних онлайн-мережах // Інформаційне суспільство, Вип. 23. – 2016. – С. 85-90. – [Електронний ресурс] – Режим доступу: http://nbuv.gov.ua/UJRN/is_2016_23_15

136. Ландэ Д.В., Снарский А.А., Безсуднов И.В. Интернетика: Навигация в сложных сетях: модели и алгоритмы. – М.: Либроком (Editorial URSS). – 2009. – 264 с. – [Электронный ресурс] – Режим доступа: http://dwl.kiev.ua/art/internetica/

137. Лобур М.В., Стех Ю.В., Шварц М.Є. Побудова асоціативних правил для прогнозування рекомендацій в колаборативних рекомендаційних системах // Квалілогія книги: Збірник наукових праць Української академії друкарства. Львів, № 2 (32). – 2017. – С. 82-86.

138. Лобур М.В., Шварц М.Є., Стех Ю.В. Моделі і методи прогнозування рекомендацій для колаборативних рекомендаційних систем // Вісник Національного університету «Львівська політехніка»: Інформаційні системи та мережі. Львів, № 901. – 2018. – С. 68-75.

139. Лысенко И.А., Смирнов А.А., Мелешко Е.В. Исследование уровней тестирования программного обеспечения инфотелекоммуникационных систем // Наука і техніка Повітряних Сил Збройних Сил України. – 2014. – № 4. – С. 79-81.

140. Мелешко Е.В. Метод встраивания двухуровневых цифровых водяных знаков в медиафайлы для защиты авторских прав // Збірник наукових праць Харківського університету Повітряних сил. – 2013. – Вип. 4. – С. 127-131.

141. Мелешко Е.В., Смирнов А.А. Исследование методов структурного

анализа социальных сетей с точки зрения информационной безопасности // Материалы XXI Международной научно-технической конференции "Современные средства связи", 20-21 октября 2016 года, Минск, Республика Беларусь – Минск: Белорусская государственная академия связи. – 2016. – С. 175-177.

142. Мелешко Є., Хох В., Резніченко В., Босько В. Дослідження методів оцінювання робастності рекомендаційних систем до атак накручування рейтингів // Збірник тез IV Всеукраїнської науково-практичної конференції «Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем MEICS-2019», м. Дніпро, 27-29 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 10-11.

143. Мелешко Є.В. Аналіз структури соціальної мережі з точки зору інформаційної безпеки // Збірник тез XVIII міжнародного науково-практичного семінару "Комбінаторні конфігурації та їх застосування", м. Кіровоград, 15-16 квітня 2016. – Кіровоград: Кіровоградський національний технічний університет. – 2016. – С. 93-97.

144. Мелешко Є.В. Дослідження засобів кластеризації графів у графовій СУБД Neo4j для виявлення співтовариств у соціальних мережах // Збірник тез VIII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство», смт. Чинадієво, 16-18 травня 2019 р. – Львів: Видавництво Львівської політехніки. – 2019. – С. 19-20.

145. Мелешко Є.В. Дослідження методів динамічного аналізу віртуальних соціальних мереж з точки зору інформаційної безпеки // Матеріали Всеукраїнської науково-практичної конференції "Кібербезпека в Україні: правові та організаційні питання", м. Одеса, 21 жовтня 2016 р. – Одеса: ОДУВС. – 2016. – С. 154-155.

146. Мелешко Є.В. Дослідження методів побудови рекомендаційних систем заснованих на фільтрації контенту // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп’ютерні технології», м. Кропивницький, 19-20 квітня 2018 р. – Кропивницький:

ЦНТУ. – 2018. – С. 234-237.

147. Мелешко Є.В. Дослідження проблем сучасних рекомендаційних систем та методів їх рішення // Збірник тез Міжнародної науково-практичної конференції «Контроль і управління в складних системах (КУСС-2018)», м. Вінниця, 15-17 жовтня 2018 р. – Вінниця: ВНТУ. – 2018. – С. 126.

148. Мелешко Є.В. Загрози інформаційній безпеці у рекомендаційних системах соціальних медіа // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м. Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 24-25.

149. Мелешко Є.В. Метод визначення подоби між користувачами у рекомендаційних системах з колаборативною фільтрацією // Збірник тез Науково-практичної конференції «Інформатика, математика, автоматика (ІМА-2019)», м. Суми, 23-26 квітня 2019 р. – Суми: СДУ. – 2019. – С. 213-214.

150. Мелешко Є.В. Метод побудови рекомендаційних систем на основі асоціативних мереж користувачів // Збірник тез XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня 2019 р. – Кропивницький: КЛА НАУ. – 2019. – С. 91-95.

151. Мелешко Є.В. Методи кластеризації графів для побудови рекомендаційних систем соціальних медіа // Збірник тез VII Міжнародної науково-практичної конференції «Обробка сигналів і негаусівських процесів», присвячена пам’яті професора Кунченка Ю.П., м. Черкаси, 23-24 травня 2019 р., – Черкаси: ЧДТУ. – 2019. – С. 100-102.

152. Мелешко Є.В. Методи кластеризації графів соціальних мереж для побудови рекомендаційних систем // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 2 (54). – С. 129-134.

153. Мелешко Є.В. Методи оцінки точності прогнозування вподобань користувачів веб-ресурсів рекомендаційними системами // Збірник тез X

Всеукраїнської науково-практичної конференції «Стан та удосконалення безпеки інформаційно-телекомунікаційних систем (SITS'2018)», с. Коблево, 21-23 червня 2018 р. – Миколаїв: НАУ та МІПРО. – 2018. – С. 58-61.

154. Мелешко Є.В. Методи оцінки якості роботи рекомендаційних систем // Системи управління, навігації та зв'язку. – Полтава: ПНТУ, 2018. – Вип. 5 (51). – С. 92-97.

155. Мелешко Є.В. Методи оцінки якості роботи рекомендаційних систем // Матеріали XX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 13-14 квітня 2018 р. – Кропивницький: КЛА НАУ. – 2018. – С. 68-72.

156. Мелешко Є.В. Методи протидії деструктивним інформаційним впливам в соціальних мережах в умовах інформаційної війни // Збірник тез Всеукраїнської науково-практичної конференції «Інформаційна безпека держави суспільства та особистості», м. Кіровоград, 16 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 139-142.

157. Мелешко Є.В. Проблеми сучасних рекомендаційних систем та методи їх рішення // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2018. – Т. 4 (50). – С. 120-124.

158. Мелешко Є.В. Розробка програмного забезпечення для виділення співтовариств у соціальній мережі // Збірник тез III Всеукраїнської науково-практичної конференції «Перспективні напрямки сучасної електроніки, інформатики і комп'ютерних систем», м. Дніпро, 21-23 листопада 2018 р. – Дніпро: ДНУ. – 2018. – С. 42-43.

159. Мелешко Є.В., Босько В.В., Резніченко В.А. Дослідження методів комп'ютерної лінгвістики для аналізу контенту веб-сайтів // Збірник тез XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня 2019 р. – Кропивницький: КЛА НАУ. – 2019. – С. 96-100.

160. Мелешко Є.В., Босько В.В., Резніченко В.А. Застосування асоціативних мереж для побудови рекомендаційних систем // Збірник тез

Міжнародної науково-практичної Інтернет-конференції «Автоматика, комп’ютерно-інтегровані технології та проблеми енергоефективності в промисловості і сільському господарстві (АКІТ-2018)», м. Кропивницький, 15-16 листопада 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 165-166.

