МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ

УДК 004.4

Тези доповідей

VI Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні технології"

20-21 квітня 2023 року

Кропивницький 2023

УДК 004.4

Матеріали VI Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології": тези доповідей, 20-21 квітня 2023 р. – Кропивницький: ЦНТУ, 2023. – 96 с.

Наведені тези пленарних та секційних доповідей за теоретичними та практичними результатами наукових досліджень і розробок. Представлені результати теоретичних досліджень в галузях проектування інформаційних систем, технологій захисту інформації, використання сучасних інформаційних технологій в управлінні системами за різними галузями народного господарства.

Матеріали публікуються в авторській редакції.

За достовірність викладених фактів, цитат та інших відомостей відповідальність несуть автори.

СЕКЦІЯ 1. ІНФОРМАЦІЙНА БЕЗПЕКА ДЕРЖАВИ, СУСПІЛЬСТВА ТА ОСОБИСТОСТІ

УДК 004.77

Н.С.Петляк[1], Ю.П.Кльоц[1]
npetlyak@khmnu.edu.ua,klots@khmnu.edu.ua
[1]*Хмельницький національний університет, м. Хмельницький*

ПІДХІД ДО АНАЛІЗУ ВИХІДНОГО ТРАФІКУ НА ОСНОВІ СИГНАТУР

З розвитком цифрових технологій зростає кількість пристроїв та мереж, що збільшує поле дії для зловмисників та робить користувачів більш вразливими до кібератак. Зловмисники використовують різні методи для атак на цифрові системи, такі як фішинг, віруси, шкідливе програмне забезпечення, атаки на сервери та інші, що можуть призвести до серйозних наслідків, зокрема таких як крадіжка конфіденційної інформації, розголошення особистих даних, порушення фінансової безпеки, втрата важливих даних тощо.

Аналіз досліджень у даній сфері показав, що IPS/IDS системи переважно розробляються та використовуються для захисту корпоративних мереж від кібератак та інших загроз, що можуть призвести до витоку даних або порушення роботи мережі. Налаштування цих систем може бути складним і вимагати наявності профільних фахівців, які знають особливості конкретної мережі та можуть врахувати більшість відомих загроз. Додаткові затрати на обладнання можуть бути необхідними для використання IPS/IDS систем, оскільки вони потребують спеціалізованого устаткування. Поточні дослідження в галузі IPS/IDS систем зазвичай спрямовані на вирішення конкретних проблем та вузьких задач. Всі вищезгадані системи орієнтовані на захист мережі від зовнішнього впливу та аналізують лише вхідний трафік.

Під час реалізації атаки з мережі на сторонній ресурс збільшується навантаження на мережеве обладнання та збільшуються часові затримки при роботі з мережею Інтернет для типових користувачів. Наявність атак, що надходять з поточної мережі, спричиняють компрометацію мережі та її власника. Функціонування системи виявлення вихідного зловмисного трафіку зменшить загальну кількість мережевих атак та буде запобігати перевантаженню мережевого обладнання.

Аналіз моделей поведінки порушників та оптимізація можливих параметрів дозволяє сформувати сигнатуру пакету для визначення зловмисного трафіка. Запропонований підхід (рис.1) передбачає виконання кількох методів задля реалізації ефективного виявлення порушників.

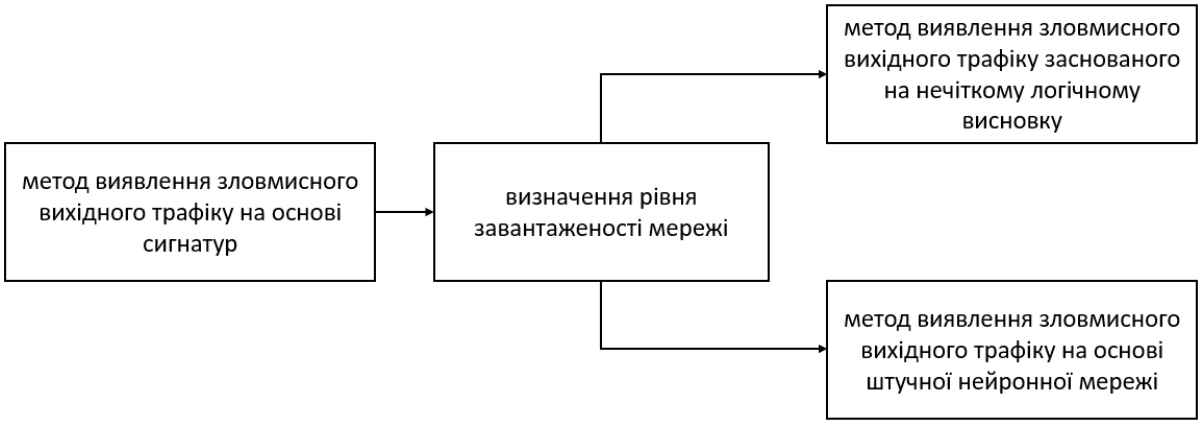


Рис. 1. Підхід до аналізу вихідного трафіку на основі сигнатур

Під час роботи даної системи, що побудована на запропонованому підході, формується сигнатура з пакета трафіку та передається на етап виконання методу виявлення зловмисного вихідного трафіку на основі сигнатур. Даний метод працює наступним чином:

Крок 1. Якщо сформована сигнатура пакету належать множині дозволених сигнатур, то з'єднання дозволяється та відбувається перехід до кроку 4.

Крок 2. Якщо сформована сигнатура пакету належать множині заборонених сигнатур, то з'єднання забороняється та відбувається перехід до кроку 4.

Крок 3. Якщо сформована сигнатура пакету не належать множині дозволених сигнатур та не належать множині заборонених сигнатур, то відбувається перехід до методу виявлення зловмисного вихідного трафіку заснованого на нечіткому логічному висновку.

Крок 4. Завершення обробки пакету.

Наступним етапом роботи системи є перевірка завантаженості мережі. Якщо завантаження не критичне, то виконується метод виявлення зловмисного вихідного трафіку заснований на нечіткому логічному висновку,

який дає достатню ефективність виявлення зловмисного трафіку та разом з цим спричиняє додаткове навантаження на мережу. Якщо ж завантаження мережі є критичним, то виконується метод виявлення зловмисного вихідного трафіку на основі штучної нейронної мережі, оскільки у порівнянні з попереднім працює швидше, хоча має меншу ефективність.

Метод виявлення зловмисного вихідного трафіку заснований на нечіткому логічному висновку містить набори правил, що дозволяють чи забороняють виконувати з'єднання. Даний метод реалізовано наступним чином:

Крок 1. Якщо сформована сигнатура пакета належить до одного з сформованих правил, що задовольняють вимогу дозволеного трафіку, то пакету дозволено з'єднання та відбувається перехід до кроку 2. В іншому випадку перехід до кроку 3.

Крок 2: Сигнатура пакета записується в множину дозволених з'єднань. Перехід до кроку 6.

Крок 3. Якщо сформована сигнатура пакета належить до одного з правил, які задовольняють вимогу щодо забороненого трафіку, то з'єднання та користувач блокуються, відбувається перехід до кроку 4. В іншому випадку перехід до кроку 5.

Крок 4. Сигнатура пакета записується в множину заборонених з'єднань. Перехід до кроку 6.

Крок 5: Сигнатура пакета записується в множину невизначених з'єднань, після чого пакет отримує дозвіл на передачу.

Крок 6: Завершення обробки пакета.

Метод виявлення зловмисного вихідного трафіку на основі штучної нейронної мережі призначений для класифікації вихідного трафіку базуючись на знаннях, що було отримано під час навчання.

Перш ніж використовувати метод потрібно реалізувати класифікатор:

1. Розробити базу даних для навчання, тестування та підтвердження роботоздатності мережі.
2. Виконати попередню обробку: обрати систему ознак та перетворити дані для того, щоб надалі подавати на вхід мережі.
3. Здійснити навчання та тестування мережі: обрати топологію мережі, функцію активації нейронів, алгоритм навчання, оцінити якість роботи мережі.
4. Провести діагностику: дослідити ступінь впливу зовнішніх факторів на прийняте рішення, впевнитись в необхідній точності, за потреби повернутися до пункту 2 задля вдосконалення якості результату.
5. Розгорнути налаштований елемент у системі, що здійснюватиме виявлення та класифікацію зловмисного трафіку в мережі.

У даній роботі було запропоновано:

- метод виявлення зловмисного вихідного трафіку на основі сигнатур, що забезпечує порівняння вихідних сигнатур пакету із сигнатурами, що зберігаються у відповідних словниках;
- метод виявлення зловмисного вихідного трафіку на основі штучної нейронної мережі, що забезпечує аналіз вихідного трафіку в умовах підвищеної завантаженості мережі;
- метод виявлення зловмисного вихідного трафіку заснований на нечіткому логічному висновку, що дає можливість підвищити точність класифікації пакетів вихідного трафіку відповідно до наявних правил для дозволу безпечних з'єднань чи блокування зловмисних.

Запропонований підхід дозволить класифікувати вихідний мережевий трафік з метою дозволу безпечних з'єднань чи блокування зловмисних, що забезпечить зменшення загальної кількості кібератак, запобігання перевантаженню мережевого обладнання та зменшення шансів на компрометацію поточної мережі та її власника.

Список літератури

1. Biswas, R., Roy, S. Botnet traffic identification using neural networks. Multimed Tools Appl 80, 24147–24171 (2021). https://doi.org/10.1007/s11042-021-10765-8
2. Dawadi, B.R.; Adhikari, B.; Srivastava, D.K. Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks. *Sensors* 2023, *23*, 2073. https://doi.org/10.3390/s23042073
3. Song, J.; Wang, X.; He, M.; Jin, L. CSK-CNN: Network Intrusion Detection Model Based on Two-Layer Convolution Neural Network for Handling Imbalanced Dataset. *Information* 2023, *14*, 130. https://doi.org/10.3390/info14020130
4. Mansoor Farooq, "Supervised Learning Techniques for Intrusion Detection System based on Multi-layer Classification Approach" International Journal of Advanced Computer Science and Applications(IJACSA), 13(3), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130338
5. Amrutha Muralidharan Nair and R Santhosh, "Mitigation of DDoS Attack in Cloud Computing Domain by Integrating the DCLB Algorithm with Fuzzy Logic" International Journal of Advanced Computer Science and Applications(IJACSA), 13(10), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0131059

УДК 004.056

Л.В. Константинова, А.О. Норов
liliyashel1976@gmail.com , alonzomer13@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

КРИПТОГРАФІЯ ТА СТЕГАНОГРАФІЯ: РІЗНИЦЯ ТА ЗАСТОСУВАННЯ В ЗАХИСТІ ІНФОРМАЦІЇ

Необхідність збільшення рівня безпеки для захисту інформації на даний момент є актуальним питанням. Криптографія та стеганографія – це два основних методи, які відіграють важливу роль в інформаційній безпеці. Обидва методи використовуються для запобігання несанкціонованому доступу для захисту інформації, але вони відрізняються своїм підходом та застосуванням.

Криптографія – це метод захисту інформації від несанкціонованого доступу шляхом перетворення її в нерозбірливу форму, відому як зашифрований текст, за допомогою математичних алгоритмів. Криптографія включає дві основні операції: шифрування і дешифрування. Шифрування – це процес перетворення відкритого тексту, тобто оригінального повідомлення, в зашифрований текст за допомогою секретного ключа або алгоритму. Дешифрування – це зворотний процес перетворення зашифрованого тексту назад у відкритий за допомогою того ж секретного ключа або алгоритму. Сила криптографії полягає в тому, що розшифрувати зашифрований текст без правильного ключа або алгоритму неможливо використанням обчислювальної техніки [1].

Існує два основних типи криптографії: криптографія з симетричним ключем і криптографія з відкритим ключем. У криптографії з симетричним ключем, один і той самий ключ використовується як для шифрування, так і для розшифрування. Цей тип криптографії простий і ефективний, але вимагає безпечного обміну секретним ключем між сторонами, що спілкуються. На противагу цьому, криптографія з відкритим ключем використовує два різні ключі: відкритий ключ для шифрування і закритий ключ для розшифрування. Відкритий ключ є широко розповсюдженим, тоді як закритий ключ зберігається в таємниці його власником. Цей тип криптографії забезпечує безпечний зв'язок, не вимагаючи безпечного обміну секретним ключем [1][2].

Криптографія використовується для забезпечення різних послуг безпеки, таких як конфіденційність, цілісність, аутентифікація. Сучасна криптографія використовує різні криптографічні алгоритми, такі як Advanced Encryption Standard (AES), Rivest-Shamir–Adleman (RSA) та Elliptic Curve Cryptography (ECC), щоб забезпечити безпечний зв'язок і захистити конфіденційну інформацію [1].

Стеганографія - це мистецтво приховування секретної інформації під прикриттям повідомлення або носія у спосіб, який не є очевидним для спостерігача [3]. Це потужний інструмент захисту конфіденційної інформації, який часто використовується разом з криптографією для забезпечення інформаційної безпеки. Мета стеганографії – приховати секретне повідомлення таким чином, щоб його не міг виявити сторонній спостерігач. Цього можна досягти шляхом вбудовування секретного повідомлення в основне повідомлення або медіа за допомогою різних методів, таких як заміна LSB (Least Significant Bit), різниця значень пікселів і методи розширеного спектра.

Метод заміни LSB передбачає модифікацію найменш значущих бітів носія для вбудовування секретного повідомлення. Метод різниці значень пікселів передбачає вбудовування повідомлення шляхом зміни різниці між сусідніми пікселями. Метод розширеного спектру передбачає поширення секретного повідомлення по всьому частотному спектру носія, що ускладнює виявлення прихованого повідомлення [3].

Стеганографія може використовуватися для різних цілей, таких як приховування інформації, нанесення водяних знаків і відбитків пальців. Приховування інформації полягає у вбудовуванні секретного повідомлення в основне повідомлення або носій. Нанесення водяних знаків передбачає вбудовування цифрового підпису або водяного знаку в носій, щоб захистити його від несанкціонованого використання або розповсюдження. Відбитки пальців включають в себе вбудовування унікального ідентифікатора в носій для ідентифікації джерела носія.

Основна відмінність між криптографією і стеганографією полягає в підході, який використовується для захисту інформації. Криптографія передбачає використання математичних алгоритмів і методів для перетворення інформації в нечитабельну форму, що робить її захищеною від несанкціонованого доступу. Вона використовується для забезпечення конфіденційності, цілісності та автентичності інформації. Мета криптографії - захистити інформацію, ускладнивши її прочитання або розуміння, навіть якщо її перехоплять сторонні особи [2][3].

Стеганографія, з іншого боку, передбачає приховування інформації на видному місці шляхом вбудовування її в інше повідомлення або носій. На відміну від криптографії, стеганографія не шифрує повідомлення або дані, а маскує їх в іншому повідомленні або носії. Мета стеганографії - зберегти існування прихованого повідомлення в таємниці, тому навіть якщо повідомлення-прикриття перехоплять, приховане повідомлення залишиться прихованим [2][3].

Ще одна ключова відмінність між криптографією і стеганографією - це рівень безпеки, який вони

забезпечують. Криптографія забезпечує вищий рівень безпеки, оскільки вона шифрує повідомлення або дані, роблячи їх нечитабельними і захищеними від несанкціонованого доступу [2][3]. Стеганографія, з іншого боку, забезпечує більш низький рівень безпеки, оскільки приховане повідомлення все ще може бути вразливим для виявлення та вилучення цілеспрямованим зловмисником [2][3].

Криптографія і стеганографія часто використовуються разом для забезпечення підвищеної безпеки інформації. Ось кілька поширених застосувань, де криптографія і стеганографія використовуються разом:

– Безпечний зв'язок. Криптографію можна використовувати для шифрування повідомлення, а стеганографію - для приховування зашифрованого повідомлення в іншому носії, наприклад, зображенні або аудіофайлі. Така комбінація забезпечує додатковий рівень безпеки, ускладнюючи виявлення як повідомлення, так і шифрування.

– Цифрові водяні знаки. Криптографія використовується для шифрування унікального ідентифікатора, а стеганографія - для приховування ідентифікатора всередині медіафайлу, наприклад, відео або зображення [2][3]. Ця комбінація дозволяє власнику захистити свою роботу від несанкціонованого копіювання та розповсюдження.

– Аутентифікація. Криптографія застосовується для забезпечення автентифікації, а стеганографія - для приховування коду аутентифікації всередині медіафайлу, наприклад, зображення або аудіофайлу [2][3]. Така комбінація забезпечує додатковий рівень безпеки, ускладнюючи виявлення як коду автентифікації, так і медіафайлу.

– Прихована комунікація. Криптографія призначена для шифрування повідомлення, а стеганографія - для приховування зашифрованого повідомлення в іншому медіафайлі, наприклад, зображенні або аудіофайлі [2][3]. Ця комбінація дозволяє сторонам спілкуватися непомітно, приховуючи як повідомлення, так і шифрування.

– Захист авторських прав. Криптографія дозволяє зашифрувати повідомлення про авторське право, а стеганографія - приховати його в медіафайлі, наприклад, у відео або зображенні [2][3]. Така комбінація забезпечує додатковий рівень безпеки, ускладнюючи виявлення як повідомлення про авторське право, так і медіафайлу.

– Обмін ключами на основі стеганографії. Криптографія забезпечує безпечний обмін ключами між двома сторонами, а стеганографія дозволяє приховати ключ у медіафайлі, наприклад, у зображенні або аудіофайлі [2][3]. Ця комбінація методів може бути використана для безпечного обміну ключами без загрози перехоплення інформації сторонніми особами.

– Цифрова криміналістика. Криптографія і стеганографія можуть бути використані для виявлення і відновлення прихованих даних у цифровому медіафайлі [2][3]. Ця комбінація методів може бути використана для відновлення прихованих повідомлень, водяних знаків або будь-яких інших даних, які були вбудовані в медіафайл з незаконною метою.

Висновки. Отже, криптографія і стеганографія є важливими методами, що використовуються в інформаційній безпеці для захисту конфіденційності, цілісності даних. Криптографія фокусується на перетворенні відкритого тексту в зашифрований, щоб забезпечити доступ до інформації лише уповноваженим особам, тоді як стеганографія фокусується на приховуванні існування інформації. Обидва методи мають різні підходи та застосування в інформаційній безпеці, і розуміння їх відмінностей та сильних сторін є важливим для розробки безпечних систем зв'язку. Ці методи часто використовуються разом для забезпечення більш високого рівня безпеки.

Список літератури

1. William Stallings, “Cryptography and Network Security: Principles and Practice”, Prentice Hall Press, 2010.

2. Douglas R. Stinson, “Cryptography: Theory and Practice”, CRC Press, 2005.

3. Jessica Fridrich, “Steganography in Digital Media Principles, Algorithms, and Application”, Cambridge University Press, 2009.

УДК 004.056

Л.В. Константинова, О.С. Сосна
liliyashel1976@gmail.com, litetab387@gmail.com
Центральноукраїнський національний технічний університет, м .Кропивницький

ОГЛЯД ЗАСОБІВ КОМПЛЕКСНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

Стрімкий розвиток інформаційних технологій привів до проблем захисту інформації або забезпечення безпеки інформації. Комплексні системи захисту інформації (КСЗІ) допомагають у вирішенні цих проблем. Дослідження та огляд методів і засобів, об'єднаних єдиним цільовим призначенням, які забезпечують необхідну ефективність захисту інформації в автоматизованих системах (АС) є актуальними в цей час.

Інформаційна безпека – стан інформації, в якому забезпечується збереження визначених політикою безпеки властивостей інформації [1].

Для характеристики основних властивостей інформації використовується модель CIA (конфіденційність (англ. Confidentiality, privacy), цілісність (англ. Integrity), доступність (англ. Availability), але існують й інші властивості [1]:

Конфіденційність. Загрози, що відносяться до несанкціонованого ознайомлення з інформацією, становлять загрози конфіденційності. Це може бути довірча конфіденційність, адміністративна конфіденційність, повторне використання об'єктів, аналіз прихованих каналів, конфіденційність при обміні (експорті/імпорті).

Цілісність. Загрози, що відносяться до несанкціонованої модифікації інформації, становлять загрози цілісності. Це може бути довірча цілісність, адміністративна цілісність, відкат і цілісність при обміні.

Доступність. Загрози, що відносяться до порушення можливості використання комп'ютерних систем або оброблюваної інформації, становлять загрози доступності. В цей пункт може входити використання ресурсів, стійкість до відмов, гаряча заміна та відновлення після збоїв.

Спостереженість. Ідентифікація і контроль за діями користувачів, керованість комп'ютерною системою становлять предмет послуг спостереженості й керованості. Цей пункт включає реєстрацію, ідентифікацію й автентифікацію, достовірний канал, розподіл обов'язків, цілісність комплексу засобів захисту, самотестування, автентифікацію при обміні, автентифікацію відправника (невідмова від авторства), автентифікацію одержувача (невідмова від одержання).

Захист інформації — сукупність методів і засобів, що забезпечують цілісність, конфіденційність і доступність інформації за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації [2].

КСЗІ — це система технічних і нетехнічних заходів, що дозволяють запобігти або ускладнити можливість доступу до інформації, оброблюваної автоматизованим способом в інформаційно-телекомунікаційних системах (ІТС).

Істотна частина проблем забезпечення захисту такої інформації може бути вирішена відомими правовими та організаційними заходами, однак, враховуючи розвиток інформаційних технологій, наявна тенденція зростання необхідності застосування технічних заходів і засобів її захисту.

Існують два основних заходи щодо захисту інформації:

Організаційні заходи. Організаційні заходи включають в себе створення концепції інформаційної безпеки, а також:

- складання посадових інструкцій для користувачів та обслуговуючого персоналу;
- створення правил адміністрування компонент інформаційної системи, обліку, зберігання, розмноження, знищення носіїв інформації, ідентифікації користувачів;
- розробка планів дій у разі виявлення спроб несанкціонованого доступу до інформаційних ресурсів системи, виходу з ладу засобів захисту, виникнення надзвичайної ситуації;
- навчання правилам інформаційної безпеки користувачів.

У разі необхідності, в рамках проведення організаційних заходів може бути створена служба інформаційної безпеки, проведена реорганізація системи діловодства та зберігання документів.

Інженерно-технічні заходи. Інженерно-технічні заходи – сукупність спеціальних технічних засобів та їх використання для захисту інформації. Вибір інженерно-технічних заходів залежить від рівня захищеності інформації, який необхідно забезпечити.

Інженерно-технічні заходи, що проводяться для захисту інформаційної інфраструктури організації, можуть включати використання захищених підключень, міжмережевих екранів, розмежування потоків інформації між сегментами мережі, використання засобів шифрування і захисту від несанкціонованого доступу.

У разі необхідності, в рамках проведення інженерно-технічних заходів, може здійснюватися установка в приміщеннях систем охоронно-пожежної сигналізації, систем контролю і управління доступом. Окремі приміщення можуть бути обладнані засобами захисту від витоку акустичної (мовної) інформації.

Захист інформації в АС повинен ґрунтуватися на таких основних принципах:

• системності;
• комплексності;
• безперервності захисту;
• розумної достатності;
• гнучкості управління і застосування;
• відкритості алгоритмів і механізмів захисту;
• простоти застосування захисних заходів і засобів.

Існують основні вимоги щодо комплексної системи захисту інформації [2] Система захисту інформації повинна забезпечувати виконання АС своїх основних функцій без істотного погіршення характеристик останньої. Вона повинна бути економічно доцільною, оскільки вартість системи захисту інформації входить у вартість АС. Захист інформації в АС повинен забезпечуватися на всіх етапах життєвого циклу, при всіх технологічних режимах обробки інформації, в тому числі при проведенні ремонтних і регламентних робіт. В систему захисту інформації повинні бути закладені можливості її вдосконалення й розвитку відповідно до умов експлуатації та конфігурації АС. Відповідно до встановлених правил КСЗІ повинна забезпечувати розмежування доступу до інформації з обмеженим доступом з відволіканням порушника на помилкову інформацію, тобто мати властивості активного і пасивного захисту. При взаємодії захищеної АС з незахищеними АС система захисту повинна забезпечувати дотримання встановлених правил розмежування доступу. Система захисту повинна дозволяти проводити облік і розслідування випадків порушення безпеки інформації в АС. Застосування системи захисту не повинно погіршувати екологічну обстановку, не бути складною для користувача, не викликати психологічної протидії та бажання обійтися без неї.

Оскільки система може бути визначена як сукупність взаємопов'язаних елементів, то призначення СЗІ полягає в тому, щоб об'єднати всі складові захисту в єдиний механізм, в якому кожен компонент, виконуючи свою функцію, одночасно забезпечує виконання функцій іншими компонентами та пов'язаний з ними логічно і технологічно [3].

Висновки. Таким чином, правильне формування комплексної системи захисту інформації, дуже важливий компонент в конфіденційності інформації загалом. Залежно від виду та форми подання інформаційних сигналів, які циркулюють в інформаційно-телекомунікаційній системі, у тому числі і в АС, при побудові КСЗІ можуть використовуватися різні засоби захисту.

Список літератури

1. Остапов С. Е. технологія захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.
2. Бурячок В. Л. Інформаційний та кіберпростори: проблеми безпеки, методи та засоби боротьби : посібник / [В. Л. Бурячок, С. В. Толюпа, В. В. Семко та ін.]. – К. : ДУТ-КНУ, 2016. – 178 с.
3. Комплексні системи захисту інформації : навчальний посібник / К63 [Яремчук Ю. Є., Павловський П. В., Катаєв В. С., Сінюгін В. В.] – Вінниця : ВНТУ, 2018. – 118 с.

УДК 004.056, 004.75

Д. Р. Рачек, студент 4-го курсу
racdanil602@gmsil.com
Київський національний університет будівництва і архітектури, м. Київ

РОСІЙСЬКО- УКРАЇНСЬКА ІНФОРМАЦІЙНА ВІЙНА

Російсько-Українська війна- це найбільший військовий конфлікт 21 століття. Проте цей конфлікт має різні фронти, і один з них – інформаційний. Недарма більшість західних інституцій називають цю війну «гібридною». Інформаційна війна йде по всьому світу, у якій країна агресор намагається підірвати підтримку України серед світових лідерів і закінчити почату у 2014 війна з ціллю захоплення України і її ресурсів.

Якщо розглянути інформаційний фронт більш детально то варто звернути увагу на те одразу після розвалу СРСР на території сучасної України гостро стояло питання притиснення російськомовних. Для нас це питання не стояло гостро, адже усім знайома фраза «Ми живемо у вільній країні, якою мовою хочу тією і спілкуюся.», але для на території РФ, де з перших днів після її (країни) створення почала своє утворення пропагандистська машина, яка у майбутньому застосовуватиметься в тандемі з цензурою: цензура відсікатиме будь-яке інакомислення, пропаганда змушуватиме думати відповідно до інтересів і цілей владної верхівки[2, 3], то для серед них вже тоді мовне питання стояло як наріжний камінь і головною риторикою в інформаційному полі було: Про насильницьку українізацію жителів південно- східних регіонів в умовах незалежності України».[1] Така політика проводилася на території усіх держав пострадянського табору, основною ціллю ж було утримати ці країни у позаблоковому статусі, або ж як каже російський президент: «Не позволить расширению НАТО».[4]

Офіційним початком російсько-української інформаційної війни, як вже зазначалося раніше, вважають 2014 рік, після закінчення «Революції гідності». Російські підконтрольні владі ЗМІ, одразу почали зображати український народ, як фашистів, а владу – корумпованою та неефективною.[5] Можливо це б і залишилося не більше чим пусті балачки, але разом з проросійськими мітингами на території Донецьку та Луганську – це відіграло суттєву роль у початку так званої АТО. Упродовж 9 років методи інформаційної війни, що використовувала РФ були наступні[1]:

1. Розповсюдження фейків та недостовірної інформації. Російські ЗМІ та соціальні мережі активно поширюють непідтверджені та недостовірні новини про Україну. Наприклад, Росія стверджувала, що в Україні війна почалася не з її боку, а з боку української влади, що українці вбивають російських громадян та знищують православні храми. Використання пропаганди.
2. Російські ЗМІ та соціальні мережі активно використовують пропагандистські методи для дискредитації України та її влади. Наприклад, вони зображають українських воїнів як нацистів та фашистів, а українську владу - як неефективну та корумповану.
3. Використання кібератак. Російські хакерські групи проводять кібератаки на українські веб-сайти та інфраструктуру. Це може призвести до зупинки роботи певних систем, витоку конфіденційної інформації та інших наслідків.
4. Вплив на громадську думку. Російські ЗМІ та соціальні мережі використовують певні методи, щоб впливати на громадську думку. Наприклад, вони використовують агресивний тон, емоційне впливання, застосовують маніпулятивні техніки

Україна не була готова до такої потужної атаки, що й спричинило купу наслідків для суспільства:

1. Інформаційна війна Росії проти України призвела до значного впливу на думки та уявлення громадян України про події в країні та у світі.
2. Російська пропаганда створила в Україні складне політичне та соціальне середовище, засноване на міфах та фальшивих новинах.
3. Інформаційна війна Росії призвела до поділу суспільства на "правильних" та "неправильних", що створює бар'єри в міжособистісних та міжетнічних відносинах.
4. Російська пропаганда має вплив на вибір політичних лідерів, на погляди щодо євроінтеграції, НАТО та інших міжнародних структур.
5. Інформаційна війна Росії може викликати у громадян України почуття безпорадності, страху та обурення, що може призвести до психологічної напруги та зниження якості життя.
6. Українське суспільство змушене витрачати значні зусилля на боротьбу з інформаційною агресією, що відбиється на ресурсах та увазі людей до інших питань.

Проте упродовж цих років українські політики не стояли осторонь і теж реформували систему інформаційної безпеки. Серед ключових документів, які врегульовують питання безпеки в інформаційному просторі слід відзначити Конституцію України, Закон України «Про ратифікацію Конвенції про кіберзлочинність», Закон України «Про національну безпеку», Закон України «Про захист інформації в інформаційно-телекомунікаційних системах», «Стратегія національної безпеки України», «Стратегія кібербезпеки України», «Доктрина інформаційної безпеки» та ін. Особливе значення мав Указ Президента України № 133/2017 «Про рішення Ради національної безпеки і оборони України від 28 квітня 2017 року «Про

застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій)», згідно з яким українські провайдери блокували російські соціальні мережі та сайти, зокрема «вконтакте», «одноклассники», «mail.ru», «яндекс», «Лабораторія Касперского», «Dr. Web» та офіційного дистриб'ютора «1С» на території України строком на 3 роки (Указ Президента України № 133/2017…). Забезпечення інформаційної безпеки суспільства покладено і в «Стратегію кібербезпеки України» та «Доктрину інформаційної безпеки». Ці зміни мали позитивний вплив, адже вони зменшили можливість російських змі впливати на інформаційне поле, викрадати інформацію, що змусило українців шукати альтернативні джерела інформації, більшість з яких є офіційними.[1]

Загалом боротьба на інформаційному фронті це складний процес, який вимагає комплексного підходу, і зацікавлені українці теж брали у цьому участь. Так, у 2014 р. створено Інформаційний спротив (ІС) – недержавний проєкт, який виконує завдання протидії зовнішній ворожій інформаційній загрозі у різних сферах суспільного і державного життя. Фактично, від початку російської агресії ІС, шляхом перевірки різної інформації сповіщав про хід російсько-української війни, а також викривав факти неправдивої інформації зі сторони проросійських ЗМІ. Іншим проєктом, який боровся зросійською інформаційною агресією став StopFake. Упродовж 2014–2020 рр. організація спростувала низку міфів, фейків, які поширювалися у вітчизняних чи іноземних ЗМІ, а також у соцмережах. Крім того, проєкт реалізує низку досліджень, спрямованих на визначення сприйняття суспільством інформації, а також формують рекомендації щодо розпізнання неправдивої інформації (StopFake), тим самим частково виконуючи функції держави у поширенні серед населення знань у цій сфері. Зважаючи на те, що Україна на початкових етапах програвала інформаційну війну Росії, діяльність цих організацій фактично не допускала крахуі засилля в інформаційному просторі країни. Згодом до недержавних інституцій підключилися і спеціально створені державні.

Протягом останніх років використовуючи наступні методи ведення інформаційної війни держава була готова до критичної точки 24 лютого 2022 року: [6]

1. Розвиток медіа грамотності та критичного мислення - це дуже важливі навички для визначення правдивої та неправдивої інформації. Люди повинні бути навчені перевіряти джерела інформації, розрізняти факти від домислів та маніпуляцій. Такі навички можуть допомогти зменшити вплив дезінформації та фейкових новин.
2. Підтримка свободи преси та інформаційної свободи - це важливий елемент для забезпечення прозорості та об'єктивності інформації. Держави повинні підтримувати незалежні ЗМІ та журналістів, а також забезпечувати доступ до інформації.
3. Розвиток кібербезпеки - це важлива складова інформаційної безпеки, яка допомагає захистити комп'ютерні системи від кібератак та інших форм кіберзлочинності. Для цього необхідно використовувати надійні паролі, оновлювати програмне забезпечення та антивіруси, а також не довіряти підозрілим електронним повідомленням і лінкам.
4. Співпраця міжнародної спільноти - це важливий елемент у боротьбі з інформаційною війною.
5. Ефективне реагування на дезінформацію та фейкові новини - дуже важливо швидко реагувати на поширення неправдивої інформації та маніпуляцій. Для цього потрібно створювати штаби кризового реагування та команди зі спеціалізованими знаннями та навичками.
6. Ефективна комунікація - дуже важливо забезпечити ефективну комунікацію з населенням, щоб передати правдиву та об'єктивну інформацію. Для цього потрібно використовувати різні канали комунікації, зокрема соціальні мережі, телебачення, радіо та інтернет.
7. Посилення законодавства - необхідно забезпечити ефективнішу захист інформації та боротися з маніпуляціями та дезінформацією. Для цього потрібно розробляти та вдосконалювати законодавство щодо інформаційної безпеки та захисту персональних даних, а також покарання винних у поширенні дезінформації та фейкових новин.

Список літератури

1. https://sdc-journal.com/index.php/journal/article/view/285/228
2. ЦЕНЗУРА В РОСІЇ: ЗАКРИЛИ ЗМІ, ЩО ПИШУТЬ ПРАВДУ ПРО УКРАЇНУ. Архів оригіналу за 14 жовтня 2014. Процитовано 7 липня 2014.
3. [[https://web.archive.org/web/20140704041741/http://tsn.ua/video/video-novini/liya-ahedzhakova-vvazhaye-scho-propaganda-i-cenzura-v-rosiyi-posilyuyutsya.html Архівовано 4 липня 2014 у Wayback Machine.] Лія Ахеджакова вважає, що пропаганда і цензура в Росії посилюються, ТСН]
4. https://www.youtube.com/watch?v=Z3m2MKaRxeg
5. https://russian.rt.com/tag/maidan
6. https://enpuir.npu.edu.ua/bitstream/handle/123456789/25591/Zhadko%2064-95.pdf?sequence=1

УДК 004.056.5

В.О. Смутко., здобувач вищої освіти першого курсу магістратури,
vitaliismutko@gmail.com
М.О. Єремєєв, здобувач вищої освіти першого курсу магістратури,
yeremeevmaol@gmail.com
О.В. Коваленко, д.т.н., доцент
clashav@gmail.com
Центральноукраїнський національний технічний університет,м. Кропивницький

ПЕРСПЕКТИВИ РОЗВИТКУ КІБЕРБЕЗПЕКИ В УКРАЇНІ В КОНТЕКСТІ СУЧАСНИХ ВИКЛИКІВ ТА ТЕНДЕНЦІЙ

У сучасному світі, де інформаційні технології займають центральне місце, кібербезпека стає дедалі більш важливою. Кібератаки та кіберзлочини стають все більш розповсюдженими, а їхні наслідки можуть бути катастрофічними для індивідів, компаній та навіть цілих країн. Розглянемо тенденції та виклики в області кібербезпеки та їхні перспективи для України.

Тенденції в області кібербезпеки [1-4]

1. Тенденція зростання кількості кібератак та їхній різноманітний характер. Сучасні кіберзлочинці користуються різними методами та технологіями для зламування систем захисту та отримання конфіденційної інформації. Наприклад, фішингові атаки, кібершпигунство та кібертероризм стають все більш популярними серед злочинців.

2. Тенденція широкого застосування штучного інтелекту та машинного навчання в кібербезпеці. Штучний інтелект може допомогти виявляти та запобігати кібератакам, а машинне навчання дозволяє створювати більш ефективні системи захисту.

3. **Тенденція є зростання популярності хмарних технологій** та зберігання даних в хмарних сервісах. Це дозволяє зберігати важливу інформацію в безпечному місці з надійними заходами захисту та забезпечує можливість забезпечувати доступ до даних з будь-якої точки світу.