161. Мелешко Є.В., Босько В.В., Резніченко В.А. Розробка рекомендаційної системи на базі СУБД Neo4j // Збірник тез V Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії (IT&I – 2018)», м. Київ, 20-21 листопада 2018 р. – Київ: КНУ. – 2018. – С. 351-352.

162. Мелешко Є.В., Гермак В.С. Дослідження впливу структури соціальної мережі на захищеність від поширення вірусної інформації // Збірник тез доповідей III Міжнародної науково-практичної конференції "Актуальні питання забезпечення кібербезпеки та захисту інформації". с. Верхнє Студене, 22-25 лютого 2017 р. – Київ: Видавництво Європейського університету. – 2017. – С. 118-119.

163. Мелешко Є.В., Дреєва Г.М. Дослідження проблем сучасних рекомендаційних систем // Збірник тез VII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство (ICS-2018)», 17-19 травня 2018 р., Україна, смт. Чинадієво. – Львів: НУ «Львівська політехніка». – 2018. – С. 31-32.

164. Мелешко Є.В., Дрєєв О.М., Дрєєва Г.М. Розробка методу ідентифікації ботів у рекомендаційних системах // Матеріали X Міжнародної науково-практичної конференції “Комплексне забезпечення якості технологічних процесів та систем”, м. Чернігів, 29-30 квітня 2020 р. – Чернігів: ЧНТУ. – 2020. – С. 165-166.

165. Мелешко Є.В., Дрєєва Г., Якименко М., Хох В. Методи моделювання складних мереж // Матеріали IX Міжнародної наукової конференції "Інформація. Комунікація. Суспільство", м. Львів, 21-23 травня 2020 р. – Львів: Видавництво Львівської політехніки. – 2020. – С. 29-30.

166. Мелешко Є.В., Дрєєва Г.М. Дезінформаційні атаки на рекомендаційні системи // Збірник тез Міжнародної наукової конференції

«Безпека в сучасному світі», м. Дніпро, 27-28 вересень 2019 р. – Дніпро: ДНУ ім. Олеся Гончара. – 2019. – С. 51-53.

167. Мелешко Є.В., Дрєєва Г.М., Гермак В.С., Резніченко В.А., Шевченко О.О. Методи визначення ботів серед користувачів соціальних мереж // Збірник тез ІІ Міжнародної науково-практичної конференції "Інформаційна безпека та інформаційні технології", м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 44.

168. Мелешко Є.В., Дрєєва Г.М., Дрєєв О.М. Метод кластеризації користувачів соціальної мережі на основі нейронних мереж // Збірник тез XXII Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 87-90.

169. Мелешко Є.В., Охотний С.М., Босько В.В. Розробка програмного забезпечення для збору та аналізу даних із соціальних мереж // Збірник тез IX Міжнародної науково-практичної конференції «Комплексне забезпечення якості технологічних процесів та систем», Т.2, м. Чернігів, 14-16 травня 2019 р. – Чернігів: ЧНТУ. – 2019. – С. 225-226.

170. Мелешко Є.В., Охотний С.М., Резніченко В.А. Дослідження методів визначення спільнот у соціальному графі // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 92-93.

171. Мелешко Є.В., Семенов С.Г., Хох В.Д. Дослідження методів побудови рекомендаційних систем в мережі Інтернет // Збірник наукових праць "Системи управління, навігації та зв'язку". Випуск 1(47). – Полтава: ПНТУ ім. Ю. Кондратюка. – 2018. – С. 131-136.

172. Мелешко Є.В., Хох В.Д. Дослідження моделей рекомендаційних систем на основі прихованих факторів // Збірник тез ІІ Міжнародної науково-

практичної конференції “Інформаційна безпека та інформаційні технології”, м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 46.

173. Мелешко Є.В., Хох В.Д., Босько В.В. Дослідження матричних факторизаційних моделей рекомендаційних систем // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 6 (58). – С. 58-62.

174. Мелешко Є.В., Хох В.Д., Минайленко Р.М. Розробка експертної системи для виявлення атак на рекомендаційні мережі // Збірник тез XVII Міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем», м. Дніпро, 20-22 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 176-177.

175. Мелешко Є.В., Хох В.Д., Сидоренко В.В. Розробка автоматизованої системи виявлення, оцінки та розробки заходів по усуненню загроз в інформаційних системах // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м. Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 38-39.

176. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження відомих моделей атак на рекомендаційні системи з колаборативною фільтрацією // Збірник наукових праць Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – №. 5 (57). – С. 67-71.

177. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження методів підвищення робастності рекомендаційних систем до інформаційних атак // Матеріали VI Міжнародної науково-практичної конференції «Актуальні питання забезпечення кібербезпеки та захисту інформації», 19-22 лютого 2020 р. – м. Київ: Вид-во Європейського університету, 2020. – С. 65-70.

178. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження робастності рекомендаційних систем з колаборативною фільтрацією до інформаційних атак // Електронне фахове наукове видання Кібербезпека: освіта, наука, техніка.– Київ: КУБГ, 2019. – Т.1, № 5. – С. 95-104.

179. Мелешко Є.В., Хох В.Д., Улічев О.С. Методи тестування

робастності рекомендаційних систем з колаборативною фільтрацією // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 88-89.

180. Мелешко Є.В., Чабан О.О. Дослідження засобів парсингу веб-сайтів // Матеріали XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня 2019 року. – Кропивницький: КЛА НАУ. – 2019. – С. 163-167.

181. Мелешко Є.В., Чабан О.О., Міхав В.В. Способи побудови рекомендаційних систем для соціальних мереж з врахуванням репутації користувачів // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 86-87.

182. Мелешко Є.В., Шингалов Д.В., Минайленко Р.М. Методи автоматизації побудови графових структур спільнот у соціальних мережах // Матеріали XIX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» присвяченого пам'яті д.ф.-м.н., професора Петренюка Анатолія Яковича, м. Кропивницький, 7-8 квітня 2017 року. – Кропивницький: КЛА НАУ. – 2017. – С. 162-164.

183. Мелешко Є.В., Якименко М.С. Методи виявлення інформаційно-емоційних впливів у текстовій інформації // Збірник тез VI Міжнародної наукової конференції «Інформація. Комунікація. Суспільство (ICS-2017)», м. Львів, 18-20 травня 2017 р. – Львів: Національний університет "Львівська політехніка". – 2017. – С. 30-31.

184. Мелешко Є.В., Якименко М.С., Резніченко В.А. Методи оцінки якості роботи алгоритмів машинного навчання // Збірник тез XXII

Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 90-94.

185. Меликов С., Мусатов Д., Савватеев А. Моделирование социальных сетей. – 2013. – [Электронный ресурс] – Режим доступа: https://kpfu.ru/docs/F117464271/MMS_socnet_cities.pdf

186. Меньшикова Н.В., Портнов И.В., Николаев И.Е. Обзор рекомендательных систем и возможностей учета контекста при формировании индивидуальных рекомендаций // ACADEMY, №6. – 2016. – С. 20-22.

187. Мерфи Дж.Дж. Технический анализ фьючерсных рынков. Теория и практика. – М.: Альпина Паблишер. – 2011. – 610 с.

188. Міхав В.В., Мелешко Є.В., Якименко М.С. Метод зберігання даних рекомендаційної системи на основі бінарних діаграм рішень // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2020. – Т. 2 (60). – С. 85-89.