Виклики в області кібербезпеки

Незважаючи на тенденції, які забезпечують більшу безпеку, існують певні виклики, з якими доведеться зіткнутися в області кібербезпеки.

1. Забезпечення захисту від нових видів кібератак. Сучасні злочинці використовують найновіші технології, щоб отримати доступ до конфіденційної інформації. Це вимагає постійного оновлення захисних систем та швидкої реакції на нові загрози.

2. Брак кваліфікованих фахівців в області кібербезпеки. Недостатнє знання та навички в цій галузі можуть призвести до серйозних наслідків, що підкреслює важливість розвитку освітніх програм у закладах вищої освіти та програм професійного розвитку.

3. Захист критичних інфраструктур від кібератак. Ці інфраструктури включають електропостачання, водопостачання, транспорт та інші сфери, які можуть бути на межі катастрофи в разі кібератаки.

Перспективи для України

Україна має потужний потенціал у сфері кібербезпеки та може стати лідером у розвитку цієї галузі з урахуванням проведеної великої кількості атак та протидії ним в останній час. Для досягнення цієї мети потрібно вирішувати виклики та використовувати можливості, які пропонують тенденції.

Особлива увага має бути приділена розвитку освітніх програм у закладах вищої освіти, які дозволяють готувати кваліфікованих фахівців у сфері кібербезпеки. Крім того, важливо залучати до цієї галузі молодих талановитих людей та підтримувати їхній розвиток.

Для забезпечення ефективного захисту від кібератак, важливо також розвивати міжнародне співробітництво та обмін досвідом з іншими країнами. Україна може взяти участь у міжнародних проєктах та ініціативах з метою підвищення своєї кібербезпеки.

Розглянемо приклади кібератак, що траплялися в Україні, а також деякі ініціативи, які вже запроваджуються у країні для забезпечення кібербезпеки.

Наприклад, в Україні відбулася серія кібератак на критичну інфраструктуру в 2015 році, коли хакери впливали на роботу електростанцій, підприємств енергетичного сектору, та інші об'єкти. У результаті було забезпечено перерву в постачанні електроенергії на певних територіях країни. Ці атаки викликали серйозну обуреність українського уряду та змусили його звернути більше уваги на кібербезпеку країни.

Уряд України відповів на ці кібератаки, затвердивши Національну стратегію кібербезпеки [5], яка визначає основні напрями дій для забезпечення кібербезпеки країни. У рамках цієї стратегії були розроблені і запроваджені різноманітні ініціативи, такі як створення Національного центру кібербезпеки, зміцнення кіберзахисту критичної інфраструктури, включаючи енергетичний сектор, транспортні мережі та фінансові установи, а також підвищення кваліфікації фахівців у галузі кібербезпеки.

Окрім того, в Україні проводяться ініціативи, спрямовані на залучення молоді до галузі кібербезпеки. Наприклад, з 2016 року у країні діє програма "Школа кібербезпеки", яка ставить за мету навчити старшокласників безпечному користуванню Інтернетом та збільшити кількість молодих фахівців у галузі кібербезпеки.

Також, Україна активно співпрацює з міжнародними організаціями з питань кібербезпеки, такими як Європейський Союз та НАТО. Наприклад, Україна брала участь у міжнародних навчаннях з кібербезпеки, які проводяться НАТО в рамках Ініціативи з партнерства.

Однак, незважаючи на всі ці ініціативи, кібербезпека залишається високою пріоритетною задачею в Україні, і кількість кібератак на українські компанії та державні установи продовжує зростати. За даними досліджень, в 2022 році кількість кібератак на українські компанії зросла порівняно з 2021 роком. Також було зареєстровано зростання кількості фішинг-атак на українських користувачів. Наразі Україна розробила Національну стратегію кібербезпеки на період до 2025 року, яка передбачає заходи з підвищення кібербезпеки в різних секторах економіки та суспільства.

Як висновок у зв'язку зі зростаючим значенням інформаційних технологій у сучасному світі, кібербезпека стає однією з найважливіших галузей розвитку. Незважаючи на те, що зроблено певний прогрес у цій галузі, виклики, пов'язані з новими видами кібератак, браком кваліфікованих фахівців та захистом критичних інфраструктур, потребують негайного вирішення. Україна може взяти участь у розвитку цієї галузі та стати лідером у світі, якщо будуть використані всі можливості, що пропонуються тенденціями.

Список літератури

1. Katherine Marconi, Harold Lehmann Big Data and Health Analytics. Auerbach Publications, 2014. 328 p.

2. Ashish Khanna, Deepak Gupta, Nilanjan Dey Applications of Big Data in Healthcare. Academic Press. 2021. 281 p.

3. Pantea Keikhosrokiani Big Data Analytics for Healthcare. Academic Press. 2022. 354 p.

4. Pardeep Kumar, Yugal Kumar, Mohamed A. Tawhid Machine Learning, Big Data, and IoT for Medical Informatics. Academic Press. 2021. 458 p.

Список літератури

1. Які російські та проросійські хакери атакують Україну. URL: https://cip.gov.ua/ua/news/yaki-rosiiski-ta-prorosiiski-khakeri-atakuyut-ukrayinu (дата звернення: 13.04.2023).

2. CERT-UA від початку року опрацювала більше двох тисяч кібератак на Україну. URL: https://cip.gov.ua/ua/news/cert-ua-vid-pochatku-roku-opracyuvala-bilshe-dvokh-tisyach-kiberatak-na-ukrayinu (дата звернення: 13.04.2023).

3. Кількість кібератак на Україну продовжує зростати. URL: https://cip.gov.ua/ua/news/kilkist-kiberatak-na-ukrayinu-prodovzhuye-zrostati (дата звернення: 13.04.2023).

4. З початку війни зафіксовано понад 1,2 млн кібератак на енергосектор. URL: https://www.kmu.gov.ua/news/z-pochatku-vijni-zafiksovano-ponad-12-mln-kiberatak-na-energosektor-farid-safarov (дата звернення: 13.04.2023).

5. Указ президента України №447 / 2021. URL: https://www.president.gov.ua/documents/4472021-40013 URL: (дата звернення: 13.04.2023).

УДК 004 + 614.8

К.М. Марченко, О.В. Оришака
e-mail: k_marchenko@i.ua, oryhsaka@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ІНФОРМАЦІЙНА БЕЗПЕКА ЛЮДИНИ Й СУСПІЛЬСТВА В УМОВАХ ВІЙНИ

Інформаційна війна є невід'ємною складовою гібридних війн [1]. Розпочата в Україні повномасштабна гаряча війна привела до значної інтенсифікації інформаційної війни між Україною та Росією, до посилення інформаційних впливів, які вилилися у справжню інформаційну агресію та досягли світового масштабу.

Саме за допомогою інформації були створені передумови для початку війни, інформація у цій війні використовується як зброя, спрямована не тільки на виведення з ладу інформаційних систем противника, а й на боротьбу за свідомість та вибір людей. Зважаючи на масштаби руйнівного застосування інформаційної зброї та наслідки в суспільствах, можна стверджувати, що інформаційна зброя є різновидом зброї масового враження [2]. Лінія інформаційного фронту проходить скрізь мас-медіа, мережу Інтернет, соціальні мережі [3]. Головною ціллю інформаційної зброї є свідомість людини та масова свідомість суспільства [4]. Наслідками дії інформаційних впливів є хибні уявлення у людей про дійсність, примусовий вибір, психічні розлади тощо [5, 6].

Задачею досліджень є вивчення інформаційних впливів на людину та суспільство в умовах війни, а також розробка безпечних методів при користуванні інформацією та заходів, що забезпечують інформаційну безпеку життєдіяльності громадян.

В умовах повномасштабної гарячої війни, яка відбувається нині в Україні, інформаційні атаки набули характеру відкритості та агресивності, інформація, що наповнює інформаційний простір стає суперечливою та перекрученою. Непомірно зростає емоціональна складова, якою супроводжується інформація, що приводить до психічного напруження та розхитування кінцевого споживача, завдає шкоди як індивідуальному, так і суспільному здоров'ю. Зросла частка суб'єктивної інформації, яка складається з думок не завжди компетентних осіб. Так, наприклад, часто-густо новини супроводжуються словом "можливо".

З огляду на масштаби та наслідки інформаційних операцій та цілей, які ставляться, можна сказати, що шкода від застосування інформаційної зброї сумірна до шкоди, спричиненої застосуванням військової зброї, адже інформаційні операції є підґрунтям для операцій військових. Тож і відповідальність інформаційних злочинців повинна бути не меншою за відповідальність військових.

Методи протидії інформаційній агресії можна поділити на оперативні та довгострокові. До оперативних методів можна віднести надання населенню об'єктивної інформації, підтвердженої фактами, всебічний аналіз ситуацій та подій. До довгострокових методів відносяться державні програми та курси з навчання інформаційній безпеці та медіа-грамотності. Актуальними є широка пропаганда та розвинення інформаційної культури в інформаційному суспільстві, навчання правилам інформаційної гігієни, розсудливості та розбірливості при роботі з інформацією. Необхідно удосконалювати інформаційне законодавство та збільшувати відповідальність за якість і достовірність розповсюдженої інформації.

Список літератури

1. Гібридна війна і журналістика. Проблеми інформаційної безпеки : навчальний посібник / за заг. ред. В. О. Жадька ; ред.-упор. : О. І. Харитоненко, Ю. С. Полтавець. – Київ : Вид-во НПУ імені М. П. Драгоманова, 2018. – 356 с.

2. Бабенко Ю. Інформаційна війна – зброя масового знищення! URL: https://www.pravda.com.ua/rus/articles/2006/04/20/4399050/

3. Курбан О.В. Сучасні інформаційні війни в мережевому он-лайн просторі [Текст]: навчальний посібник / О.В. Курбан. – Київ: ВІКНУ, 2016. - 286 с.

4. Сасин Г. В. Інформаційна війна: сутність, засоби реалізації, результати та можливості протидії (на прикладі російської експансії в український простір) / Г. В. Сасин // Грані. - 2015. - № 3. - С. 18-23.

5. Почепцов Г. Сучасні інформаційні війни [Текст]. Вид. 2-ге, допов. Київ : КиєвоМогил. акад., 2016 - 502 с.

6. Горбулін В.П., Додонов О.Г., Ланде Д.В. Інформаційні операції та безпека суспільства: загрози, протидія, моделювання: монографія. — К. : Інтертехнологія, 2009. — 164 с.

УДК 004.056+004.75

К. О. Задорожний, Є. В. Мелешко
kostazadorozhnij9@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ ПРОБЛЕМ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В ОДНОРАНГОВИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ ТА ШЛЯХІВ ЇХ ВИРІШЕННЯ

Одноранговa комп'ютерна мережа, яка є частиною сильно розподілених систем, містить різноманітну кількість вузлів для формування комп'ютерної мережі. Ці вузли використовуються для обміну вмістом, що містить аудіо-, відео-, текстові дані та інші різні види файлів, без використання одного сервера, як в архітектурі клієнт-сервера. Такий тип організації робить комп'ютерну мережу дуже вразливою. У сфері інформаційної безпеки одноранговых мереж існує п'ять цілей: анонімність, доступність, автентифікація файлів, контроль доступу та чесна торгівля [1, 3].

Загальна характеристика одноранговых комп'ютерних мереж

Одноранговa комп'ютерна мережа – це технологія, що реалізує об'єднання одноранговых вузлів рівного статусу з собі подібними для обміну даними. Кожен вузол є і приймачем, і надавачем послуг [2]. Візьмемо для прикладу Bit Torrent, під час завантаження файлу ми завантажуємо різні частини файлу одночасно від різних одноранговых пристроїв, які, можливо, вже завантажили або завантажують цей файл. Це збільшує швидкість завантаження. Одноранговий вузол виступає як сервер, так і клієнт одночасно. Це самоорганізована та динамічно налаштована структура, оскільки кількість вузлів може збільшуватися та зменшуватися в будь-який час. Відповідний Peer несе відповідальність за пошук відповідної інформації. Якщо необхідні дані є загальними, їх легко знайти, але стає складніше знайти дані, які рідко доступні серед аналогів [1].

Класифікація атак на одноранговi комп'ютерні мережі

Загалом атаки на піринговi мережі поділяються на дві великі категорії: активні та пасивні. Активну атаку можна визначити як таку, що спрямована на один або декілька вузлів P2P-мережі. Основний мотив активної атаки – викликати пошкодження вузлів. Пасивні атаки спрямовані виключно на саму мережу, а не на її конкретні вузли. Основний мотив пасивної атаки – це порушення доступності, щоб учасники були обмежені у використанні конкретних послуг.

Ось деякі серйозні загрози інформаційній безпеці в одноранговых комп'ютерних мережах:

1. *Атака підслуховуванням (Eavesdropping attack).* Спрямована на те, що в P2P-мережі зловмисник може підключитися до мережі та «прослуховувати» зв'язок, що відбувається між вузлами. Це загрожує цілісності інформації та конфіденційності [1, 2].

2. *Комунікаційні перешкоди (Communication Jamming).* У цій атаці зловмисник може заблокувати мережу та порушити її роботу. Це може статися через постійне надсилання хибних даних, приєднання до мережі та виходу з неї, а також маршрутизації її, що призводить до відмови в обслуговуванні [1].

3. *Ін'єкція та модифікація (Injection and Modification).* Зловмисник може завантажити файли, що містять шкідливий код: віруси, хробаки тощо, і спробувати заразити вузли в мережі. Він також може передавати фальшиві файли, створюючи перевантаження в мережі та намагаючись знизити пропускну здатність [1].

4. *Атака «Сивілли» (Sybil attack).* Вона небезпечна тим, що зловмисник створює кілька псевдовузлів, і ці вузли розривають зв'язок між фактичними одноранговими вузлами, і обмін файлами не здійснюється [4].

5. *Атака «Відмова в обслуговуванні» (Denial of service, DoS)* – це атака на комп'ютер або мережу, що робить неможливим подальше використання їх сервісів. Існує два способи виконати таку атаку: наповнювати мережу фіктивними пакетами (при цьому цільова доставка істинного трафіку ускладнюється) або ж змусити вузли працювати над складними (вибагливими по ресурсам) обчисленнями, аби у вузла не було змоги відповісти на інші запити, що надходять [1 , 2, 4].

6. *Розподілена відмова в обслуговуванні (Distributed denial of service, DDoS).* В атаці залучено декілька хостів. В цьому випадку атакуючі комп'ютери часто є персональними комп'ютерами з широкосмуговою мережею з'єднання, що були уражені вірусом або зламані шкідливим програмним забезпеченням. Злочинець при цьому може дистанційно керувати цими машинами і спрямувати атаку на будь-який хост чи мережу. Вони можуть бути двох типів:

– *TCP DDoS Attack*. Атака, при якій зловмисник блокує всі сервіси жертви, позбавляючи її ресурсів для відновлення з'єднання. Це можна досягти за допомогою атаки "Отруєння індексу" (Index poisoning attack), коли зловмисник вставляє недійсні індекси в мережу, використовуючи індекс жертви або будь-який інший ідентифікатор. Таким чином, кожен раз, коли інший вузол запитує відповідний файл, він буде спрямований до системи жертви, що встановлює повністю відкрите TCP-з'єднання. Важливо відзначити, що жертва не обов'язково повинна бути частиною P2P-мережі для цієї атаки.

– *DDoS-атака на пропускну здатність (Bandwidth Attack)*. Спрямована на зайняття всієї доступної пропускної здатності жертви, що призводить до того, що жертва не може поділитися нічим з іншими користувачами. В такій ситуації зловмисник виступає як сусід, з яким пов'язані багато вузлів у таблиці маршрутизації, і кожен раз, коли вузол бажає поділитися чимось, він може вибрати зловмисника найближчим

сусідом замість вибору іншого нормального вузла. З урахуванням мільйонів користувачів, це може призвести до перевантаження пропускної здатності та спричинити DDoS-атаку на пропускну здатність [1, 2].

Можливі шляхи вирішення проблем безпеки в однорангових комп'ютерних мережах

З розвитком технологій та зростанням популярності однорангових мереж з'явилася необхідність у забезпеченні безпеки цих систем. На сьогоднішній день існує багато загроз та проблем, які потребують розгляду та вирішення. Було досліджено основні можливі шляхи вирішення проблем безпеки в однорангових комп'ютерних мережах.

Захист від атаки підслуховуванням. Першим кроком у запобіганні цієї атаки є використання міцної фізичної безпеки, а наступний крок – використання шифрування будь-якої важливої інформації.

Захист від атаки «Сивілли». Щоб захиститися від атаки «Сивілли», можна використовувати метод самореєстрації [7, 8]. Це може бути включення IP-адреси вузла до його ідентифікатора. Таким чином, зловмисний вузол не зможе підробити справжніх вузлів, і його можна помітити, якщо він створить більше сутностей, так як він буде прив'язаний до обмеженої кількості IP-адрес. Проте це рішення не є повністю безпечним, оскільки існує можливість генерації підроблених ідентифікаторів для інших вузлів, що можуть бути помічені системою як зловмисні. Іншим способом захисту є використання складного протоколу на основі публічно-приватного ключа. Кожен вузол повинен підписувати свої повідомлення та періодично відповідати на запити з боку центрального вузла, але цей метод також має свої слабкі місця.

Захист від DoS-атаки. Щоб захистити мережу від такої атаки використовують «ціноутворення». Оскільки неможливо відрізнити справжні важкі обчислення, що можуть виконуватись вузлом, від фіктивних, тому неможливо вчасно виявити і запобігти цій атаці. Тому вузол, що надає послугу, повинен надавати своїм клієнтам так звані «головоломки», адже лише у випадку її вирішення клієнту можна довіряти та відповідати на запит. Проте недоліком цього методу є те, що деякі справжні клієнти (наприклад, мобільні пристрої) можуть швидше розряджатись, вирішуючи ці «головоломки». Також можна блокувати вразливі порти, на які можуть здійснюватися атаки. Ще одним способом є додавання спеціальних правил до брандмауера та використання спеціалізованих маршрутизаторів [5].

Захист від DDoS-атак. Існує три кроки, які допоможуть запобігти DDoS-атакам. Спочатку необхідно пропускати Інтернет-трафік через брокерську компанію, щоб допомогти клієнтам відфільтрувати шкідливу інформацію. По-друге, необхідно використовувати моніторинг взаємодії вузлів для виявлення DDoS-атак. Також брокери можуть мати свій власний чорний та/або білий список, які дозволяють їм припинити роботу трафіку у чорному списку до того, як він потрапить до кінцевого користувача. При цьому брокери завжди дозволяють трафік з білого списку [6].

Висновки. Дослідження проблем інформаційної безпеки в однорангових комп'ютерних мережах демонструє, що ці мережі вразливі до різноманітних атак, які можуть стати причиною втрати даних, втрати конфіденційності, та перерви у зв'язку між вузлами. Можливі шляхи вирішення проблем інформаційної безпеки в однорангових комп'ютерних мережах включають застосування шифрування, аутентифікації, контролю доступу та інших методів захисту, які дозволяють забезпечити інформаційну безпеку комп'ютерної мережі та унеможливити інформаційні атаки з боку зловмисників. У зв'язку з тим, що однорангові комп'ютерні мережі стають все більш популярними, вирішення проблем інформаційної безпеки є критично важливою задачею для забезпечення ефективної роботи мережі та захисту від можливих загроз. Для досягнення максимального рівня інформаційної безпеки необхідно поєднувати різні методи та підходи, що дозволить створити ефективну систему захисту однорангових комп'ютерних мереж від різних типів інформаційних атак.

Список літератури

1. P. Wararkar, N. Kapil, V. Rehani, Y. Mehra and Y. Bhatnagar, "Resolving Problems Based on Peer to Peer Network Security Issue's," Procedia Computer Science, vol. 78, 2016, pp. 652-659. – URL: https://www.sciencedirect.com/science/article/pii/S1877050916001150

2. Л. Куперштейн, М. Кренцін, А. Дудатьєв, і В. Каплун, "Аналіз проблем безпеки піринговых мереж," Інформаційні технології та комп'ютерна інженерія, Вип. 2(54), С. 5-14, 2022. – URL: https://itce.vntu.edu.ua/index.php/itce/article/view/881

3. Peer-to-peer – URL: https://uk.wikipedia.org/wiki/Peer-to-peer

4. L. Washbourne, "A Survey of P2P Network Security," Cryptography and Security, 2015. – URL: https://arxiv.org/ftp/arxiv/papers/1504/1504.01358.pdf

5. L. Kupershtein, T. Martyniuk, O. Voitovych, B. Kulchytskyi, A. Kozhemiako et al. "DDoS-attackde-tection using artificial neural networks in Matlab," Proc. SPIE 11176, Photonics Applications in As-tronomy, Communications, Industry, and High-EnergyPhysics Experiments, 2019, doi: https://doi.org/10.1117/12.2536478

6. І. С. Каплун, "Прогнозування та аналіз DDOS – атак на інформаційні web-ресурси," Молодь в науці: дослідження, проблеми, перспективи, Вінницький національний технічний університет, 2018. – URL: https://conferences.vntu.edu.ua/index.php/mn/mn2018/paper/viewFile/5646/4789

7. W. Stallings, "Cryptography and Network Secrity: Principles and Practices," 7th edition, Pearson, 2016, 768 p. (ISBN 13: 978-1-292-15858-7)

8. A. Vasudeva, M. Sood, "Survey on sybil attack defense mechanisms in wireless ad hoc networks," Journal of Network and Computer Applications, vol. 120, pp. 78-118, 2018. doi: https://doi.org/10.1016/j.jnca.2018.07.006

УДК 004.056+004.75

І.В. Варченко, Є.В. Мелешко
mrcrazyyt67@gmail.com, elismeleshko@gmail.com
[1]Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ КІБЕРАТАК ТА ШКІДЛИВИХ ПРОГРАМ З ВИКОРИСТАННЯМ ЛЮДСЬКОГО ФАКТОРУ

Комп'ютерний вірус – це вид шкідливого програмного забезпечення, яке здатне поширювати свої копії з метою інфікування та пошкодження даних на пристрої жертви. Віруси потрапляють на комп'ютер через Інтернет, через носії інформації (наприклад, CD, DVD тощо). Основні види вірусів:

– Файлові – інфікують файли з розширеннями «.exe» або «.com».

– Скриптові – підвид файлових вірусів, написаних на різних мовах скриптів (VBS, JavaScript, BAT, PHP тощо). Цей тип здатний інфікувати інші формати файлів, наприклад HTML.

– Завантажувальні – атакують завантажувальні сектори змінних носіїв (диски, дискети та флеш-накопичувачі), встановлюючись під час запуску пристрою.

– Макровіруси вбудовуються в програми для роботи з текстами або електронними таблицями. Прикладом є віруси, що вбудовуються у Microsoft Word (.doc) та Microsoft Excel (.xls).

Також існують цільові та нецільові атаки і окремо виділяють кіберзброю.

Як правило нецільові атаки конкретно ні на кого не направлені. Прикладом таких атак може бути: троянський вірус, фішингові атаки, homograph attack, Drive by Download, тощо.

Термін «троянський вірус» є дещо неточним, але зазвичай використовується замість більш правильного терміну «троян». Вірус заражає звичайні комп'ютерні файли – захоплює окремий файл і руйнує чи зловмисно змінює його у процесі. Потім він спробує поширитись на інші комп'ютери, заражаючи інші файли. Вірус часто маскується під безкоштовне програмне забезпечення або вкладення в електронному листі, а потім, як тільки користувач дає дозвіл додатку на встановлення на свій ПК, вірус відкриває шлюзи. Як тільки у трояна з'являється доступ до комп'ютера користувача, він може робити будь-що, але більшість цих шкідливих програм прагнуть отримати повний контроль над комп'ютером. Іншими словами, всі дії на комп'ютері користувача записуються і відправляються на сервер, вказаний трояном. За допомогою троянів зловмисники можуть перетворити комп'ютер на «зомбі» і використовувати його для запуску кібератак по всьому світу.

Фішинг (англ. phishing, від fishing – риболовля, вивужування і password – пароль) – вид інтернет-шахрайства, мета якого – отримати ідентифікаційні дані користувачів. Сюди належать крадіжки паролів, номерів кредитних карток, банківських рахунків та іншої конфіденційної інформації. Прикладом фішингу можуть бути підроблені повідомлення від банків, провайдерів, платіжних систем та інших організацій, що прийшли на пошту про те, що з якоїсь причини одержувачу терміново потрібно передати або оновити особисті дані. Вказані причини можуть бути різні. Це може бути втрата даних, поломка в системі та інше. Зайшовши на підроблений сайт, користувач вводить у відповідні рядки свій логін та пароль, а далі аферисти отримують доступ у кращому випадку до його поштової скриньки, у гіршому – до електронного рахунку.

Homograph attack – це метод обману, за допомогою якого зловмисник використовує схожість сценаріїв символів, щоб створити та зареєструвати фальшиві домени існуючих сайтів та обдурити користувачів і заманити на підроблені веб-ресурси. Homograph attack і фішинг пов'язані між собою, так як за допомогою даного методу можна зробити підробку на потрібний сайт, наприклад, зловмисник може створити сайт google.com, де «о» будуть не англійські, а український, підробити дизайн сайту і використовувати у фішингових листах.

Атаки Drive-By Download – це випадки, коли зловмисник може змусити користувача завантажити програмне забезпечення, просто відвідавши веб-сайт. Користувач відвідує скомпрометований веб-сайт. Веб-сайт або надсилає фоновий запит, або виконує перенаправлення запиту на веб-сайт зловмисника-контролера. Це можна зробити тихо, сповістивши користувача за допомогою JavaScript і iframe HTML. Запит завантажує «набір експлойтів», який перевіряє користувача на наявність великої кількості відомих вразливостей у його операційній системі, браузері чи плагінах браузера. Flash, Java, програми для читання PDF-файлів і відеопрогравачі часто є вразливими цілями. Після виявлення експлойта він може використовувати його, щоб змусити браузер користувача завантажити «корисне навантаження» зловмисного програмного забезпечення. Корисне навантаження може відображати рекламу в браузері користувача, зробити його комп'ютер частиною ботнету, реєструвати натискання клавіш або викрадати облікові дані, захоплювати банківські сеанси, інсталювати програми-вимагачі або встановити бекдор, щоб надати зловмиснику доступ у майбутньому.

Цільові атаки, як правило, направлені на компанії та конкретних людей. Тут важливо розуміти, що частіше всього захист у компаніях дуже сильний і просто так зламати не вийде, тож, для взлому можуть використовуватись підставні сайти, фішингові листи, програми, що імітують роботу потрібного компанії ПЗ або начебто нове оновлення для ПЗ. Прикладом може бути атака на Центральний банк Бангладеш, що була однією з найбільших хакерських крадіжок в історії. У лютому 2016 року зловмисники використали систему

SWIFT для переказу майже $1 млрд з рахунку банку в Нью-Йорку на різні рахунки в Азії. Однак через помилку в написанні слова "foundation" (фонд) в одному з переказів, операцію було помічено та зупинено, і їм вдалося вкрасти лише 100 млн. З вкрадених грошей вдалося повернути лише близько $15 млн. Людський фактор грав велику роль у цій атаці. З одного боку, хакери використовували соціальну інженерію для отримання доступу до комп'ютерів співробітників банку та встановлення шпигунського ПЗ. З іншого боку, слабкий захист банку та низький рівень контролю з боку SWIFT також сприяли успіху атаки.

Кіберзброя. На даний момент є лише один приклад такого виду вірусів і це Stuxnet і його модифікації. Stuxnet – вважається першою в світі кіберзброєю. Використовуючи 4 вразливості «нульового дня», атака була спрямована на програмовані логічні контролери (PLC). Потрапляючи на пристрій вірус перевіряв його на ознаки програмного забезпечення Siemens Step 7. Промислові комп'ютери, що служать ПЛК, використовують для автоматизації та моніторингу електромеханічного обладнання. Після виявлення потрібного ПК, вірус оновлював код програми через Інтернет та надсилав помилкові інструкції що викликають пошкодження, до електромеханічного обладнання, яким керував ПК. При цьому вірус надсилав головному контролеру помилковий відгук. Будь-хто, хто спостерігав за обладнанням, не мав би ознак проблеми, доки обладнання не почало б самознищуватися.

Висновок. Найслабша ланка в системах інформаційної безпеки не технології, а людина. Який би сильний захист комп'ютерних систем не був у компаніях, поки в цьому захисті існують люди – хакерські атаки будуть існувати. Також з цього випливає, що в першу чергу треба вчити людей кіберграмотності, якщо вони працюють в компанії особливо тому, що частіше всього хакери роблять акцент на їх не грамотності або на якісь слабкості типу «писати лист так щоб його хотілось відкрити». Також не потрібно забувати і про простих людей які не працюють в компаніях, але можуть бути поширювачами якогось вірусу, який для них не шкідливий. Поки існують люди які вважають що старіші версії програм, операційних систем, браузерів тощо «стабільніші», то поширювачів якогось вірусу буде тільки більше.

Список літератури

1. What Is Stuxnet? [Електронний ресурс] // Trellix – Режим доступу до ресурсу: https://www.trellix.com/en-us/security-awareness/ransomware/what-is-stuxnet.html
2. Комп'ютерний вірус [Електронний ресурс] – Режим доступу до ресурсу: https://www.eset.com/ua/support/information/entsiklopediya-ugroz/kompyuternyy-virus/
3. Хакерська крадіжка золотовалютних резервів Бангладеш [Електронний ресурс] / Вікіпедія. – 2022. – Режим доступу до ресурсу: https://uk.wikipedia.org/w/index.php?title=Хакерська_крадіжка _ золотовалютних_резервів_Бангладеш
4. Граматика підвела: хакери упустили куш в $1 мільярд [Електронний ресурс] // Укрінформ. – 2023. – Режим доступу до ресурсу: https://www.ukrinform.ua/rubric-world/1980314-gramatika-pidvela-hakeri-upustili-kus-v-1-milard.html
5. Фішинг [Електронний ресурс] – Режим доступу до ресурсу: https://www.eset.com/ua/support/information/entsiklopediya-ugroz/fishing/
6. Drive By Downloads [Електронний ресурс] – Режим доступу до ресурсу: https://guides.codepath.com/websecurity/Drive-By-Downloads
7. Троян: [Електронний ресурс] – Режим доступу до ресурсу: https://www.eset.com/ua/support/information/entsiklopediya-ugroz/troyan/

УДК 629.7.02

Д.О. Радецкий, Р.М. Минайленко, О.К. Конопліцька-Слободенюк, О.К. Савеленко
aron70@ukr.net
Центральноукраїнський національний технічний університет, м. Кропивницький

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ ЖИВУЧОСТІ КОМП'ЮТЕРНОЇ МЕРЕЖІ АСУ ПІДПРИЄМСТВОМ НА ОСНОВІ КЛАСТЕРНОЇ ТЕХНОЛОГІЇ

У сучасному виробництві застосовуються системи автоматизованого керування підприємством (бухгалтерія, склади, оперативне планування) для таких систем важлива надійність і швидкість обробки даних. Існує чимало засобів побудувати надійну систему. Дискові масиви RAID, наприклад, дозволяють не переривати обробку запитів до інформації, що зберігається на дисках, при виході з ладу одного або декількох елементів масиву. Не бракує технологій, що гарантують надмірність інших підсистем серверу. Так, резервні блоки живлення дозволять скорегувати відмову цієї компоненти. Джерела безперебійного живлення підтримають працездатність системи у випадку збоїв у мережі енергопостачання. Багатопроцесорні материнські плати забезпечать функціонування серверу у випадку відмови процесора.

Проте жодний із цих варіантів не врятує, якщо з ладу вийде вся обчислювальна система цілком. Саме у цьому випадку на допомогу приходить кластеризація. Першим кроком до створення кластерів можна вважати широко поширені в пору розквіту міні-комп'ютерів системи "гарячого резерву". За цією технологією в мережі з декількох серверів один або два не виконують ніякої корисної роботи, але готові почати функціонувати, як тільки вийде з ладу яка-небудь основна система.

Найпростіший варіант кластеризації це - віддзеркалення, коли один із мережних серверів виступає в ролі "дзеркала" для іншого. На "дзеркальному" сервері встановлене те ж саме програмне забезпечення, що і на основному, синхронізація програмного забезпечення підтримується шляхом передач даних по комунікаційному каналу, що з'єднує сервери в кластері (синхронізація серверів). Збій виявляє "дзеркало" і ініціює виконання програми на ньому, при цьому всі запити опрацьовуються точно так само, як і на основному сервері, за рахунок синхронізації роботи серверів. Таку конфігурацію кластера називають "активний/резервний".

Так само існує конфігурація "активний/активний" - це два сервери, що підмінюють один одного у випадку збою. Наприклад, Web-сервер і сервер додатків в одній мережі можуть виконувати функції "дзеркала" один для іншого. Якщо вийде з ладу Web-сервер, його функції візьме на себе сервер додатків і навпаки. Деякі програмні рішення навіть дозволяють будь-якому комп'ютеру в мережі виступати в якості "дзеркала" для будь-якої іншої машини в тій самій мережі. За такою логікою підмінити Web-сервер зможе, наприклад, високопродуктивна робоча станція користувача, або ж група таких станцій, між якими буде розподілене все обчислювальне навантаження. З погляду користувачів основний і дзеркальний сервер - це єдина система, користувачі не знають і не можуть дізнатися, який із серверів на даний момент обслуговує його запити.

Перед кластерами поставлені дві задачі: потужні обчислення і підтримка розподілених баз даних з високим рівнем готовності, особливо таких, для яких потрібна підвищена надійність.

Можливість кластерної технології дозволяє насамперед побудувати унікальну архітектуру, що має достатню продуктивність, стійкість до відмов апаратури або програмного забезпечення і при цьому легко нарощується і модернізується, але універсальними засобами, зі стандартних компонентів і за помірну ціну (незрівнянно нижчу, ніж ціна унікального стійкого до збоїв комп'ютера або системи з масовим паралелізмом).

УДК 629.7.02

О.О. Стукаленко, Р.М. Минайленко, О.К. Конопліцька-Слободенюк, Н.М. Якименко
aron70@ukr.net
Центральноукраїнський національний технічний університет, м. Кропивницький

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ ІОТ НА ОСНОВІ СИСТЕМИ ВИЯВЛЕННЯ ЗОВНІШНІХ ВТОРГНЕНЬ

Інтернет речей (англ. Internet of Things, IoT) — концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані датчики, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами, за допомогою використання стандартних протоколів зв'язку. Окрім датчиків, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитувати та обробляти дані для автоматизації процесів, дозволяючи виключити необхідність участі людини.

Кількість пристроїв Internet of Things (IoT), збільшується, але, як показують дослідження, багато з цих пристроїв є принципово небезпечними, наражаючи інтернет та всіх його користувачів на різноманітні атаки. Найпоширенішим методом використання пристроїв IoT є зовнішні впливи (ботнети). Наприклад, такі ботнети як Mirai, використовують незахищені пристрої IoT, щоб здійснити атаки DDoS на критичну інтернет-інфраструктуру.

За прогнозами аналітиків у найближчі роки очікується справжній бум інтернету речей. Так, за прогнозами Gartner, до 2020 року кількість підключених до всесвітньої мережі пристроїв становитиме 26 мільярдів, а дохід від продажу устаткування, програмного забезпечення та послуг становитиме 1,9 трлн дол. Деякі інші аналітичні агентства висловлюють ще більш оптимістичні прогнози.

Найбільші світові ІТ компанії вже почали перегони за лідерство на ринку ІОТ. Так корпорація Intel у 2014 році після випуску «SoC Edison» оголосила конкурс «*Make it Wearable*» з призовим фондом $1,3 млн на найкраще застосування своєї системи для концепції IoT та створила власний підрозділ «Internet of Things Solutions Group» для розвитку цього напрямку. Компанія «Google» на початку 2014 року за 3,2 млрд доларів купила невелику фірму «Nest Labs», яка займається випуском інтелектуальних термостатів. Виробники побутової техніки також працюють у цьому напрямку. Так на виставці CES 2014 у Лас- Вегасі була представлена велика кількість побутової техніки (холодильники, телевізори, пральні машини) з можливістю підключення до інтернет.

Лідерами у розробці та впровадженні інтернету речей є країни, в яких розвинена індустрія виробництва мікропроцесорів та вбудованих комп'ютерів; це США, Китай, Південна Корея. Також значний прогрес у цій галузі демонструють європейські країни та Японія.

Інтернет речей може викликати величезні зміни у повсякденному житті, надавши звичайним користувачам абсолютно новий рівень комфорту. Але якщо елементи такої системи не будуть належним чином захищені від несанкціонованого втручання, за допомогою надійного криптографічного алгоритму, замість користі вони принесуть шкоду, надавши кіберзлочинцям лазівку для підриву інформаційної безпеки.