189. Міхав В.В., Мелешко Є.В. Метод оптимізації швидкодії бінарних діаграм рішень при представленні даних рекомендаційної системи // Збірник тез II Міжнародної науково-практичної конференції "Інформаційна безпека та інформаційні технології", м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 17.

190. Міхав В.В., Мелешко Є.В. Порівняння стратегій редагування бінарних діаграм рішень для роботи з графовими даними // Матеріали IX Міжнародної наукової конференції "Інформація. Комунікація. Суспільство", м. Львів, 21-23 травня 2020 р. – Львів: Видавництво Львівської політехніки. – 2020. – С. 17-18.

191. Нежное введение в алгоритм оптимизации Адама для глубокого обучения // Блог Машинное обучение, нейронные сети, искусственный интеллект. – 2017. – [Электронный ресурс] – Режим доступа: https://www.machinelearningmastery.ru/adam-optimization-algorithm-for-deep-learning/

192. Никишин Е.С. Методы выделения сообществ в социальных графах – 2016. – [Электронный ресурс] – Режим доступа: http://www.machinelearning.ru/

193. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. – СПб.: Питер. – 2018. – 480 с.

194. Новиков О.В. Методы ускорения работы рекомендательных систем для высоконагруженных веб-сайтов // Прикладная информатика, №5(47). – 2013. – Р. 29-34. – [Электронный ресурс] – Режим доступа: https://cyberleninka.ru/article/n/metody-uskoreniya-raboty-rekomendatelnyh-sistem-dlya-vysokonagruzhennyh-veb-saytov

195. Остапенко Г.А. Информационные операции и атаки в социотехнических системах. Учебное пособие для вузов / Под ред. чл.-корр. РАН В.И. Борисова. – М.: Горячая линия – Телеком. – 2007. – 134 с.

196. Охотний С.М., Мелешко Є.В. Визначення центральностей у соціальному графі засобами графової бази даних Neo4j // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 247-248.

197. Охотний С.М., Мелешко Є.В. Збирання даних про користувачів віртуальної соціальної мережі за допомогою web-кроулера // Збірник тез II Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький, 20-22 квітня 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 157-159.

198. Охотний С.М., Мелешко Є.В., Константинова А.А. Розробка бота для соціальної мережі Facebook на основі фреймворка Selenium // Збірник тез Всеукраїнської науково-практичної Інтернет-конференції "Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті", м. Кропивницький, 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 202-203.

199. Пархоменко П.А., Григорьев А.А., Астраханцев Н.А. Обзор и

экспериментальное сравнение методов кластеризации текстов // Труды ИСП РАН, №2. – 2017. – [Электронный ресурс] – Режим доступа: https://cyberleninka.ru/article/n/obzor-i-eksperimentalnoe-sravnenie-metodov-klasterizatsii-tekstov.

200. Пасічник В.В., Артеменко О.І. Особливості побудови просторово-орієнтованих мобільних туристичних рекомендаційних систем // Матеріали XIV міжнародної конференції "Контроль і управління в складних системах (КУСС-2018)", м. Вінниця, 15-17 жовтня 2018 р. – Вінниця: ВНТУ. – 2018. – С. 68-72.

201. Пасічник В.В., Іванущак Н.М. Дослідження та моделювання складних мереж // Східно-Європейський журнал передових технологій, Вип. 2, № 3(44). – 2010. – С. 43-48.

202. Петерс Э. Фрактальный анализ финансовых рынков. Применение теории хаоса в инвестициях и экономике. – М.: Интернет-трейдинг. – 2004. – 304 с.

203. Пономарев А.В. Обзор методов учета контекста в системах коллаборативной фильтрации // Труды СПИИРАН, №7(30). – 2013. – С. 169-188.

204. Пректер Р., Фрост А. Волновой принцип Эллиотта. Ключ к пониманию рынка. – М.: Альпина Паблишер. 2012. – 269 с.

205. Пятикоп Е.Е. Исследование метода коллаборативной фильтрации на основе сходства элементов // Наукові праці Донецького національного технічного університету, №2, Сер.: Інформатика, кібернетика та обчислювальна техніка. – 2013. – С. 109-114. – [Электронный ресурс] – Режим доступа: http://nbuv.gov.ua/UJRN/Npdntu_inf_2013_2_18

206. Райгородский А.М. Математические модели Интернета // Журнал “Квант” №4, – 2012. – С. 12-16. – [Электронный ресурс] – Режим доступа: https://elementy.ru/nauchno-populyarnaya_biblioteka/431792

207. Райгородский А.М. Модели случайных графов и их применение // Труды МФТИ, Т. 2, № 4. – 2010. – С. 130-140.

208. Свешников А.Г., Тихонов А.Н. Теория функций комплексной переменной. – М.: Наука. – 1967. – 308 с.

209. Сегаран Т. Программируем коллективный разум // Пер. с англ. – СПб: Символ-Плюс. – 2013. – 368 с.

210. Смірнов О.А., Даниленко Д.О., Мелешко Є.В. Метод обнаружения вредоносного программного обеспечения. Часть 1. Корреляционный анализ сетевого трафика // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті». – 2012. – № 4(95). – С. 8-14.

211. Снарский А.А., Ландэ Д.В. Моделирование сложных сетей: учебное пособие. – К.: Инжиниринг. – 2015. – 212 с. – [Электронный ресурс] – Режим доступа: http://dwl.kiev.ua/art/mss/

212. Суслова В.А., Городов А.А. Методы моделирования социальных сетей // Решетневские чтения, №2(19). – 2015. – С. 133-134. – [Электронный ресурс] – Режим доступа: http://cyberleninka.ru/article/n/metody-modelirovaniya-sotsialnyh-setey

213. Тихонов В.И., Миронов М.А. Марковские процессы. – М.: Сов. Радио. – 1977. – 481 с.

214. Тюрин Ю.Н., Макаров А.А. Анализ данных на компьютере (4-е изд., перераб.). – Москва: ИД "ФОРУМ". – 2010. – 367 с.

215. Улічев О.С., Мелешко Є.В. Програмне моделювання поширення інформаційно-психологічних впливів у віртуальних соціальних мережах // Сучасні інформаційні системи. – 2018. – Т. 2, № 2. – С. 35-39.

216. Улічев О.С., Мелешко Є.В. Математична модель розповсюдження інформації в сегменті соціальної мережі // Матеріали XX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 13-14 квітня 2018 р. – Кропивницький: КЛА НАУ. – 2018. – С. 68-72.

217. Улічев О.С., Мелешко Є.В. Моделювання розповсюдження інформаційно-психологічних впливів у сегменті соціальної мережі // Збірник тез VII Міжнародної наукової конференції «Інформація. Комунікація.

Суспільство (ICS-2018)», 17-19 травня 2018 р., Україна, смт. Чинадієво. – Львів: НУ «Львівська політехніка». – 2018. – С. 29-30.

218. Улічев О.С., Мелешко Є.В. Програмна модель розповсюдження інформаційно-психологічних впливів в сегменті соціальної мережі // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м. Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 34-35.

219. Улічев О.С., Мелешко Є.В. Програмна модель соціальної мережі та стратегій поширення інформаційно-психологічних впливів // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький, 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 136-220.

220. Фаулер М., Садаладж П.Дж. NoSQL: новая методология разработки нереляционных баз данных. – М.: Вильямс. – 2013. – 192 с.

221. Форман Д. Много цифр. Анализ больших данных при помощи Excel. – Москва: Альпина Паблишер. – 2016. – 464 с.