Основною концепцією IoT є можливість підключення об'єктів (речей), які людина може використовувати в повсякденному житті, наприклад, холодильник, кондиціонер, автомобіль, велосипед і навіть кросівки. Всі ці об'єкти (речі) повинні бути оснащені вбудованими датчиками або сенсорами, які мають можливість обробляти інформацію, що надходить з навколишнього середовища, обмінюватися нею і виконувати різні дії в залежності від отриманої інформації. Прикладом впровадження такої концепції є система «розумний будинок» або «розумна ферма». Ця система аналізує дані навколишнього середовища і в залежності від показників регулює температуру в приміщенні. У зимовий період регулюються інтенсивність опалення, а в разі спекотної погоди будинок має механізми відкривання і закривання вікон, завдяки чому провітрюється будинок, і все це відбувається без втручання людини.

Кількість пристроїв Internet of Things (IoT), постійно збільшується, але, як показують дослідження, багато з цих пристроїв є принципово небезпечними, наражаючи інтернет та всіх його користувачів на різноманітні атаки.

Таке становище мотивувало до розробки нових методів реагування на атаки IoT та протидії зовнішнім вторгненням. Матеріали данного дослідження можуть бути використані для аналізу специфічних для IoT мережевих моделей поведінки (наприклад, обмежена кількість кінцевих точок або регулярні часові проміжки між передачею пакетів даних), що може допомогти точно визначити параметри для збору та аналізу мережевого трафіка пристроїв IoT. Визначення таких параметрів є ключовим фактором в подальшому дослідженні та створенні програмного забезпечення системи системи кібербезпеки IoT на основі системи виявлення зовнішніх вторгнень.

УДК 004.8+004.056

Р. О. Ткачук, Д. В. Бащенко, Є. В. Мелешко
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ ТА СПОСОБІВ ЇХ ЗАСТОСУВАННЯ ДЛЯ ВИЯВЛЕННЯ ІНФОРМАЦІЙНИХ АТАК НА КОМП'ЮТЕРНІ МЕРЕЖІ

Метою даної роботи було дослідити, які види нейронних мереж можна використовувати для виявлення інформаційних атак на комп'ютерні мережі.

Як показало дослідження, для аналізу мережного трафіку та виявлення аномалій або інформаційних атак можна використовувати нейронні мережі, які здатні моделювати часові та просторові залежності в даних, зокрема, багатошарові нейронні мережі (MLP) [1], рекурентні нейронні мережі (RNN) [2, 3], згортково-рекурентні мережі (CRNN) [4], графові нейронні мережі (GNN) [5, 6].

Аналіз мережного трафіку за допомогою нейромереж для виявлення атак потребує таких кроків:

– *Обробка даних*. Мережевий трафік є потоком пакетів, які мають різні параметри, такі як IP-адреси, порти, протоколи тощо. Для аналізу цих даних необхідно перетворити їх на числові або категоріальні ознаки, які можуть бути подані на вхід нейронної мережі. Наприклад, можна використати *швидке кодування* (one-hot encoding) для категоріальних ознак або нормалізацію для числових ознак. Також можна використати методи відбору ознак для зменшення розмірності даних та усунення шуму чи надмірності.

– *Вибір архітектури нейронної мережі*. Залежно від характеру даних та мети аналізу можна вибрати різні типи нейронних мереж, наприклад, багатошаровий перцептрон (MLP), рекурентну нейронну мережу (RNN), згорткову нейронну мережу (CNN), графову нейронну мережу (GNN) тощо. Кожен тип нейромережі має свої переваги та недоліки, а також специфічні параметри. Є доцільним для рішення конкретної задачі використати різні нейромережі, порівняти їх ефективність у навчанні, і тільки після обрати найкращу для неї.

– *Навчання нейронної мережі*. Для навчання нейронної мережі необхідно розділити дані на навчальну, валідаційну та тестову вибірки. Навчальна вибірка використовується для налаштування ваги нейронної мережі за допомогою алгоритму оптимізації, такого як стохастичний градієнтний спуск (SGD) або адаптивний градієнтний спуск (Adam). Валідаційна вибірка використовується для перевірки якості навчання та регуляризації моделі, наприклад, за допомогою ранньої зупинки (early stopping) або методу відсіву (dropout). Тестова вибірка використовується для оцінки якості роботи мережі на нових даних.

– *Оцінка та інтерпретація результатів*. Для оцінки результатів аналізу мережного трафіку можна використовувати різні метрики та засоби, такі як точність (precision), повнота (recall), F-мера, ROC-крива тощо. Це дозволяє виміряти, наскільки добре модель здатна класифікувати трафік на нормальний та аномальний, виявляти різні типи атак. Для інтерпретації результатів можна використовувати різні методи візуалізації чи пояснення роботи моделі, наприклад, методи важливості ознак чи локальних апроксимацій. Ці методи дозволяють зрозуміти, які ознаки чи частини даних впливають на рішення моделі і чому.

Нейронні мережі є потужним інструментом для аналізу мережевого трафіку та виявлення кібератак, оскільки вони здатні навчатися та адаптуватися до нових типів загроз. Але вони вимагають передобробки даних, вибору архітектури, навчання та оцінки результатів. Для кожного етапу існують різні методи та параметри, які потрібно підбирати залежно від характеру даних та мети аналізу. Тож, розробникам доводиться багато експериментувати та підбирати потрібну архітекту і параметри. Також нейронні мережі мають високу обчислювальну складність, необхідність великого обсягу навчальних даних, ризик перенавчання чи недонавчання, складність інтерпретації результатів тощо, що необхідно враховувати.

Список літератури

1. Wang M., Lu Y., Qin J. A dynamic MLP-based DDoS attack detection method using feature selection and feedback // Computers & Security, Vol. 88, 2020, doi: https://doi.org/10.1016/j.cose.2019.101645

2. Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network // Physica D: Nonlinear Phenomena, Vol. 404, 2020.

3. Sivamohan S., Sridhar S. S., Krishnaveni S. An effective recurrent neural network (RNN) based intrusion detection via bi-directional long short-term memory // In 2021 international conference on intelligent technologies (CONIT), IEEE, 2021, pp. 1-5.

4. Liu H., Lang B., Liu M., Yan H. CNN and RNN based payload classification methods for attack detection // Knowledge-Based Systems, Vol. 163, 2019, pp. 332-341.

5. Scarselli F., Gori M., Tsoi A. C., Hagenbuchner M., Monfardini G. The Graph Neural Network Model // IEEE Transactions on Neural Networks, Vol. 20, Issue 1, 2009, pp. 61-80.

6. Zhang B., Li J., Chen C., Lee K., Lee I. A Practical Botnet Traffic Detection System Using GNN // In: Meng, W., Conti, M. (eds) Cyberspace Safety and Security, CSS 2021. Lecture Notes in Computer Science(), vol. 13172, Springer, Cham, 2022, doi: https://doi.org/10.1007/978-3-030-94029-4_5

УДК 629.7.02

П.І. Толкачьов, Р.М. Минайленко, О.К. Конопліцька-Слободенюк, Л.І. Поліщук
aron70@ukr.net
Центральноукраїнський національний технічний університет, м. Кропивницький

ВИКОРИСТАННЯ МІКРОКОНТРОЛЕРІВ СЕРІЇ RASPBERRY PI ДЛЯ СТВОРЕННЯ СИСТЕМИ МОНІТОРИНГУ ТА КІБЕРБЕЗПЕКИ IOT- ПРИСТРОЇВ

Інтернет речей є наступним кроком на шляху до оцифрування сучасного суспільства, де предмети і люди пов'язані один з одним через комунікаційні мережі і з'являється можливість повідомляти про їх стан та стан навколишнього середовища. Розробники таких програм намагаються якомога більше удосконалювати свої роботи. Інтернет речей створює нові можливості та забезпечує конкурентні переваги для бізнесу як на існуючих, так і на нових ринках.

У найближче десятиліття завдяки інтернету речей (IoT) очікується сплеск обсягу даних, генерованих підприємствами, що впливатиме на ефективність і конкурентоспроможність бізнесу..

Інтернет речей називають третьою хвилею інформаційної революції. Другий було поширення смартфонів і мобільних комунікацій. Четвертою, ймовірно, буде штучний інтелект, він візьме на себе подальший розвиток інформаційного середовища, в яку будуть інтегруватися люді. IoT - це не тільки мільярди речей, підключених до мережі, але і новий спосіб управління бізнес-процесами. Уже зараз технології інтернету речей набули широкого поширення. Наприклад, виробники авіадвигунів, будівельної техніки можуть стежити за роботою виробів в режимі реального часу, а керуючі компанії отримувати інформацію за показниками лічильників жителів району. Інтернет речей генерує величезну кількість інформації і швидко росте.

З метою створення системи кібербезпеки IoT-пристроїв та досягнення поставлених цілей пропонується міні-ПК Raspberry Pi. Він здатний забезпечувати як і прості практичні завдання, наприклад вивчення комп'ютера і основ роботи з ним, інтернет-серфінгу, а також для програвання відео- та прослуховування аудіофайлів, так і більш складні задачі, наприклад бути частиною IoT системи.

Raspberry Pi - це мініатюрний, розміром з кредитну карту, комп'ютер вартістю близько 25 доларів за базову модель і 35 - за більш новітню, який має величезну популярність і розійшовся по світу в кількості більше 4,5 мільйонів штук.

Однією з основних переваг міні-ПК Raspberry Pi є співвідношення якості продуктів і його вартості. В першу чергу, міні-ПК Raspberry Pi відводиться роль комп'ютера, призначеного для вивчення з його допомогою базових інформаційних технологій в школі. Позиціонується міні-ПК Raspberry Pi як дешеве рішення для початківців розробників. З урахуванням задовільною потужності цього пристрою, низького енергоспоживання і малої собівартості його можна використовувати для створення особистого міні сервера.

Стосовно до статистичного управління процесами – методу моніторингу виробничих процесів з метою управління якістю продукції безпосередньо в процесі виробництва, міні-ПК Raspberry Pi є першим доступним технічним рішенням такого розміру, яке можна використовувати повсюдно для програмування на багатьох мовах і в якості мікроконтролерів для управління роботизованими пристроями.

Одна з головних і привабливих особливостей міні-ПК Raspberry Pi - наявність на платі апаратних портів введення / виведення GPIO (General Purpose Input / Output, інтерфейс вводу / виводу загального призначення), що відкриває перспективи використання його в робототехнічних проектах і пристроях «розумного будинку»

У роботі пропонується система моніторингу та захисту IoT-пристроїв на основі міні-ПК Raspberry Pi. Розробка може контролювати стан датчиків через Інтернет, коли оновлення інформації на веб-сервері зчитується за певним алгоритмом, що подається в Raspberry Pi, а потім система відповідає на відповідні інструкції з високим ступенем захисту. Користувач може безпосередньо входити в систему та взаємодіяти із вбудованим пристроєм у режимі реального часу.

Система гнучка для розміщення широкого спектру вимірювальних приладів з відповідними інтерфейсами. Вона має різноманітні переваги, такі як енергоефективність, інтелект, низька вартість, портативність та висока продуктивність.

УДК 004.056.55

Н.О. Щур
старший викладач кафедри комп'ютерної інженерії та кібербезпеки,
thalitana@ztu.edu.ua
Державний університет «Житомирська політехніка», м. Житомир

ШИФРУВАННЯ ДАНИХ У PYTHON ЗА ДОПОМОГОЮ МОДУЛЯ FERNET

Fernet являє собою криптографічний модуль для легкого та надійного шифрування даних у програмах на мові Python. Шифрування з використанням Fernet дозволяє захистити дані від несанкціонованого доступу та гарантувати їх конфіденційність. Fernet є кросплатформним, тому його можна використовувати для будь-якої операційної системи.

Шифрування виконується за допомогою симетричного алгоритму AES-128 у режимі CBC. А для кодування шифрованого тексту використовується стандарт Base64, що дозволяє представляти його в різних форматах, зокрема у вигляді URL-адреси. Крім того, Fernet надає можливість додавати мітку часу після шифрування для підвищення безпеки.

Щоб використати Fernet у своїй програмі, потрібно відкрити командний рядок та встановити бібліотеку cryptography за допомогою команди pip. Наприклад, *pip install cryptography* (для Windows) та *pip3 install cryptography* (Для Linux/macOS).

Fernet має три основні методи для створення ключів, шифрування та дешифрування даних [1]:

- generate_key() – метод використовується для генерації випадкового ключа довжиною 128 біт. Ключ генерується за допомогою криптографічно надійного генератора випадкових чисел, тому він є достатньо випадковим та безпечним для шифрування;
- encrypt() – метод використовується для шифрування даних із використанням секретного ключа та створення зашифрованого токену. Цей метод приймає байтовий рядок даних для шифрування та повертає зашифрований токен;
- decrypt() – метод використовується для розшифрування зашифрованого токену із використанням секретного ключа та повертає байтовий рядок з розшифрованими даними.

Токени Fernet – це зашифровані дані та додаткова інформація про час створення та час дії токену, закодовані відповідно до специфікації Base64url. Кодування Base64url токена Fernet є конкатенацією декількох полів [2] (табл. 1).

Таблиця 1
Поля токену Fernet

Поле токена	Розмірність (в бітах)	Опис
Версія	8	Число, яке вказує, яка версія Fernet використана, наразі існує лише одна версія, її значення дорівнює 128 у десятковій чи 0x80 у шістнадцятковій системі числення
Мітка часу	64	Містить кількість секунд, що минули з 1 січня 1970 року за UTC та датою створення токену
IV	128	Унікальний вектор ініціалізації, що є набором випадкових чисел, згенерованих за допомогою методу os.urandom() та використовується для шифрування та дешифрування за алгоритмом AES
Зашифрований текст	число кратне 128	Містить зашифроване вихідне повідомлення, останній блок якого доповнюється перед шифруванням до 128-бітного блоку AES. Fernet реалізує доповнення блоку за допомогою алгоритму PKCS#7
HMAC	256	Наведені вище чотири поля об'єднуються, а потім перетворюються за допомогою коду автентифікації повідомлень HMAC із використанням хеш-функції SHA-256. Таким чином забезпечується цілісність та автентичність даних. Варто зауважити, що вхідні дані для HMAC не закодовані Base64url

Отже, токени містять зашифроване текстове повідомлення, а також інформацію, необхідну для перевірки цілісності цього повідомлення. Це означає, що токени забезпечують конфіденційність, цілісність і автентифікацію даних. Після створення токену Fernet його можна передавати в інші системи, зберігати у базі даних, або використовувати для передачі конфіденційної інформації по мережі.

При розшифруванні токену потрібно спочатку виконати його декодування за допомогою специфікації Base64url. Після чого переконатися, що перший байт маркера дорівнює 0x80 і що маркер не застарів. Далі повторно обчислити HMAC та перевірити чи він збігається з HMAC, який уже зберігається у токені. Для розшифрування тексту потрібно використовувати той самий ключ, який був використаний для його шифрування.

Розглянемо приклад використання Fernet для шифрування і розшифрування тексту в Python (рис. 1). У цьому прикладі ми спочатку генеруємо ключ за допомогою методу generate_key(). Після чого створимо об'єкт Fernet, передавши йому згенерований ключ. Також, як було зазначено вище, ключ для Fernet генерується випадковим чином, тому при кожному запуску програми буде створюватися новий ключ.

Важливо пам'ятати, що перед шифруванням повідомлення необхідно перетворити його на байти, тому що Fernet шифрує лише байти. Тобто перед шифруванням текст необхідно закодувати у байтовий рядок за допомогою методу encode(), а після розшифрування – декодувати назад у рядок з використанням методу decode().

```python
# імпортуємо модуль Fernet з бібліотеки cryptography
from cryptography.fernet import Fernet

# генеруємо ключ
key = Fernet.generate_key()

# створюємо об'єкт Fernet з ключем
fernet = Fernet(key)

# текст, який будемо шифрувати
message = "Hello, World!".encode()

# шифруємо дані та створюємо токен
token = fernet.encrypt(message)

# виводимо зашифрований токен
print(token)

# розшифровуємо токен та виводимо результат
decrypted_message = fernet.decrypt(token)
print(decrypted_message.decode())
```

Рис. 1. Шифрування тесту за допомогою Fernet

Шифруємо текст за допомогою методу encrypt(). Зашифрований текст ми виводимо на екран. Далі розшифровуємо текст за допомогою методу decrypt() і виводимо розшифрований текст на екран (рис. 2).

```
b'gAAAAABj9kB9zCDtvVNb8uqCHAO_4kU0t073wH9puYSIiB6924pa5y1P4QfapP367FQV7t_cZJtpnFb25Tf7zFDZXrxnXbmnbw=='
Hello, World!
```

Рис. 2. Зашифрований та розшифрований текст

Також за допомогою Fernet можна зашифрувати та розшифрувати файли у Python. У прикладах нижче (рис. 3-4), файли plaintext.txt, encrypted.bin та decrypted.txt повинні знаходитись у тій самій папці, що і файл key.key. Файл key.key містить секретний ключ для шифрування та розшифрування файлів. Його також потрібно зберегти в безпечному місці, оскільки він дає доступ до змісту зашифрованих файлів.

```python
from cryptography.fernet import Fernet

# Генеруємо ключ
key = Fernet.generate_key()

# Зберігаємо ключ у файл
with open('key.key', 'wb') as key_file:
    key_file.write(key)

# Завантажуємо ключ з файлу
with open('key.key', 'rb') as key_file:
    key = key_file.read()

# Шифруємо файл
with open('plaintext.txt', 'rb') as plaintext_file:
    plaintext = plaintext_file.read()

fernet = Fernet(key)
encrypted = fernet.encrypt(plaintext)

# Зберігаємо зашифрований файл
with open('encrypted.bin', 'wb') as encrypted_file:
    encrypted_file.write(encrypted)
```

Рис. 3. Зашифрування файлу за допомогою Fernet

```python
from cryptography.fernet import Fernet

# Завантажуємо ключ з файлу
with open('key.key', 'rb') as key_file:
    key = key_file.read()

# Розшифровуємо файл
with open('encrypted.bin', 'rb') as encrypted_file:
    encrypted = encrypted_file.read()

fernet = Fernet(key)
decrypted = fernet.decrypt(encrypted)

# Зберігаємо розшифрований файл
with open('decrypted.txt', 'wb') as decrypted_file:
    decrypted_file.write(decrypted)
```

Рис. 4. Розшифрування файлу за допомогою Fernet

Отже, основними перевагами Fernet є безпечний механізм генерації ключів, надійний алгоритм шифрування AES-128 із використанням випадкового вектору ініціалізації, додавання мітки часу до повідомлення, його перетворення за допомогою HMAC і SHA-256 для виявлення будь-яких спроб зміни.

Загалом Fernet є корисним та ефективним інструментом, який допомагає розробникам Python захистити дані невеликих обсягів від несанкціонованого доступу, а також дозволяє перевірити цілісність цих даних.

Список літератури

1. Fernet (symmetric encryption) [Електронний ресурс] – Режим доступу: https://cryptography.io/en/latest/fernet/

2. A Deep Dive Into Fernet Module in Python [Електронний ресурс] – Режим доступу: https://pythonistaplanet.com/fernet/

СЕКЦІЯ 2. ПРОГРАМУВАННЯ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

УДК 621.39 (045)

Аль-Мудхафар Акіл Абдулхуссейн М.[1], Т.В. Смірнова[2], К.О. Буравченко[2], О.А. Смірнов[2]
almudhaffar2004@gmail.com, sm.tetyana@gmail.com, buravchenkok@gmail.com, dr.SmirnovOA@gmail.com
[1]Національний авіаційний університет, м. Київ
[2]Центральноукраїнський національний технічний університет,м. Кропивницький

МЕТОД ОЦІНКИ ТА ПІДВИЩЕННЯ КОРИСТУВАЛЬНИЦЬКОГО ДОСВІДУ АБОНЕНТІВ В SDN НА ОСНОВІ ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ

Еволюційні процеси, які в першу чергу торкнулися комп'ютерних технологій, призвели до появи кількох типів обчислювальних мереж, що представляють сукупність комп'ютерних пристроїв, об'єднаних в одну систему. Основним призначенням такої системи є доступ користувачів до спільних ресурсів та можливість обміну даними між абонентами у процесі роботи.

Сучасний стан та тенденції розвитку комп'ютерних мереж показали, що потенціал зростання продуктивності та пропускної спроможності мереж на основі традиційних технологій практично вичерпаний. В даний час телекомунікаційні мережі будуються на базі безлічі комутаційних пристроїв, кожен з яких може працювати незалежно від інших пристроїв, самостійно визначаючи правила, за якими воно оброблятиме пакети, що надходять на нього. Також сучасні комутаційні пристрої виробляють свої правила комутації пакетів за закладеними у них алгоритмами. Правила обміну службовою інформацією між пристроями та способи її застосування для вироблення узгоджених правил обробки пакетів визначаються безліччю різноманітних службових протоколів, кожен із яких вирішує деяку підмножину завдань адміністрування мережі. Використання безлічі службових протоколів аж ніяк не робить мережне адміністрування простим. Вирішення цієї проблеми значно спростилося разом із порівняно недавнім появою концепції Програмно конфігурованих мереж (SDN). Такі мережі дозволять прискорити маршрутизацію, підвищити зручність конфігурування, віртуалізації, налаштування якості обслуговування мереж зв'язку.

Головними завданнями SDN є відокремлення функцій передачі трафіку від функцій управління (включаючи контроль як самого трафіку, так і пристроїв, що здійснюють його передачу). Це відбувається за рахунок створення спеціального програмного забезпечення, яке може працювати на окремому сервері (комп'ютері) та яке знаходиться під контролем адміністратора мережі. Всі маршрутизатори та комутатори об'єднуються під керуванням контролера SDN або Мережевої Операційної Системи (МОС), яка забезпечує додаткам доступ до управління мережею та постійно відстежує конфігурацію засобів мережі.

Хоча SDN є відповідним рішенням для IT та хмарних провайдерів та підприємств, SDN стикається з деякими проблемами, які перешкоджають його продуктивності та впровадженню. Однією із таких проблем є відсутність дієвого механізму управління користувальницьким досвідом кінцевих користувачів. Таким чином, **метою даної роботи** є розробка методу оцінки та підвищення користувальницького досвіду абонентів мереж SDN на основі використання машинного навчання. Для досягнення поставленої мети було вирішено наступні задачі:

1. Проведено аналіз існуючих механізмів керування користувальницьким досвідом абонентів.
2. Розроблено узагальнену модель оцінки та підвищення користувальницького досвіду абонентів мереж SDN.
3. Проведено аналіз регресійних моделей на можливість їх використання для встановлення взаємозв'язку між параметрами мережі та користувальницьким досвідом.
4. Розроблено алгоритм роботи методу.

Розглянемо архітектуру програмно-конфігурованих мереж. В них функція комутатора з управління переносяться на окремий центральний пристрій – контролер SDN. Такий підхід дозволяє керувати та контролювати стан мережі на логічно централізованому контролері. З іншого боку, з'являється можливість рівню управління відокремитися від фізичної складової, використовуючи логічне уявлення мережі загалом. Взаємодія між рівнем передачі здійснюється за допомогою єдиного уніфікованого відкритого інтерфейсу. В архітектурі SDN виділяють 3 рівні:

– рівень інфраструктури мережі, що представляє набір мережевих структур (комутатори та канали зв'язку);

– рівень управління, що складається з операційної системи, що забезпечує додаткам мережеві сервіси та програмний інтерфейс для управління мережевими пристроями та мережею;

– рівень додатків.

Одне з центральних місць в архітектурі SDN займає мережева операційна система, яка є операційною системою, що забезпечує обробку, зберігання та передачу даних в інформаційній мережі. Мережева операційна

система або контролер SDN визначає взаємопов'язану групу протоколів верхніх рівнів, що забезпечують основні функції мережі: адресацію об'єктів, функціонування служб, безпеку даних, керування мережею. Контролер SDN формує дані про стан всіх ресурсів мережі та забезпечує доступ до них для програм управління мережею. Ці програми керують різними аспектами функціонування мережі, включаючи побудову топології, прийняття маршрутизуючих рішень, балансування навантаження. За допомогою цього протоколи фахівці самі можуть визначати та контролювати, які вузли, за яких умов та з якістю можуть взаємодіяти в мережі. Таким чином, на контролер ще можуть бути покладені задачі по оцінці та підвищенні користувальницького досвіду абонентів, що обслуговуються мережею.

В останні роки технічна спільнота переключила певну увагу з одного пов'язаного показника, якості обслуговування (QoS), на більш орієнтовану на споживача метрику, якість досвіду (QoE). У той час як QoS стоїть між мережею та додатком, QoE зосереджено на абоненті. Зокрема, QoE фокусується на людині як користувачеві, який взаємодіє з додатком, і людині як клієнту, який має справу з постачальником послуг.

Різниця між QoE і QoS підкреслена нижче:

QoS – якість обслуговування:

– характеристики/поведінка мережі;

– гарантії продуктивності, надані провайдером мережі на основі вимірювань;

QoE – якість досвіду:

– вплив поведінки мережі на кінцевого користувача;

– деякі недоліки можуть залишитися непоміченими;

– деякі недоліки можуть зробити додаток марним;

– не фіксується мережевими вимірюваннями.

– QoE враховує очікування користувача, QoS є більш раціональним на основі технічних вимірювань).

На підставі вищезазначеного пропонується модель аналізу оцінки загального QoE за допомогою показників QoS, які можна оцінити більш об'єктивним способом: $\left\{\bigcup_{i=1}^{n} S_i\right\} = \{S_1, S_2, ..., S_n\}$, де: $S_i \subseteq S, \left(i = \overline{1,n}\right) n$ – кількість послуг; $S_i = \left\{\bigcup_{j=1}^{m_i} S_{ij}\right\} = \left\{S_{i1}, S_{i2}, \ldots S_{jm_i}\right\}$, із $S_{ij}\left(j = \overline{1,m_i}\right)$ – підмножина елементів системи забезпечення якості.

Підмножини QoE метрики $S_{ij} \subseteq S_i$ можуть бути представлені як: $S_{ij} = \left\{\bigcup_{p=1}^{r_{ij}} S_{ijp}\right\} = \left\{S_{ij1}, S_{ij2}, \ldots, S_{ijr_j}\right\}$, де: $S_{ijp}\left(p = \overline{1,r_{ij}}\right)$ показники QoE, що характеризують – QoE для S_{ij}; r_{ij} – кількість таких показників.

На другому етапі вибирають показники QoS і QoE, використовуючи багатофакторний кореляційно-регресійний аналіз. Для побудови багатофакторної регресійної моделі необхідно виконати наступні кроки:

1. Вибираються всі можливі фактори QoS, що впливають на досліджуваний показник QoE (або процес). Для кожного фактора визначаються його чисельні характеристики.

2. Вибирається форма регресійної або багатомірної моделі, тобто знаходження аналітичного виразу, що найбільш повно відображає зв'язок факторних характеристик з рівнодійною, тобто відбувається вибір відповідних функцій.

Висновки. Мережі SDN вже давно стали основою побудови телекомунікаційних мереж операторського класу. Проте, в них є певна кількість недоліків, які необхідно усунути. Відповідно, метою даної статті є розробка моделі та відповідного методу оцінки якості користувальницького досвіду абонентів мереж SDN. Таким чином, вперше було розроблено метод оцінки та підвищення користувальницького досвіду абонентів мереж SDN на основі використання машинного навчання, що полягає у послідовному проведенні автоматизованого опитування користувачів, вимірюванні показників якості обслуговування абонентів, виборі та побудові регресійної моделі із множини визначених моделей та керування користувальницьким досвідом за виміряними параметрами якості обслуговування абонентів мережі SDN, що на відміну від відомих, надає змогу підвищувати якість користувальницького досвіду у режимі реального часу. Даний метод дозоляє будувати точні моделі взаємозв'язку параметрів QoE та QoS та підвищує на величину до 10% якість користувальницького досвіду абонентів мереж SDN.

УДК 004.031 42

Т.Х. Фаталієв
tfataliyev@gmail.com
Інститут інформаційних технологій, м. Баку, Азербайджан

ВПЛИВ ВИКЛИКІВ ІНДУСТРІЇ 4.0 НА ІНТЕГРАЦІЮ Е-НАУКИ ТА Е-ОСВІТИ

Сучасні теоретико-методологічні переваги наукових досліджень визначаються можливостями їх застосування та зв'язком із освітою. Будучи виробником нових знань, наука та освіта взаємно об'єднуються, і проблема їх інтеграції є актуальною і постійно досліджується світовими вченими. В останні десятиліття відбулося і продовжується бурхливими темпами зростати безпрецедентний розвиток цифрових інформаційних, комунікаційних та інтелектуальних технологій та систем, які істотно впливають на цей інтеграційний процес. Інтеграція науки та освіти означає спільне використання потенціалу науки та освітніх установ у рамках взаємних інтересів. Інтеграційні процеси охоплюють широкий спектр різних напрямів діяльності та виявляються у найрізноманітніших формах. Прискорилася участь науковців у навчання, а викладачів та студентів у наукових дослідженнях, внаслідок чого підвищується ефективність наукових досліджень, покращується якість освіти та підготовки науково-технічних кадрів, збільшується приплив молоді до наукових досліджень. Таким чином, інтегровані структури науки та освіти забезпечують підготовку якісно нових фахівців, необхідного ринком праці, а технологічні зміни, що базуються на застосуванні нових знань у виробництві, сприяють економічному зростанню.

Відомо, що останнім часом під впливом викликів 4-ї промислової революції (Індустрія 4.0, *Industry 4.0*) у світі розпочався новий етап у галузі побудови Інформаційного суспільства (ІС). Індустрія 4.0 характеризується інтелектуальною автоматизацією, що поєднує фізичний та цифровий світи через Інтернет речей (ІР) та кіберфізичні системи (КФС). Індустрія 4.0 відкриває нові перспективи для інтеграції науки і освіти, що досліджується в даній роботі.

Слід зазначити, що застосування інформаційно-комунікаційних технологій у всіх сферах життя, включаючи науку та освіту (е-науки та е-освіти) передбачено напрямом дії С7 (ІКТ-додатки) Плану дій Світового саміту з ІС (*World Summit on the Information Society - WSIS*). Роботи, що проводяться в цьому напрямі, прискорила успішне впровадження е-науки та е-освіти та відіграє важливу роль у прискоренні сталого розвитку, підвищенні прозорості та підзвітності. Так, окрім інформатизації у цих структурах і послідовно ведеться робота з формування єдиного інформаційного простору, придатного для використання кожного з них, створення інформаційних ресурсів та розвитку інфраструктур. Метою реалізованих робіт є забезпечення спільної діяльності у віртуальному просторі колективів наукових та освітніх установ, а також осіб, які займаються науковими дослідженнями та освітнім процесом, маючи доступ до відповідної інформаційно-комунікаційної інфраструктури до науково-технічних, освітніх інформаційно-обчислювальних ресурсів через високошвидкісну мережу. У той же час можна розглядати е-науку та е-освіту окремо як складні системи з технічними та технологічними компонентами, що складаються з підсистем інфраструктури, генерації даних, збору, зберігання, обробки, пошуку, аналізу, передачі, уявлення тощо.

Можливості, що надаються викликами Індустрії 4.0 та широким застосуванням її передових технологій ІР, КФС, штучного інтелекту (Ш І), хмарних обчислень, аналітики великих даних та інших інтелектуальних рішень, відкривають нові перспективи для розвитку та інтеграції е-науки та е-освіти. З їхньою трансформацією на платформі Індустрії 4.0 з'являються широкі можливості для реструктуризації та інтеграції науки та освіти як корпоративних середовищ у вигляді Науки 4.0 (*Science 4.0*) [1] та Освіта 4.0 (*Education 4.0*) [2]. Таким чином, Науку 4.0 та Освіта 4.0 можна розглядати як еволюцію традиційної науки та освіти, що поєднує інформацію реального та віртуального світу з урахуванням технологічних інструментів нової цифрової доби. Також слід зазначити, що застосування передових технологій Індустрії 4.0 принесе високу продуктивність, більшу гнучкість, кращий контроль та оптимізацію процесів, сталий розвиток та інші переваги у цих структурах.

Мережеві платформи е-науки та е-освіти є основними базами формування інтеграційного процесу. Ці діючі мережеві та обчислювальні е-інфраструктури здатні створювати швидкий зв'язок між структурами, надавати безліч послуг користувачам і водночас інтегруватися з міжнародними науковими та освітніми мережами.

Концептуальна модель зазначеного підходу враховує досліди рішень, сформовані в результаті застосування технологій Індустрії 4.0, а також інтелектуальних додатків, таких як розумні лабораторії, розумні бібліотеки, розумні університети, розумні будинки, розумні міста, цифрові двійники та ін. До концептуальних питань реалізації цього підходу віднести таке:

• Наука та освіта сприймаються як єдине корпоративне середовище,

• Його фізична інфраструктура включає телекомунікаційні мережі, центри обробки даних, будівлі, науково-дослідні та навчальні лабораторії, енерго-тепло-водопостачання, логістику і т.д.

Така інтегрована КФС має забезпечувати таке:

- безперебійне енерго- та водопостачання; клімат контроль; контроль доступу; охорона будівель та відеоспостереження; управління будівлею, виявлення небезпек та попередження тощо;

- обслуговування мережевих ресурсів, об'єктів та обладнання; мережевий моніторинг та кібербезпека; електронні послуги; безперервна діагностика та ін;

- Розробка, управління та безпека інформаційного забезпечення;

- підтримка наукової та освітньої діяльності, управлінських процесів тощо.

Слід зазначити, що об'єкт дослідження є складною системою, що поєднує групи ризиків таких, як співробітники, ресурси, процеси та технології. Для такого середовища забезпечення кіберстійкості, тобто цілеспрямованої роботи системи при впливі атак з кіберпростору є актуальною проблемою, яка потребує превентивних рішень. Під кіберстійкості мається на увазі здатність системи передбачати, витримувати, відновлюватися та адаптуватися до несприятливих умов, навантажень, атак, які використовують чи активуються кіберресурсами. Тут, поряд із спільними питаннями забезпечення комплексної безпеки, важливі й такі [3]:

• Розробка смарт та кіберімунних IT-рішень для забезпечення кіберстійкості;

• Поінформованість та навчання співробітників про кібербезпеку та кіберстійкість;

• Співпраця з вітчизняними та міжнародними організаціями тощо.

До основних рис концепції належать такі:

• Високий рівень взаємодії смарт-об'єктів у своєму складі;

• генерація великих даних та інтеграція смарт-об'єктів з аналітикою великих даних;

• Динамічна реконфігурація смарт-об'єктів для більшої гнучкості;

• Досягнення високої ефективності досліджень, викладацької діяльності та управління за рахунок забезпечення глобального зворотного зв'язку та координації з аналітикою великих даних.

В результаті інтеграції науки та освіти за такого підходу буде досягнуто:

-Прискорення єдності науки та освіти, спільного розвитку як єдиного соціального інституту, укрупнення їх взаємного розвитку;

-Ефективне використання матеріально-технічної бази, інфраструктури та електронних ресурсів;

-Підвищення можливостей ефективної діяльності науки, освіти та управління;

-Розширення можливостей застосування результатів наукових досліджень в освіті, формування людського капіталу високого рівня;

-розширення можливостей підвищення якості та ефективного управління наукою та вищою освітою, а також областями дошкільної та загальної освіти та ін.

У цьому вся інтеграційному процесі Громадянська наука (ГН, *Citizen Science*) також виділяється своїми широкими можливостями. Як новий напрям е-науки ГН грає дедалі важливішу роль у розвитку освіти та розширенні можливостей навчання як добровільного співробітництва громадян, у наукових дослідженнях [4]. З використанням технологій Індустрії 4.0 цей процес розвивається швидкими темпами. Розглянемо як приклад Європейську онлайн-платформу ГН (https://eu-citizen.science/projects). Ця платформа не тільки робить проекти та дані у сфері ДН більш доступними, але також служить простір для спільної роботи, надаючи громадянам, науковцям, вчителям, учням, школам та іншим зацікавленим сторонам корисні інструменти, керівництва, навчання та обмін передовим досвідом кількома мовами та стає полігоном інтеграції науки та освіти. Перспективи інтеграції ГН в освіту та навчання, як і раніше, залежатимуть від застосування глобальних цифрових платформ та рішень Індустрії 4.0.

Реалізація запропонованого підходу є складне завдання, що вимагає нормативного, фінансового, технічного та технологічного забезпечення і тому може бути реалізована поетапно.