222. Хоган Б. Анализ социальных сетей в Интернете // Веб-сайт «ПостНаука» о современной фундаментальной науке и учёных. – 2013. – [Электронный ресурс]. – Режим доступа: https://postnauka.ru/longreads/20259

223. Чжун К. Однородные цепи Маркова. – М.: Мир. – 1954. – 264 с.

224. Шахиди А. Введение в анализ ассоциативных правил // Технологии анализа данных BASEGROUP LABS. – 2020. – [Электронный ресурс] – Режим доступа: https://basegroup.ru/community/articles/intro

225. Швагер Дж. Технический анализ: Полный курс. – М.: Альпина Паблишер. – 2017. – 804 с.

226. Шингалов Д.В., Мелешко Є.В., Босько В.В. Дослідження моделей репутації користувачів соціальної мережі // Збірник тез II Міжнародної науково-практичної конференції "Інформаційна безпека та інформаційні технології", м. Кропивницький, 2-3 квітня 2020 року. – Кропивницький: ЦНТУ. – 2020. – С. 29.

227. Шингалов Д.В., Мелешко Є.В., Босько В.В. Методи нормалізації даних для моделей машинного навчання // Збірник тез XXII Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 197-200.

228. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М. Дослідження програмних засобів для аналізу та візуалізації соціальних графових структур // Збірник тез V Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії (IT&I-2018)», м. Київ, 20-21 листопада 2018 р. – Київ: КНУ. – 2018. – С. 159-160.

229. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М., Резніченко В.А. Математична модель рекомендаційної системи з врахуванням емоційного забарвлення коментарів у якості контексту // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кропивницький: ЦНТУ, 2018. – Вип. 31. – С. 181-186.

230. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М., Резніченко В.А. Методи автоматичного аналізу тональності контенту у соціальних мережах для виявлення інформаційно-психологічних впливів // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – 2017. – Вип. 30. – С. 196-202.

231. Шингалов Д.В., Мелешко Є.В., Улічев А.С. Дослідження Баєсових мереж довіри як засобів для моделювання динамічних процесів у складних мережах // Збірник тез XVII Міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем», м. Дніпро, 20-22 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 284-285.

ДОДАТКИ

Додаток А. Акти впровадження дисертаційних досліджень

ЗАТВЕРДЖУЮ

Виконавчій директор компанії «LineUp»
Тарас КІБІТКІН

„12” 02 2020 р.

А К Т
впровадження результатів наукових досліджень дисертаційної роботи Мелешко Єлизавети Владиславівни

Комісія в складі голови – Тараса Кібіткіна і членів: Ігора Бабіча та Аліни Кібіткіної

склала дійсний акт у тому, що при проектуванні комплексної рекомендаційної системи компанії «LineUp» були використані наступні результати наукових досліджень Мелешко Єлизавети Владиславівни:

1. Метод колаборативної фільтрації, який відрізняється від існуючих використанням продукційних правил для визначення подоби користувачів та використанням показників активності користувачів для формування рекомендацій, що дозволило підвищити стійкість системи у випадку недостатньої кількості вхідних даних та під час холодного старту.
2. Метод імітаційного програмного моделювання користувачів та об’єктів рекомендаційної системи соціальної мережі або веб-ресурсу на основі існуючих і розроблених методів моделювання структури складних мереж та методів моделювання поведінки користувачів, що дозволило генерувати вхідні дані для тестування якості роботи алгоритмів формування рекомендацій.

Голова комісії Тарас КІБІТКІН

Члени комісії: Ігор БАБІЧ
Аліна КІБІТКІНА

МІНІСТЕРСТВО РОЗВИТКУ ЕКОНОМІКИ, ТОРГІВЛІ
ТА СІЛЬСЬКОГО ГОСПОДАРСТВА УКРАЇНИ

ДЕРЖАВНЕ ПІДПРИЄМСТВО
"ПІВДЕННИЙ ДЕРЖАВНИЙ ПРОЕКТНО-
КОНСТРУКТОРСЬКИЙ ТА НАУКОВО-ДОСЛІДНИЙ
ІНСТИТУТ АВІАЦІЙНОЇ ПРОМИСЛОВОСТІ"
(ДП "ПІВДЕНДІПРОНДІАВІАПРОМ")

вул. Сумська, 130а, м. Харків,
61023, УКРАЇНА
Тел: (057) 704-10-47
E-mail: yuzhgap@i.ua
www.yuzhgap.com.ua
код ЄДРПОУ 14307759

ЮЖГИПРОНИИАВИАПРОМ

MINISTRY OF ECONOMIC DEVELOPMENT
AND TRADE OF UKRAINE

STATE ENTERPRISE "SOUTHERN NATIONA
DESIGN & RESEARCH INSTITUTE
OF AEROSPACE INDUSTRIES"
(SE YUZHGIPRONIIAVIAPROM)

130a Sumska St. Kharkiv City
61023 UKRAINE
Phone: + 38 (057) 704-10-47
E-mail:yuzhgap@i.ua
www.yuzhgap.com.ua

_____ № _____
на № 183 від 10.12.2019

ЗАТВЕРДЖУЮ:

Директор Державного підприємства «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості», канд. техн. наук

_____ **Р.В. АРТЮХ**

АКТ
впровадження результатів наукових досліджень дисертаційної роботи Мелешко Єлизавети Владиславівни

Комісія Державного підприємства "Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості" у складі:

Голова комісії: Чмихун Костянтин Євгенійович – головний інженер;

Члени комісії: Чорний Віктор Олексійович – заступник директора з виробництва; Топольницька Олена Олександрівна – головний архітектор проекту, склала дійсний акт у тому, що у робочому процесі ДП «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості» при проведенні дослідно-

проектувальних робіт були використані наступні результати наукових досліджень Мелешко Єлизавети Владиславівни:

1. Метод визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах з використанням математичного апарату марківських та напівмарківських процесів, що дає можливість визначення ймовірностей перебування конкретної рекомендаційної системи в своїх можливих станах в довільний момент часу.

2. Математична модель стійкої рекомендаційної системи на основі визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило здійснити оптимізацію загальних витрат на обслуговування системи в умовах внутрішніх дестабілізуючих факторів.

Голова комісії: Чмихун К.Є.

Члени комісії: Чорний В.О.

Топольницька О.О.

Міністерство економічного
розвитку і торгівлі України

**ДЕРЖАВНЕ ПІДПРИЄМСТВО
"ХАРКІВСЬКИЙ НАУКОВО-
ДОСЛІДНИЙ ІНСТИТУТ ТЕХНОЛОГІЇ
МАШИНОБУДУВАННЯ"
(ДП "ХНДІТМ")**

Україна, 61016, м. Харків,
вул. Кривоконівська, 30
тел./факс: +38 (057) 372-40-50
e-mail: tehmash@ukr.net
www.tehmash.kharkov.ua

Ministry of Economic
Development and Trade of Ukraine

**STATE ENTERPRISE
"KHARKOV SCIENTIFIC-RESEARCH
INSTITUTE OF MECHANICAL
ENGINEERING TECHNOLOGY"**

30, Krivokonevskaya Str.,
Kharkiv, 61016, Ukraine
phone/fax: +38 (057) 372-40-50
e-mail: tehmash@ukr.net
www.tehmash.kharkov.ua

Вих. № 89 від "15" червня 2019 року

ЗАТВЕРДЖУЮ:
Директор Державного підприємства
«Харківський науково-дослідний інститут
технології машинобудування»,
д.т.н., доцент
_____ ***В.В. Косенко***

АКТ
впровадження результатів дисертаційної роботи
Мелешко Єлизавети Владиславівни

Комісія Державного підприємства "Харківський науково-дослідний інститут технології машинобудування" у складі:

голова – Добротворський Сергій Семенович, вчений секретар, д.т.н., професор; члени комісії: Кобзєв Олександр Сергійович, начальник науково-технічного відділу, к.т.н., старший науковий співробітник; Свиридов Юрій Митрофанович, начальник відділу, склала дійсний акт у тому, що у ДП "Харківський науково-дослідний інститут технології машинобудування" при вдосконаленні системи захисту інформації. були використані наступні результати наукових досліджень Мелешко Єлизавети Владиславівни:

1. Математична модель підсистеми інформаційної безпеки стійкої рекомендаційної системи на основі визначення динаміки ймовірностей перебування системи в своїх можливих станах, що дозволило визначити оптимальну частоту перевірки на наявність інформаційної атаки та профілів ботів.