Список літератури

1. T.Kh. Fataliyev, Sh.A. Mehdiyev, The impact of Industry 4.0 on the formation of Science 4.0, Problems of Information Technology, vol. 13, no.2, pp. 37-45, 2022

2. E. Vilalta-Perdomo, R. Michel-Villarreal, R. Thierry-Aguilera, Integrating Industry 4.0 in Higher Education Using Challenge-Based Learning: An Intervention in Operations Management. Educ. Sci. 2022, 12, 663. https://doi.org/10.3390/ educsci12100663

3. T.Kh. Fataliyev, N.N. Verdiyeva, Science 4.0: Complex security problems and solution mechanisms. "Cybersecurity for Critical Infrastructure Protection via Reflection of Industrial Control Systems", NATO Science for Peace and Security SeriesD: Information and Communication Security, Amsterdam: IOS Press. (DBLP), Vol. 62, pp.151-153, 2022

4. J. Roche, etc., Citizen Science, Education, and Learning: Challenges and Opportunities, Front. Sociol., vol. 5:613814, 2020. https://doi.org/10.3389/fsoc.2020.613814

УДК 004.032.26

О.В. Тазетдінов[1], В.Г. Бабенко[1]
o.v.tazetdinov.asp22@chdtu.edu.ua, v.babenko@chdtu.edu.ua
[1]*Черкаський державний технологічний університет, м. Черкаси*

МЕТОД НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ АВТОМАТИЗОВАНОГО ПОШУКУ ВРАЗЛИВОСТЕЙ ПРОГРАМНОГО КОДУ

За останні роки кількість виявлених та описаних вразливостей програмного забезпечення стрімко зросла [1]. Більшість класичних методів виявлення вразливостей базуються на правилах та метриках подібності коду, які сформовані на основі досвіду експертів. Загалом, дуже складно описати специфіку вразливостей програмного забезпечення та їх причини, також важко визначити відповідні правила їх виявлення, що були б описані точно та повністю. Нейронні мережі [2, 3] можуть успішно вирішувати подібні задачі, оскільки дозволяють автоматизовано витягувати складні характеристики з вхідних даних, уникаючи проблем високої вартості, нестабільності та неповноти ручної побудови функцій і емпіричного визначення правил.

Для розв'язання вищевказаних проблем, було розроблено метод автоматизованого виявлення вразливостей програмного коду на основі нейронної мережі, що використовує Машини Екстремального Навчання (МЕН). Застосування саме цих нейронних мереж прямого поширення дозволить підвищити ефективність навчання.

Оскільки нейронна мережа може приймати лише вектор як вхідні дані, то символьне представлення вихідного коду потрібно додатково перетворити на векторне представлення.

Якість векторного представлення можна оцінити косинусом подібності між векторами у векторному просторі, яку можна обчислити за формулою:

$$\cos ine(A,B) = \frac{A * B}{\| A \|_2 \| B \|_2} \quad (1),$$

де A і B - це вектори. Чим ближче значення до 1 або −1, тим більше схожі два вектори.

Вихідну функцію МЕН можна позначити як:

$$f(x) = \begin{cases} h(x)H^T(\lambda I + HH^T)^{-1}T, N < L \\ (\lambda I + HH^T)^{-1}H^T h(x)T, N \geq L \end{cases} \quad (2),$$

де H відноситься до псевдообернення Мура-Пенроуза, L відноситься до числа нейронів прихованих шарів, I відноситься до матриці тотожності N, а λ відноситься до фактору регуляризації.

На етапі виявлення вразливостей цільові програми трансформуються в проміжне уявлення так само, як і на етапі навчання. Модель, отримана на етапі навчання, виконується як класифікатор, генеруючи прогнози відповідно до перетворених вхідних даних.

Запропонований метод синтаксичного аналізу застосовується безпосередньо для пошуку всіх чутливих до безпеки пунктів програми без опису вразливості. З виявлених зразків витягуються чутливі до безпеки точки програми. Потім ці фрагменти програми діють як критерії для виконання традиційного статичного аналізу на рівні вихідного коду. А визначена функція (2) може бути використана для оцінки ефективності МЕН.

Список літератури

1. G. Tang, L. Meng, H. Wang et al., "A comparative study of neural network techniques for automatic software vulnerability detection," in Proceedings of the 2020 International Symposium on Teoretical Aspects of Software Engineering (TASE), Hangzhou, China, 2020.

2. J. Viega and J. T. Bloch, "A static vulnerability scanner for C and C++ code," in Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC 2000, vol. 257, New Orleans, LA, USA, 2000.

3. H. Yuan, B. Wang, and L. Niu, "Kernel extreme learning machine for learning from label proportions," Lecture Notes in Computer Science, vol. 400, p. 409, 2018.

УДК 004.4

В.В.Кіш, Н.І.Йовбак
kish.viktor@student.uzhnu.edu.ua, yovbak.nika@student.uzhnu.edu.ua
ДВНЗ »Ужгородський національний університет», м.Ужгород

СИСТЕМА КОНТРОЛЮ ВЕРСІЙ – Git

Git – це розподілена система контролю версій, найбільш популярна на цей час система, яка дозволяє відстежувати історію розробки ПЗ і спільно працювати над складними проєктами з будь-якої точки світу.

Якщо декілька розробників працюють над одним і тим самим додатком, іноді одним і тим самим файлом або навіть у суміжних строках, **Git** дозволяє фіксувати зміни паралельно за допомогою декілька **бранчей (branch),** а потім зливати усі зміні в один бранч, наприклад основну кодову базу. Альтернативою без контролю версій було би збереження змін у кожного розробника локально, а потім вручну змішування змін на девайсі одного із них.

Основними завданнями, які виконують СКВ, є:

▪ зберігання файлів в репозиторії;

▪ підтримка перевірки файлів в репозиторії;

▪ створення різних варіантів одного документа, так званої гілки, із загальною історією змін до точки розгалуження і з різними – після неї;

▪ знаходження конфліктів при зміні вихідного коду і забезпечення синхронізації при роботі в середовищі з багатьма користувачами розробки.

▪ відстеження авторів змін;

▪ надання інформації про те, хто і коли додав або змінив конкретний набір рядків у файлі;

▪ ведення журналу змін, в який користувачі можуть записувати пояснення про те, що і чому вони змінили в цій версії;

▪ контроль прав доступу користувачів, дозволяючи або забороняючи читання або зміну даних, залежно від того, хто запитує цю дію.

Переваги Git перед іншими системами:

Окремою перевагою є можливість порівняти зміни за допомогою **утиліти diff (або git diff),** що дозволяє бачити лише зміни без порівняння усіх файлів.

Головна перевага **Git** полягає в тому, що він надзвичайно швидкий і прозорий, а також корисний для нелінійної розробки та ефективний як для невеликих проєктів, так і для великих систем із тисячами учасників.

Git не є централізованим сервером як файлові сервера, а розподілена система. Замість цього він є ***розподіленим (distributed)***, тобто він може зберігатися одночасно локально та на серверах сервісів таких як *GitHub, BitBucket, GitLab* тощо. Виглядає він як звичайна директорія з файлами, за виключенням наявності директорії **".git",** що зберігає зміни за допомогою різниць, тобто ***diff-ів.*** Тобто, якщо в історії змін є великий файл, то **Git** не зберігає його копії, а тільки різницю між його станом до й після зміни. Ця директорія називається **репозиторій**.

Якщо сервер із віддаленим репозиторієм виходить із ладу, ви можете відновити свій код із локальної копії. Якщо це стосується вашої локальної копії, ви можете завантажити код із серверів за лічені хвилини.

Для того щоб користуватись **git,** необхідно відкрити консоль та скористатись ключовим словом **git "команда".**

Після деяких змін стан файлів фіксується за допомогою **коммітів (commit). Комміт** – це контрольна точка в історії змін, наприклад, комміт включає у себе зміну 3 файлів, створення 2 інших файлів, видалення одного файлу.

Для організації коммітів використовуються **бранчі** та **теги**.

Бранчі - це вказівники на певний комміт, що змінюється при додавання інших. Вони використовуються при розробці версії або певної фічі.

Коли приближається час релизу, розробник створює **тег**, що не відміну від бранча не може бути зміненим. Теги звичайно мають маркування як і версії ПЗ.

Операції

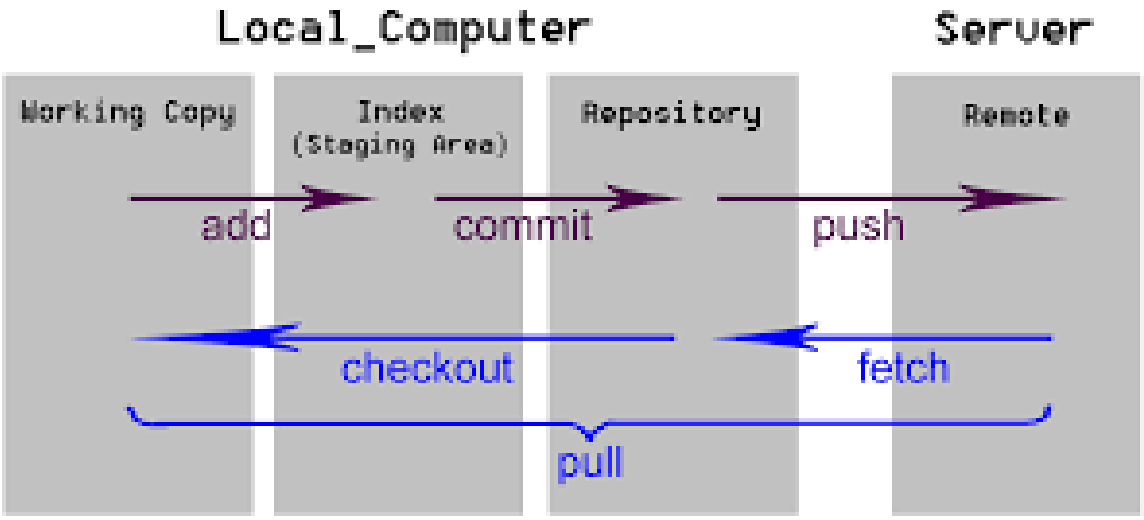
Git дозволяє полегшити життя розробника за допомогою повністю або частково автоматичних операцій:

▪ операції **commit/push/pull/fetch** забезпечують додавання та синхронізацію змін між локальним та віддаленими репозиторіями;

▪ операції **merge/rebase/cherry-pick** надають можливість змішувати зміни між бранчами;

▪ операції **status/log** надають можливість слідкувати за змінами у проекті;

▪ операції **checkout/revert/reset** забезпечують відміну змін та навігацію між коммітами, тегами або бранчами.



Базові команди git:

▪ **git status:**

вказує статус репозиторій

▪ **git init:**

ініціалізує репозиторій у конкретній папці

не рекомендується створювати один репозиторій всередині іншого

▪ **git add…:**

індексує файли, котрі ви вказали

щоб вказати конкретний файл, потрібно записати його назву **git add index.js**

щоб вказати всі файли потрібно написати **git add .**

▪ **git commit:**

добавляє файли та зміни до репозиторію

зберігає ідентифікатор кожного комміту

скориставшись прапорцем **–m,** можна вказати коментарій до комміту **git commit -m "initial commit"**

▪ **git push:**

добавляє ваш репозиторій до віддаленого серверу, щоб легко розробляти додаток разом з командою

потрібно скористатись **git push** посилання_до_репозиторію

також при видаленні репозиторію локально завжди можна буде завантажити його з серверів.

Отже, *однією з найбільш популярних систем контролю версій (СКВ) на сьогодні є* ***Git*** – розподілена система керування версіями файлів, яка на сьогоднішній день є незамінним інструментом для спільної розробки та системою контролю версій та дозволяє вернутись до будь-якого стану розробки проєкту.

Список використаної літератури

1. Пугачов Р.В. Системи контролю версіями: навч.-метод. посібник / Пугачов Р.В., Любченко Н.Ю. Соболь М.О.– Харків : НТУ "ХПІ", 2019. – 130 с.
2. https://git-scm.com/doc
3. https://docs.github.com/en

УДК 004.415.538

В.В. Алексеєнко
work.alekseenko@gmail.com
Житомирський державний університет імені Івана Франка,м. Житомир

АНАЛІЗ ПРОЦЕСУ ТЕСТУВАННЯ МОБІЛЬНИХ ДОДАТКІВ

Розробка якісного та конкурентоспроможного мобільного додатка вимагає зусиль багатьох спеціалістів: дизайнерів, розробників, QA-спеціалістів. Основним фокусом QA-спеціалістів є виведення на ринок продукту, що не має помилок та відповідає потребам користувачів.

Тестування мобільних додатків — це процес перевірки якості розробленого додатка для мобільних пристроїв. Цей процес передбачає виконання різноманітних тестів для виявлення будь-яких потенційних помилок або проблем у додатку, які можуть спричинити незадовільну взаємодію з користувачем. Тестування мобільних додатків може включати перевірку основних функцій додатків, інтерфейсу користувача, сумісності та продуктивності.

Функціональне тестування — це процес перевірки функціональності мобільного додатка, щоб переконатися, що він відповідає потребам користувачів. Цей процес допомагає перевірити точність функцій додатка, те, як він опрацьовує різні вхідні дані та різні умови роботи.

Для проведення функціонального тестування мобільних додатків використовуються різні методи та техніки, наприклад:

- Тестування на основі вимог (означає перевірку того, чи відповідає додаток вимогам, викладеним у специфікації).
- Тестування різноманітних функцій (спосіб перевірити різні аспекти програми, такі як автентифікація, реєстрація, платежі та пошук тощо).
- Тестування на основі тест-кейсів (тестування проводиться, щоб перевірити, чи виконує мобільний додаток те, що для нього було заплановано. Це робиться за допомогою прописаних тест-кейсів).
- Тестування на основі сканування глибини (тестування додатка за допомогою глибинного сканування означає перевірку його в різних ситуаціях та виявлення помилок або проблем, що можуть виникнути при використанні додатка).

Отож, у процесі функціонального тестування мобільних додатків, QA-спеціалісти перевіряють різні функціональні можливості додатка, щоб переконатися, що він виконує всі заплановані завдання та задовольняє потреби користувачів. Функціональне тестування мобільних додатків може бути проведене вручну, або з використанням автоматизованих тестів.

Тестування інтерфейсу користувача — це вид функціонального тестування, який перевіряє, наскільки добре додаток взаємодіє з користувачем й відповідає на його запити. Тестування інтерфейсу користувача передбачає перевірку того, чи різні елементи інтерфейсу, такі як кнопки, текстові поля, меню та інші, працюють належним чином і забезпечують правильну взаємодію з користувачем.

У процесі тестування інтерфейсу користувача перевіряються такі аспекти, як:

- Відповідність елементів інтерфейсу дизайну та очікуванням користувачів.
- Коректність та швидкість відповіді на взаємодію користувача зі спливаючими вікнами, повідомленнями та іншими елементами, що з'являються в процесі роботи з додатком.
- Перевірка коректності введення даних в текстові поля та їх збереження в базі даних.
- Перевірка коректності роботи меню та інших навігаційних елементів додатка.
- Перевірка наявності та коректності роботи інтерактивних елементів, таких як кнопки, чекбокси, радіокнопки тощо.

Після проведення тестування інтерфейсу користувача, QA-спеціалісти повинні підготувати звіт з результатами тестування, де будуть відображені проблеми, виявлені в процесі тестування, їх причини та рекомендації щодо вирішення цих проблем. Звіт допоможе розробникам додатка виправити помилки та забезпечити якість роботи додатка.

Тестування сумісності мобільних додатків — це процес перевірки роботи додатка на різних мобільних пристроях та платформах. Тестування сумісності дозволяє перевірити роботу додатка на різних моделях та версіях мобільних пристроїв, а також на різних операційних системах.

У процесі тестування сумісності перевіряють:

- роботу додатка на різних версіях ОС (операційних систем) та їх патчах;
- роботу додатка з різним розширенням екрана та їх орієнтацією;
- роботу додатка на різних моделях мобільних пристроїв, зокрема на смартфонах та планшетах;
- роботу додатка на різних мережах, тобто з використанням Wi-Fi та мобільної мережі;
- роботу додатка з іншими програмами на мобільному пристрої, наприклад, з іншими додатками, що встановлені на пристрої.

Для тестування сумісності мобільних додатків використовують різні інструменти та сервіси. Наприклад, для тестування на різних моделях пристроїв використовують фізичні пристрої або емулятори, такі як Android Emulator для тестування на Android-пристроях та iOS Simulator для тестування на iOS-пристроях.

Тестування продуктивності мобільних додатків (Performance Testing) допомагає перевірити, наскільки швидко та ефективно додаток працює під великим навантаженням або при використанні ресурсів пристрою. Виокремлюють, також, декілька видів тестування продуктивності:

- Тестування витривалості (Endurance Testing) — цей тип тестування включає роботу додатка протягом тривалого часу, щоб побачити, чи вона стабільна та придатна для використання в умовах тривалого використання.
- Тестування швидкості реакції додатку на дії користувача (Responsiveness Testing), яке дозволяє визначити час від реакції додатка на дії користувача до відображення результатів на екрані.
- Тестування навантаження (Load Testing) — цей тип тестування дозволяє побачити, скільки людей можуть використовувати додаток одночасно без втрати продуктивності та якості роботи.

Також для тестування продуктивності можуть використовуватись інші інструменти, які дозволяють вимірювати швидкість завантаження додатка, споживання ресурсів пристрою (батареї, пам'яті, процесора тощо) та інші параметри продуктивності.

Окрім цього, важливим аспектом є тестування безпеки мобільних додатків. Тестування безпеки має на меті забезпечити, що додаток захищений від різних видів атак та зломів, таких як витік даних, підробка даних, хакерські атаки та інші. Це включає перевірку наявності в додатку різних типів аутентифікації, шифрування даних, захисту від зломів та іншого.

Для проведення тестування мобільних додатків часто використовуються різні інструменти для автоматизації тестування, такі як:

- Appium — це відкрите програмне забезпечення для тестування мобільних додатків, що дозволяє автоматизувати тестування для додатків на різних платформах, включаючи iOS та Android.
- Selendroid — це інструмент для автоматичного тестування мобільних додатків, який використовує Selenium для тестування додатків на платформі Android.
- Calabash — це інструмент для автоматичного тестування мобільних додатків, який підтримує додатки для платформ iOS та Android.
- Robotium — це інструмент для автоматичного тестування додатків для Android, який дозволяє автоматизувати різні види тестування, такі як тестування одиниць, функціональне тестування та тестування користувацьких сценаріїв.
- Espresso — це інструмент для автоматичного тестування додатків для платформи Android, який забезпечує швидке тестування великих додатків та дозволяє тестувати взаємодію з користувачем.
- XCUITest — це інструмент для автоматичного тестування додатків для платформи iOS, який дозволяє тестувати додатки на декількох пристроях одночасно та забезпечує швидке виконання тестів.
- TestComplete Mobile — це інструмент для автоматичного тестування мобільних додатків, який підтримує додатки для платформ iOS та Android та дозволяє автоматизувати різні види тестування.
- MonkeyTalk — це інструмент для автоматичного тестування мобільних додатків, який підтримує додатки для платформ iOS та Android та дозволяє автоматизувати тестування користувацьких сценаріїв.