Практичне використання отриманих результатів дозволяє підвищити рівень безпеки інформації в комп'ютерних системах критичного застосування.

Акт впровадження результатів наукових досліджень дисертаційної роботи не є приводом для матеріальної винагороди

Голова комісії : _____ С.С. Добротворський
Члени комісії: _____ О.С. Кобзєв
_____ Ю.М. Свиридов

ЗАТВЕРДЖУЮ
Директор ХНДІСЕ,
д-р. юрид. наук., проф.,
засл. юрист України

О.М. Клюєв
«__» _____ 2020 р.

А К Т
впровадження результатів наукових досліджень дисертаційної роботи у Харківському науково-дослідному інституті судових експертиз ім. Засл. проф. М.С. Бокаріуса
Мелешко Єлизавети Владиславівни

Комісія в складі голови – завідувач сектору комп'ютерно-технічних та телекомунікаційних досліджень, к.т.н. Можаєв Михайло Олександрович і членів: провідний судовий експерт Мороховський Юрій Дмитрович, науковий співробітник Гомон Володимир Олексійович
склала дійсний акт у тому, що у секторі комп'ютерно-технічних та телекомунікаційних досліджень лабораторії інженерно-технічних, екологічних, військових досліджень та досліджень відео-, звукозапису при вдосконаленні системи захисту інформації були використані наступні результати наукових досліджень Мелешко Єлизавети Владиславівни:

Метод виявлення бот-мереж у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів, що дозволило виявляти бот-мережі та розрізняти їх за множинами об'єктів атаки.

Практичне використання отриманих результатів дозволяє підвищити рівень безпеки інформації в комп'ютерних системах критичного застосування.

Акт впровадження результатів наукових досліджень дисертаційної роботи не є приводом для матеріальної винагороди

Голова комісії: Михайло МОЖАЄВ

Члени комісії: Юрій МОРОХОВСЬКИЙ

Володимир ГОМОН

ЗАТВЕРДЖУЮ:
Проректор з наукової роботи
Центральноукраїнського національного
технічного університету

_____О.М. Левченко

«__» _____28. 05. 2020 р.

АКТ

про впровадження результатів дисертаційної роботи
Мелешко Єлизавети Владиславівни
"МЕТОДОЛОГІЯ ЗАБЕЗПЕЧЕННЯ СТІЙКОСТІ
РЕКОМЕНДАЦІЙНИХ СИСТЕМ ДО ДЕСТАБІЛІЗУЮЧИХ ФАКТОРІВ
У КОМП'ЮТЕРНИХ МЕРЕЖАХ"
на здобуття наукового ступеня доктора технічних наук

Комісія у складі: голови – завідуючого кафедрою кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету доктора технічних наук, професора Смірнова О.А. та членів комісії – доцента кібербезпеки та програмного забезпечення, кандидата технічних наук Дрєєва О.М., доцента кафедри кібербезпеки та програмного забезпечення, кандидата технічних наук, доцента Боська В.В., склала цей акт про те, що при розробці лекційних, практичних та лабораторних занять з навчальних дисциплін «Основи захисту інформації», «Захист інформації в комп'ютерних системах», «Проектування комп'ютерних систем та мереж» та «Програмування комп'ютерних мереж» у навчальному процесі Центральноукраїнського національного технічного університету були використані наступні результати наукових досліджень Мелешко Єлизавети Владиславівни:

1. Метод імітаційного програмного моделювання користувачів та об'єктів рекомендаційної системи соціальної мережі або веб-ресурсу на основі існуючих і розроблених методів моделювання структури складних мереж та методів моделювання поведінки користувачів, що дозволило генерувати вхідні дані для тестування якості роботи алгоритмів формування рекомендацій.

2. Метод колаборативної фільтрації, який відрізняється від існуючих використанням продукційних правил для визначення подоби користувачів та використанням показників активності користувачів для формування рекомендацій, що дозволило підвищити стійкість системи у випадку недостатньої кількості вхідних даних та під час холодного старту.

3. Метод виявлення інформаційної атаки на рекомендаційну систему на основі аналізу трендів рейтингів об'єктів, що дозволило знизити кількість витрат на моніторинг безпеки системи за рахунок зняття необхідності пошуку ботів при відсутності ознак атаки.

4. Метод виявлення бот-мереж у рекомендаційній системі на основі графової кластеризації та аналізу дій користувачів, що дозволило виявляти бот-мережі та розрізняти їх за множинами об'єктів атаки.

Застосування результатів дисертаційних досліджень Мелешко Єлизавети Владиславівни дозволило підвищити рівень засвоєння навчального матеріалу з дисциплін «Основи захисту інформації», «Захист інформації в комп'ютерних системах», «Проектування комп'ютерних систем та мереж» та «Програмування комп'ютерних мереж» за рахунок більш поглибленого вивчення сучасних та перспективних методів синтезу рекомендаційних систем та підсистем виявлення інформаційних атак на них у комп'ютерних мережах.

Голова комісії
завідувач кафедри кібербезпеки
та програмного забезпечення
Центральноукраїнського національного
технічного університету
доктор технічних наук, професор О.А. Смірнов

Члени комісії:
доцент кафедри кібербезпеки
та програмного забезпечення
кандидат технічних наук О.М. Дрєєв

доцент кафедри кібербезпеки
та програмного забезпечення
кандидат технічних наук, доцент В.В. Босько

ЗАТВЕРДЖУЮ:
Проректор з науково-педагогічної
роботи Національного технічного
університету «ХПІ»
_____Геннадій ХРИПУНОВ
«02» 09 2020 р.

ДОВІДКА

про впровадження результатів дисертаційної роботи
доцента кафедри «Обчислювальна техніка та програмування»
Мелешко Єлизавети Владиславівни

Матеріали дисертаційної роботи Мелешко Єлизавети Владиславівни, в якій розроблені моделі і методи забезпечення стійкості рекомендаційних систем до дестабілізуючих факторів у комп'ютерних мережах, використовуються дисертантом у навчальному процесі Національного технічного університету «Харківський політехнічний інститут» на кафедрі «Обчислювальна техніка та програмування» при підготовці спеціалістів спеціальності 123 «Комп'ютерна інженерія»: при викладанні навчальних дисциплін «Захист інформації» загальним обсягом 194 академічних години та «Комп'ютерні системи та їх тестування» загальним обсягом 164 академічних години використано розроблені в дисертаційній роботі:

– метод визначення динаміки ймовірностей перебування рекомендаційної системи в своїх можливих станах;
– математична модель стійкої рекомендаційної системи;
– метод колаборативної фільтрації;
– математична модель підсистеми інформаційної безпеки стійкої рекомендаційної системи ;
– метод імітаційного програмного моделювання користувачів та об'єктів рекомендаційної системи соціальної мережі або веб-ресурсу;
– метод виявлення інформаційної атаки на рекомендаційну систему.

Запропонований в дисертаційній роботі Мелешко Є.В. метод виявлення бот-мереж у рекомендаційній системі використовується при виконанні

наукових й курсових робіт студентів та у дипломних роботах спеціальності 123 «Комп'ютерна інженерія».

Результати дисертаційної роботи доцента кафедри «Обчислювальна техніка та програмування» НТУ «ХПІ» Мелешко Є.В. відображені у ряді наукових статей та методичних вказівок, матеріали яких використовуються при роботі зі студентами.

Завідувач кафедри «ОТП»,
д.т.н., професор — Сергій СЕМЕНОВ

Декан факультету «КІТ»,
к.е.н., доцент — Максим ГЛАВЧЕВ

Проф. каф. «ОТП» НТУ «ХПІ»,
д.т.н., професор — Георгій КУЧУК

Додаток В. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації

Список публікацій здобувача за темою дисертації

1. Meleshko Ye., Drieiev O., Yakymenko M., Lysytsia D. Developing a model of the dynamics of states of a recommendation system under conditions of profile injection attacks // Eastern-European Journal of Enterprise Technologies (ISSN 1729-3774). – 2020. – Vol. 4, No 4(106). – P. 14-24. **(SCOPUS)**

2. Meleshko Ye., Raskin L., Semenov S., Sira O. Methodology of probabilistic analysis of state dynamics of multi-dimensional semi-Markov dynamic systems // Eastern-European Journal of Enterprise Technologies (ISSN: 1729-3774). – 2019. – Vol. 6, No 4(102). – P. 6-13. **(SCOPUS)**

3. Ulichev O., Meleshko Ye., Smirnov O., Khokh V., Goncharenko Iu. Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process // CEUR-WS, Vol 2588, Lviv, Ukraine (ISSN: 1613-0073). – 2019. – P. 215-227. **(SCOPUS)**

4. Ulichev O.S., Meleshko Ye.V., Sawicki D., Smailova S. Computer modeling of dissemination of informational influences in social networks with different strategies of information distributors // Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, Wilga, Poland (ISSN: 0277-786X). – 2019. – 111761T. **(SCOPUS)**

5. Ulichev O., Meleshko Y., Khokh V. The computer simulation method of a social network structure for the research of dissemination processes of informational influences // Scientific and Practical Cyber Security Journal (SPCSJ) 4(3). – Georgia, Tbilisi, 2019. – P. 34-47.

6. Meleshko Ye. Computer model of virtual social network with recommendation system // Innovative technologies and scientific solutions for industries. – 2019. – №2(8). – P. 80-85.

7. Meleshko Ye. Method of generating recommendations lists with considering activity indexes of users in a recommendation system // Advanced information systems. – 2019. – T. 3, № 1. – P. 43-47.

8. Мелешко Є.В. Методи кластеризації графів соціальних мереж для побудови рекомендаційних систем // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 2 (54). – С. 129-134.

9. Meleshko Ye. Method of collaborative filtration based on associative networks of users similarity // Advanced information systems. – 2018. – T. 2, № 4. – P. 55-59.

10. Мелешко Є.В. Проблеми сучасних рекомендаційних систем та методи їх рішення // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2018. – Т. 4 (50). – С. 120-124.

11. Мелешко Є.В. Методи оцінки якості роботи рекомендаційних систем // Системи управління, навігації та зв'язку. – Полтава: ПНТУ, 2018. – Вип. 5 (51). – С. 92-97.

12. Мелешко Е.В. Метод встраивания двухуровневых цифровых водяных знаков в медиафайлы для защиты авторских прав // Збірник наукових праць Харківського університету Повітряних сил. – 2013. – Вип. 4. – С. 127-131.

13. Meleshko Ye., Drieiev O., Drieieva H. Method of identification bot profiles based on neural networks in recommendation systems // Advanced Information Systems. – 2020. – Vol. 4, No. 2 – P. 24-28.

14. Міхав В.В., Мелешко Є.В., Якименко М.С. Метод зберігання даних рекомендаційної системи на основі бінарних діаграм рішень // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2020. – Т. 2 (60). – С. 85-89.

15. Мелешко Є.В., Хох В.Д., Босько В.В. Дослідження матричних

факторизаційних моделей рекомендаційних систем // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – Т. 6 (58). – С. 58-62.

16. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження відомих моделей атак на рекомендаційні системи з колаборативною фільтрацією // Збірник наукових праць Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2019. – №. 5 (57). – С. 67-71.

17. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження робастності рекомендаційних систем з колаборативною фільтрацією до інформаційних атак // Електронне фахове наукове видання Кібербезпека: освіта, наука, техніка.– Київ: КУБГ, 2019. – Т.1, № 5. – С. 95-104.

18. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М., Резніченко В.А. Математична модель рекомендаційної системи з врахуванням емоційного забарвлення коментарів у якості контексту // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кропивницький: ЦНТУ, 2018. – Вип. 31. – С. 181-186.

19. Улічев О.С., Мелешко Є.В. Програмне моделювання поширення інформаційно-психологічних впливів у віртуальних соціальних мережах // Сучасні інформаційні системи. – 2018. – Т. 2, № 2. – С. 35-39.

20. Мелешко Є.В., Семенов С.Г., Хох В.Д. Дослідження методів побудови рекомендаційних систем в мережі Інтернет // Збірник наукових праць "Системи управління, навігації та зв’язку". Випуск 1(47). – Полтава: ПНТУ ім. Ю. Кондратюка. – 2018. – С. 131-136.

21. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М., Резніченко В.А. Методи автоматичного аналізу тональності контенту у соціальних мережах для виявлення інформаційно-психологічних впливів // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в

сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – 2017. – Вип. 30. – С. 196-202.

22. Лысенко И.А., Смирнов А.А., Мелешко Е.В. Исследование уровней тестирования программного обеспечения инфотелекоммуникационных систем // Наука і техніка Повітряних Сил Збройних Сил України. – 2014. – № 4. – С. 79-81.

23. Даниленко Д.О., Смірнов О.А., Мелешко Є.В. Дослідження методів виявлення вторгнень в телекомунікаційні системи та мережі // Системи озброєння і військова техніка. – 2012. – № 1. – С. 92-100.

24. Смірнов О.А., Даниленко Д.О., Мелешко Є.В. Метод обнаружения вредоносного программного обеспечения. Часть 1. Корреляционный анализ сетевого трафика // Науково-технічний журнал «Інформаційно-керуючі системи на залізничному транспорті». – 2012. – № 4(95). – С. 8-14.

25. Дрєєв О.М., Смірнов О.А., Мелешко Є.В., Коваленко О.В. Метод прогнозування завантаженості серверу телекомунікаційної мережі // Системи обробки інформації. – 2012. – Вип. 3(2). – С. 181-187.

Відомості про апробацію результатів дисертації

1. Mohammed A.S., Meleshko Y., Balaji S.B., Semenov S. Collaborative filtering method with the use of production rules // Proceedings of ICCIKE, Amity University Dubai; United Arab Emirates. – 2019. – с. 387-391. **(SCOPUS)**

2. Мелешко Е.В., Смирнов А.А. Исследование методов структурного анализа социальных сетей с точки зрения информационной безопасности // Материалы XXI Международной научно-технической конференции "Современные средства связи", 20-21 октября 2016 года, Минск, Республика Беларусь – Минск: Белорусская государственная академия связи. – 2016. – С. 175-177.

3. Шингалов Д.В., Мелешко Є.В., Босько В.В. Методи нормалізації даних для моделей машинного навчання // Збірник тез XXII Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 197-200.

4. Мелешко Є.В., Якименко М.С., Резніченко В.А. Методи оцінки якості роботи алгоритмів машинного навчання // Збірник тез XXII Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 90-94.

5. Мелешко Є.В., Дрєєва Г.М., Дрєєв О.М. Метод кластеризації користувачів соціальної мережі на основі нейронних мереж // Збірник тез XXII Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» імені А.Я. Петренюка, м. Запоріжжя, 15-16 травня 2020 р. – Запоріжжя-Кропивницький: КЛА НАУ. – 2020. – С. 87-90.

6. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження методів підвищення робастності рекомендаційних систем до інформаційних атак // Матеріали VI Міжнародної науково-практичної конференції «Актуальні питання забезпечення кібербезпеки та захисту інформації», 19-22 лютого 2020 р. – м. Київ: Вид-во Європейського університету, 2020. – С. 65-70.

7. Мелешко Є.В., Дрєєва Г., Якименко М., Хох В. Методи моделювання складних мереж // Матеріали IX Міжнародної наукової конференції "Інформація. Комунікація. Суспільство", м. Львів, 21-23 травня 2020 р. – Львів: Видавництво Львівської політехніки. – 2020. – С. 29-30.

8. Міхав В.В., Мелешко Є.В. Порівняння стратегій редагування бінарних діаграм рішень для роботи з графовими даними // Матеріали IX Міжнародної наукової конференції "Інформація. Комунікація. Суспільство", м. Львів, 21-23 травня 2020 р. – Львів: Видавництво Львівської політехніки. –

2020. – С. 17-18.

9. Мелешко Є.В., Дрєєв О.М., Дрєєва Г.М. Розробка методу ідентифікації ботів у рекомендаційних системах // Матеріали X Міжнародної науково-практичної конференції “Комплексне забезпечення якості технологічних процесів та систем”, м. Чернігів, 29-30 квітня 2020 р. – Чернігів: ЧНТУ. – 2020. – С. 165-166.

10. Шингалов Д.В., Мелешко Є.В., Босько В.В. Дослідження моделей репутації користувачів соціальної мережі // Збірник тез II Міжнародної науково-практичної конференції “Інформаційна безпека та інформаційні технології”, м. Кропивницький, 2-3 квітня 2020 року. – Кропивницький: ЦНТУ. – 2020. – С. 29.

11. Міхав В.В., Мелешко Є.В. Метод оптимізації швидкодії бінарних діаграм рішень при представленні даних рекомендаційної системи // Збірник тез II Міжнародної науково-практичної конференції “Інформаційна безпека та інформаційні технології”, м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 17.

12. Мелешко Є.В., Хох В.Д. Дослідження моделей рекомендаційних систем на основі прихованих факторів // Збірник тез II Міжнародної науково-практичної конференції “Інформаційна безпека та інформаційні технології”, м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 46.

13. Мелешко Є.В., Дрєєва Г.М., Гермак В.С., Резніченко В.А., Шевченко О.О. Методи визначення ботів серед користувачів соціальних мереж // Збірник тез II Міжнародної науково-практичної конференції “Інформаційна безпека та інформаційні технології”, м. Кропивницький, 2-3 квітня 2020 р. – Кропивницький: ЦНТУ. – 2020. – С. 44.

14. Мелешко Є., Хох В., Резніченко В., Босько В. Дослідження методів оцінювання робастності рекомендаційних систем до атак накручування

рейтингів // Збірник тез IV Всеукраїнської науково-практичної конференції «Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем MEICS-2019», м. Дніпро, 27-29 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 10-11.

15. Шингалов Д.В., Мелешко Є.В., Улічев А.С. Дослідження Баєсових мереж довіри як засобів для моделювання динамічних процесів у складних мережах // Збірник тез XVII Міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем», м. Дніпро, 20-22 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 284-285.

16. Мелешко Є.В., Хох В.Д., Минайленко Р.М. Розробка експертної системи для виявлення атак на рекомендаційні мережі // Збірник тез XVII Міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем», м. Дніпро, 20-22 листопада 2019 р. – Дніпро: ДНУ. – 2019. – С. 176-177.

17. Мелешко Є.В., Охотний С.М., Резніченко В.А. Дослідження методів визначення спільнот у соціальному графі // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп’ютерних систем та мереж, комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 92-93.

18. Мелешко Є.В., Хох В.Д., Улічев О.С. Методи тестування робастності рекомендаційних систем з колаборативною фільтрацією // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп’ютерних систем та мереж, комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 88-89.

19. Мелешко Є.В., Чабан О.О., Міхав В.В. Способи побудови рекомендаційних систем для соціальних мереж з врахуванням репутації користувачів // Збірник тез Всеукраїнської науково-практичної конференції «Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті», м. Кропивницький, 13-14 листопада 2019 р. – Кропивницький: ЦНТУ. – 2019. – С. 86-87.

20. Мелешко Є.В., Дрєєва Г.М. Дезінформаційні атаки на рекомендаційні системи // Збірник тез Міжнародної наукової конференції «Безпека в сучасному світі», м. Дніпро, 27-28 вересень 2019 р. – Дніпро: ДНУ ім. Олеся Гончара. – 2019. – С. 51-53.

21. Мелешко Є.В. Методи кластеризації графів для побудови рекомендаційних систем соціальних медіа // Збірник тез VII Міжнародної науково-практичної конференції «Обробка сигналів і негаусівських процесів», присвячена пам'яті професора Кунченка Ю.П., м. Черкаси, 23-24 травня 2019 р., – Черкаси: ЧДТУ. – 2019. – С. 100-102.

22. Мелешко Є.В., Чабан О.О. Дослідження засобів парсингу веб-сайтів // Матеріали XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня 2019 року. – Кропивницький: КЛА НАУ. – 2019. – С. 163-167.

23. Мелешко Є.В. Дослідження засобів кластеризації графів у графовій СУБД Neo4j для виявлення співтовариств у соціальних мережах // Збірник тез VIII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство», смт. Чинадієво, 16-18 травня 2019 р. – Львів: Видавництво Львівської політехніки. – 2019. – С. 19-20.

24. Мелешко Є.В., Босько В.В., Резніченко В.А. Дослідження методів комп'ютерної лінгвістики для аналізу контенту веб-сайтів // Збірник тез XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та

їх застосування», м. Кропивницький, 17-18 травня 2019 р. – Кропивницький: КЛА НАУ. – 2019. – С. 96-100.

25. Мелешко Є.В. Метод побудови рекомендаційних систем на основі асоціативних мереж користувачів // Збірник тез XXI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 17-18 травня 2019 р. – Кропивницький: КЛА НАУ. – 2019. – С. 91-95.

26. Мелешко Є.В., Охотний С.М., Босько В.В. Розробка програмного забезпечення для збору та аналізу даних із соціальних мереж // Збірник тез IX Міжнародної науково-практичної конференції «Комплексне забезпечення якості технологічних процесів та систем», Т.2, м. Чернігів, 14-16 травня 2019 р. – Чернігів: ЧНТУ. – 2019. – С. 225-226.

27. Мелешко Є.В. Метод визначення подоби між користувачами у рекомендаційних системах з колаборативною фільтрацією // Збірник тез Науково-практичної конференції «Інформатика, математика, автоматика (ІМА-2019)», м. Суми, 23-26 квітня 2019 р. – Суми: СДУ. – 2019. – С. 213-214.

28. Мелешко Є.В. Розробка програмного забезпечення для виділення співтовариств у соціальній мережі // Збірник тез III Всеукраїнської науково-практичної конференції «Перспективні напрямки сучасної електроніки, інформатики і комп'ютерних систем», м. Дніпро, 21-23 листопада 2018 р. – Дніпро: ДНУ. – 2018. – С. 42-43.

29. Шингалов Д.В., Мелешко Є.В., Минайленко Р.М. Дослідження програмних засобів для аналізу та візуалізації соціальних графових структур // Збірник тез V Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії (IT&I-2018)», м. Київ, 20-21 листопада 2018 р. – Київ: КНУ. – 2018. – С. 159-160.

30. Мелешко Є.В., Босько В.В., Резніченко В.А. Розробка рекомендаційної системи на базі СУБД Neo4j // Збірник тез V Міжнародної науково-практичної конференції «Інформаційні технології та взаємодії (IT&I – 2018)», м. Київ, 20-21 листопада 2018 р. – Київ: КНУ. – 2018. – С. 351-352.

31. Мелешко Є.В., Босько В.В., Резніченко В.А. Застосування асоціативних мереж для побудови рекомендаційних систем // Збірник тез Міжнародної науково-практичної Інтернет-конференції «Автоматика, комп'ютерно-інтегровані технології та проблеми енергоефективності в промисловості і сільському господарстві (АКІТ-2018)», м. Кропивницький, 15-16 листопада 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 165-166.

32. Мелешко Є.В. Дослідження проблем сучасних рекомендаційних систем та методів їх рішення // Збірник тез Міжнародної науково-практичної конференції «Контроль і управління в складних системах (КУСС-2018)», м. Вінниця, 15-17 жовтня 2018 р. – Вінниця: ВНТУ. – 2018. – С. 126.

33. Мелешко Є.В. Методи оцінки точності прогнозування вподобань користувачів веб-ресурсів рекомендаційними системами // Збірник тез X Всеукраїнської науково-практичної конференції «Стан та удосконалення безпеки інформаційно-телекомунікаційних систем (SITS'2018)», с. Коблево, 21-23 червня 2018 р. – Миколаїв: НАУ та МІПРО. – 2018. – С. 58-61.

34. Мелешко Є.В., Дреєва Г.М. Дослідження проблем сучасних рекомендаційних систем // Збірник тез VII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство (ICS-2018)», 17-19 травня 2018 р., Україна, смт. Чинадієво. – Львів: НУ «Львівська політехніка». – 2018. – С. 31-32.

35. Мелешко Є.В. Загрози інформаційній безпеці у рекомендаційних системах соціальних медіа // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м. Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 24-25.

36. Мелешко Є.В. Методи оцінки якості роботи рекомендаційних систем // Матеріали XX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 13-14 квітня 2018 р. – Кропивницький: КЛА НАУ. – 2018. – С. 68-72.

37. Мелешко Є.В. Дослідження методів побудови рекомендаційних систем заснованих на фільтрації контенту // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький, 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 234-237.

38. Улічев О.С., Мелешко Є.В. Програмна модель соціальної мережі та стратегій поширення інформаційно-психологічних впливів // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький, 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 136-220.

39. Охотний С.М., Мелешко Є.В. Визначення центральностей у соціальному графі засобами графової бази даних Neo4j // Збірник тез III Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 247-248.

40. Улічев О.С., Мелешко Є.В. Моделювання розповсюдження інформаційно-психологічних впливів у сегменті соціальної мережі // Збірник тез VII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство (ICS-2018)», 17-19 травня 2018 р., Україна, смт. Чинадієво. – Львів: НУ «Львівська політехніка». – 2018. – С. 29-30.

41. Мелешко Є.В., Хох В.Д., Сидоренко В.В. Розробка автоматизованої системи виявлення, оцінки та розробки заходів по усуненню загроз в інформаційних системах // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м.

Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 38-39.

42. Улічев О.С., Мелешко Є.В. Програмна модель розповсюдження інформаційно-психологічних впливів в сегменті соціальної мережі // Збірник тез VIII Всеукраїнської науково-практичної конференції «Безпека інформаційних технологій (ITSec 2018)», м. Київ, 16-18 травня 2018 р. – Київ: НАУ. – 2018. – С. 34-35.

43. Улічев О.С., Мелешко Є.В. Математична модель розповсюдження інформації в сегменті соціальної мережі // Матеріали XX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 13-14 квітня 2018 р. – Кропивницький: КЛА НАУ. – 2018. – С. 68-72.

44. Мелешко Є.В., Гермак В.С. Дослідження впливу структури соціальної мережі на захищеність від поширення вірусної інформації // Збірник тез доповідей III Міжнародної науково-практичної конференції "Актуальні питання забезпечення кібербезпеки та захисту інформації". с. Верхнє Студене, 22-25 лютого 2017 р. – Київ: Видавництво Європейського університету. – 2017. – С. 118-119.

45. Охотний С.М., Мелешко Є.В., Константинова А.А. Розробка бота для соціальної мережі Facebook на основі фреймворка Selenium // Збірник тез Всеукраїнської науково-практичної Інтернет-конференції "Автоматика та комп’ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті", м. Кропивницький, 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 202-203.

46. Мелешко Є.В., Якименко М.С. Методи виявлення інформаційно-емоційних впливів у текстовій інформації // Збірник тез VI Міжнародної наукової конференції «Інформація. Комунікація. Суспільство (ICS-2017)», м. Львів, 18-20 травня 2017 р. – Львів: Національний університет "Львівська політехніка". – 2017. – С. 30-31.

47. Охотний С.М., Мелешко Є.В. Збирання даних про користувачів віртуальної соціальної мережі за допомогою web-кроулера // Збірник тез II Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології», м. Кропивницький, 20-22 квітня 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 157-159.

48. Мелешко Є.В., Шингалов Д.В., Минайленко Р.М. Методи автоматизації побудови графових структур спільнот у соціальних мережах // Матеріали XIX Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування» присвяченого пам'яті д.ф.-м.н., професора Петренюка Анатолія Яковича, м. Кропивницький, 7-8 квітня 2017 року. – Кропивницький: КЛА НАУ. – 2017. – С. 162-164.

49. Мелешко Є.В. Дослідження методів динамічного аналізу віртуальних соціальних мереж з точки зору інформаційної безпеки // Матеріали Всеукраїнської науково-практичної конференції "Кібербезпека в Україні: правові та організаційні питання", м. Одеса, 21 жовтня 2016 р. – Одеса: ОДУВС. – 2016. – С. 154-155.

50. Мелешко Є.В. Аналіз структури соціальної мережі з точки зору інформаційної безпеки // Збірник тез XVIII міжнародного науково-практичного семінару "Комбінаторні конфігурації та їх застосування", м. Кіровоград, 15-16 квітня 2016. – Кіровоград: Кіровоградський національний технічний університет. – 2016. – С. 93-97.

51. Мелешко Є.В. Методи протидії деструктивним інформаційним впливам в соціальних мережах в умовах інформаційної війни // Збірник тез Всеукраїнської науково-практичної конференції «Інформаційна безпека держави суспільства та особистості», м. Кіровоград, 16 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 139-142.