Усі ці аспекти тестування мобільних додатків покликані, щоб забезпечити якість та безпеку додатка для користувачів. Правильно проведене тестування може допомогти уникнути проблем та помилок в додатку, які можуть завдати шкоди користувачам або спричинити негативний вплив на репутацію розробника додатка.

Список літератури

1. Kotenko, N., Zhyrova, T., & Kuleba, M. Дослідження особливостей тестування мобільних додатків. Управління розвитком складних систем. 2020. Vol. 41. P. 55–60. URL: https://doi.org/10.32347/2412-9933.2020.41.55-60

2. Жирова Т.О. Проблеми тестування інтерфейсу Web-додатків. IX Міжнародна конференція молодих вчених «Молоді вчені 2018 - від теорії до практики»: збірник матеріалів. Дніпро-Варна:«Дике Поле». 2018. С. 184-188.

3. Мелкозерова О. Аналіз інструментів для автоматизованого тестування програмного забезпечення. Комп'ютерні науки та кібербезпека. Міжнародний електронний науково-теоретичний журнал. 2019. №1. С. 75 – 84.

UDC 004.056.53

V. I. Bakalo
Vlad_B212@i.ua
National Aviation University, Kyiv

PHISHING AS THE MOST COMMON CYBER THREAT

Scientific and technical progress and the development of information technologies increase the volume of information. It is an important resource for the state, commercial structures, and private individuals. Therefore, its protection is very important and relevant during martial law. The war in Ukraine carries not only economic and humanitarian threats but also informational threats.

After the start of the full-scale Russian invasion of Ukraine, the number of cyber fraud cases increased significantly, as was reported at a joint press briefing of the National Security Council, the National Bank of Ukraine, and Kyivstar, held at the National Bank on February 15, 2023. A significant part of them is phishing attacks, as evidenced by the data shown in the diagram.

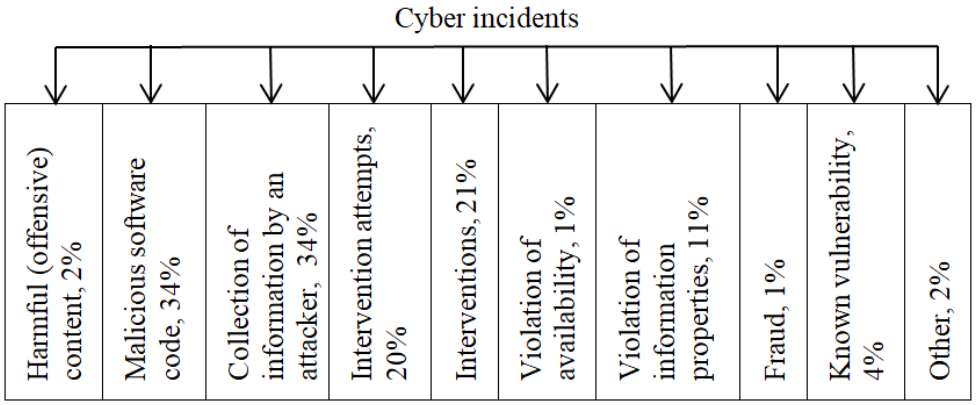


Figure 1 - Cyber incidents

The hostilities in the country brought not only devastation but also shook the psyche of the people. Unstable emotional behavior and stressful situations became favorable backgrounds for various types of deception.

Mass mailing of various types of information is also a threat because it can contain not only computer viruses but messages spreading fakes and negativity.

After almost 36 years since it was first mentioned and almost 30 years since the first primitive forms of chat attacks appeared, phishing is still one of the biggest threats on the Internet.

Like any technology, phishing develops and improves from a simple imitation of a trusted service to attacks on critical infrastructure (Colonial Pipeline, USA).

In the 1990s, the term "phishing" was coined to refer to the use of fraudulent e-mails to "fish" information from unsuspecting users. AOL Inc. (an American media company, provider of online services, and electronic bulletin boards) was a popular Internet content system; attackers used phishing and instant messaging to impersonate AOL employees to trick users into revealing their credentials to steal accounts.

Phishing emails began to be used in the 2000s to trick users into revealing their bank account credentials. The emails contained a link to a malicious site that duplicated an official bank site, but the domain was a slight variation of the official domain name (for example, paypai.com instead of paypal.com). Attackers later targeted other accounts, such as eBay and Google, to steal credentials, steal money, commit fraud, or spam other users.

The first phishing lawsuit was filed in 2004 against a California teenager who created an imitation of America Online. With this fake website, he was able to obtain sensitive information from users and access credit card details to withdraw money from their accounts.

Social engineering as a method of manipulating the human psyche helps cyber fraudsters in their illegal actions. Phishing letters are an effective, cheap, free tool for quickly gaining access to various types of data.

Pushing to spontaneous actions before the threat of account blocking, loss of money, or customers, people act recklessly and do not recognize unreasonable demands, or suspicious requests. They recognize the deception only after they stop worrying, and carefully consider everything - they see the result of the fraud.

By using a seemingly innocent email, cybercriminals can gain a small foothold and build on it.

Acting according to simple logic, fraudsters gain access to personal data. Using the simple steps shown in the diagram, they receive the financial resources of a trusting user.

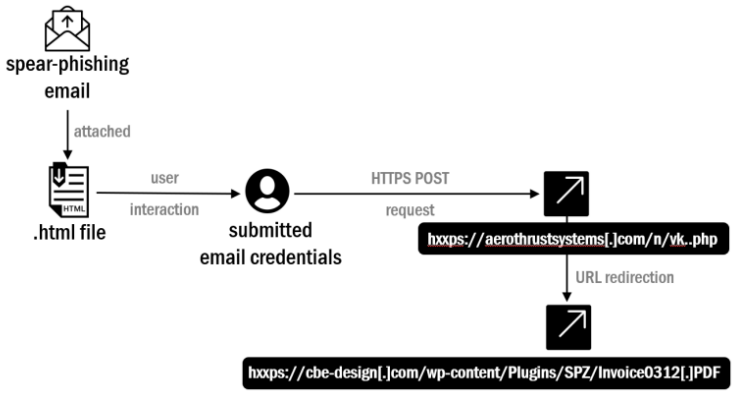


Figure 2 - The generalized attack chain scheme

Using human weaknesses, phishing is developing and improving, and today the following types of phishing are known:

• deceptive phishing (e-mail links that redirect users to erroneous data collection websites, such as login credentials);

• website spoofing (javascript commands are used to change the URL they point to. This can happen if you open a fake website address instead of a legitimate one);

• bypassing the filter (images are used instead of text, so it is difficult for anti-phishing filters to detect them);

• voice phishing (calls from banks with the requirement to provide confidential data for theft of money, shopping);

• SMS-phishing (a message that is deceptive) "you won a prize, click ... to apply");

• InSession Phishing (opens a pop-up window that misleads the user);

• mobile phishing (encouraging mobile subscribers to transfer money from their account to a fraudster's account);

• farming (changing the Domain Name System of the web page visited by the user to a specially created one).

• spoofing (the tactic of pretending to be someone in order to gain access to confidential data or bank accounts).

Financial gain is the main goal of most phishing attacks. With specific targets that include credit card data and large sums of money, their efforts are aimed at e-commerce, social media, financial institutions, payment systems, IT companies, telecommunications companies, and delivery companies.

Phishing is constantly evolving to bypass security and human detection, so you need to constantly learn to recognize the latest phishing strategies. It only takes one person to be phished to trigger a major data breach. This is why it is one of the most critical threats to mitigate, and the most difficult because it requires human protection.

Hundreds of thousands of Ukrainians become victims of cyber fraud, losing tens of millions of hryvnias every day. Ukrainian banks block such funds in the accounts of criminals, and during the investigation of crimes, the money is returned. However, it is not always possible to block the stolen funds in time, because they are transferred to territory that is not under the control of law enforcement agencies of Ukraine.

Protection against phishing is an important component of the safe operation of any structure and the financial security of a person. Therefore, simple actions will help you avoid trouble:

- using spam filters, we protect ourselves from it. Filters analyze the message's origin, the software used to send it, and its appearance to determine whether it is spam. But we must take into account that spam filters are not always 100% accurate (they can block e-mails from legitimate sources);

- changing your browser settings can prevent you from opening fraudulent websites because they keep a list of fake websites and when you access such websites, the address is blocked or a warning message appears. These settings help to open only trusted websites;

- banking and financial institutions use monitoring systems to prevent phishing;

- remember that secure websites with a valid Secure Socket Layer (SSL) certificate start with "https".

References

1. List of categories of cyber incidents. URL: https://www.cip.gov.ua/ua/news/perelik-kategorii-kiberincidentiv.

2. O. Trofymenko, Yu. Prokop, N. Loginova, O. Zadereyko. Cybersecurity of Ukraine: analysis of the current state. Protection of information. 2019. (vol. 21, № 3).

3. Dovgan O.D., Doronin I.M. Escalation of cyber threats to the national interests of Ukraine and legal aspects of cyber protection: monograph. Kyiv: «ArtEk» Publishing House, 2017.

УДК 004.4

Б.Ю. Вінтенко[1], О.А. Смірнов[2], О.В. Коваленко[2], С.А. Смірнов[2]
boris.vintenko@gmail.com, dr.SmirnovOA@gmail.com, dr.kovalenkoov@gmail.com, smirnov.ser.81@gmail.com
[1]ПАТ "Науково-виробниче підприємство "Радій", м. Кропивницький
[2]Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ НОРМАТИВНОЇ ДОКУМЕНТАЦІЇ ТА СТАНДАРТІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ УПРАВЛІННЯ АЕС, ВАЖЛИВИХ ДЛЯ БЕЗПЕКИ

На сьогоднішній день одним з основних джерел електричної енергії є атомні електростанції (АЕС). В багатьох країнах світу, зокрема в Словаччині, Франції, Україні обсяги атомної генерації становлять понад 50%. Використання атомної енергії є екологічно чистим та економічно вигідним, проте має високі вимоги до безпеки та надійності для забезпечення захисту людини та навколишнього середовища від радіаційного випромінювання ядерного палива. В сучасному світі для управління технологічними процесами та забезпечення безпеки на АЕС використовуються цифрові системи управління з широким застосуванням комп'ютерів та контролерів. Від якості та надійності програмного забезпечення (ПЗ) таких систем залежить безпечна експлуатація АЕС.

Цифрові системи управління потребують особливих підходів до розробки та оцінювання надійності в порівнянні з аналоговими системами, оскільки в силу свого характеру та призначення допускають більшу свободу під час проектування. Ці підходи мають відповідати визначеним вимогам. Вимоги до підходів проектування, реалізації, верифікації, валідації та інших етапів життєвого циклу таких систем описуються в міжнародних стандартах та публікаціях МАГАТЕ й у галузевих нормативних документах різних країн. Ці документи є обов'язковими для використання всіма учасниками процесів проектування, розробки, впровадження та експлуатації комп'ютерних систем управління (КСУ)

Метою роботи є визначення стандартів, що можуть використовуватися розробниками програмного забезпечення для комп'ютерних систем управління АЕС, важливих для безпеки.

Об'єктом дослідження є процес розробки програмного забезпечення для комп'ютерних систем управління АЕС.

Предметом є дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення для комп'ютерних систем управління АЕС.

У результаті дослідження визначені критерії безпеки комп'ютерних систем управління, досліджено стандарти IEC 61508 та IEC 61513, досліджена настанова з безпеки МАГАТЕ SSG-39, визначені категорії функцій управління та класи безпеки систем АЕС, досліджені стандарти, що містять вимоги до програмного забезпечення, інтерфейсів користувача та кодування MISRA C/C++.

Комп'ютерні системи управління (КСУ) згідно з стандартом IEC 61508 є окремим випадком систем контролю та управління (СКУ). СКУ – це система, що приймає вхідні сигнали від процесу (датчиків) та/або оператора, обробляє вхідні параметри згідно з необхідними технологічними алгоритмами та генерує вихідні сигнали, зумовлюючи роботу керованого об'єкту необхідним чином. СКУ, що базуються на КСУ все в більших об'ємах використовуються для виконання функцій безпеки на об'єктах, які можуть створювати ризики для людей та навколишнього середовища (атомні електростанції, газова та нафтова промисловість, транспорт, хімічне виробництво). СКУ на таких об'єктах мають відповідати певним характеристикам безпеки (резервування, самодіагностика, стійкість до відмов). КСУ мають високий рівень програмної складової, для якої існують, зокрема, вимоги інформаційної безпеки (Security Requirements). Згідно з ними, дані та програмне забезпечення мають бути захищеними від несанкціонованого доступу та несанкціонованих змін. Інформаційна безпека описана в групі стандарті ISO/IEC 27000, нормативному документі NIST SP 800-82 «Guide to Industrial Control Systems (ICS) Security». Разом з тим, для КСУ важливими є вимоги функціональної безпеки (Safety Requirements). Функціональна безпека системи забезпечує коректність виконання функцій управління об'єктом, у випадку відмови системи – переведення об'єкту керування в безпечний стан. Її вимоги стосуються не тільки програмної складової, а і апаратного забезпечення, електричного та електронного обладнання та механічних пристроїв, а також власне інформаційної безпеки.

Загальні стандарти функціональної безпеки для КСУ наведено в міжнародних стандартах, таких як IEC 61508. Стандарт IEC 61508 « Functional safety of electrical/electronic/programmable electronic safety-related systems» є стандартом верхнього рівня для функціональної безпеки (ФБ) КСУ. Виконання вимог цього стандарту підвищує надійність та життєздатність продукту, а також формалізує та систематизує процеси проектування, розробки, тестування та впровадження.

Дослідження стандарту IEC 61508 дозволяє зробити висновок, що він містить важливі вимоги до всіх етапів життєвого циклу програмного забезпечення КСУ. Згідно стандарту IEC 61508 застосування багатьох підходів, залежить від конкретних практик, засобів та мов програмування, що використовуються, але на більшості рівнів повноти безпеки (SIL) можна виділити загальні підходи до розробки ПЗ

Вимоги стандарту IEC 61508 не є спеціалізованими виключно для атомної галузі. Для різних областей діяльності вимоги стандарту деталізовані в інших стандартах, наприклад IEC 61511 – для небезпечних виробництв, IEC 62061 – для машинобудування. Стандарт IEC 61513 «Nuclear power plants – Instrumentation and control important to safety – General requirements for systems» є головним в ієрархії стандартів для КСУ, важливих для безпеки атомних станцій. Він встановлює загальні вимоги для архітектури та обладнання КСУ АЕС, важливих для безпеки, що створюються на основі як аналогових, так і цифрових технологій. В багатьох країнах вимагається сертифікація КСУ атомних станцій згідно з стандартами IEC 61508/IEC 61513.

Аналіз вимог стандарту IEC 61513 дозволяє зробити висновок, що він не містить прямих вимог до особливостей розробки програмного забезпечення. Проте він визначає процеси, в яких беруть участь розробники програмного забезпечення. Також він вказує на стандарт, що визначає класифікацію КСУ АЕС з функціональної безпеки, а також на стандарти, що містять конкретні вимоги до програмного забезпечення в залежності від класу функціональної безпеки.

Настанова з безпеки МАГАТЕ SSG-39 «Design of Instrumentation and Control Systems for Nuclear Power Plants, Specific Safety Guide No. 39» є публікацією, що спирається на стандарт IEC 61513 та відповідає його вимогам. Цей документ також містить рекомендації щодо всіх етапів розробки КСУ АЕС, важливих для безпеки.

В результаті дослідження публікації МАГАТЕ SSG-39 можна зробити висновок, що цей документ містить важливі вимоги до менеджменту розробки, проектування, апаратного та програмного забезпечення КСУ. Він містить посилання на інші відповідні міжнародні стандарти, що дає можливість проектантам КСУ підібрати необхідні для вивчення документи.

Функції систем контролю та управління АЕС класифікуються в категорії, що означають важливість функції для безпеки. Важливість функції для безпеки оцінюється наслідками її відмови тоді, коли необхідне її виконання, та наслідками хибного спрацювання.

Нормативний документ України НП 306.2.202-2015 «Вимоги з ядерної та радіаційної безпеки до інформаційних та керуючих систем, важливих для безпеки атомних станцій» та стандарт державного підприємства «НАЕК «Енергоатом» СОУ НАЕК 100:2022 «Інформаційні та керуючі системи, важливі для безпеки атомних електричних станцій. Загальні технічні вимоги» аналогічно до стандарту IEC 61226 поділяють функції КСУ класів безпеки 2 та 3 за НП 306.2.141-2008 на категорії «А», «В» та «С» залежно від їх ролі в забезпеченні та підтримці безпеки, а також можливим невиконанням або помилковим виконанням своїх функцій.

Висновки. З метою створення високонадійного та якісного програмного забезпечення комп'ютерних систем керування АЕС, важливих для безпеки, в світі існують стандарти проектування та галузеві нормативні документи. Результатом дослідження стандартів та галузевих нормативних документів, які мають відношення до розробки ПЗ КСУ АЕС, важливих для безпеки є наступне:

– міжнародні стандарти та нормативні документи мають різні рівні спеціалізації. В документах вищого рівня наводяться загальні вимоги до всіх етапів життєвого циклу КСУ та містяться посилання на документи наступного рівня. Документи наступного рівня є більш спеціалізованими і мають відношення до більш вузьких сфер діяльності;

– кожна країна світу може визначати свій набір стандартів та нормативних документів, який вимагається використовувати під час проектування та реалізації КСУ для АЕС;

– в нормативних документах містяться вимоги та рекомендації як щодо організації процесу розробки, так і загальні рекомендації та приклади написання коду програм різними мовами програмування;

– через те, що вимоги розглянутих стандартів мають бути актуальними протягом тривалого часу, ці стандарти не містять формалізованих методик написання коду програм;

– не існує формалізованих методик сертифікації коду на відповідність вимог розглянутих стандартів;

– кожна організація, яка займається проектуванням, розробкою та верифікацією програмного забезпечення КСУ для АЕС, повинна розробити власні методики з застосування вимог та оцінки критеріїв виконання вимог стандартів та нормативів.

Таким чином, виходячи з вишеперерахованого, перспективами подальших досліджень є детальний аналіз вимог міжнародних стандартів та галузевих нормативних документів до програмного забезпечення КСУ АЕС та розробка методики його написання та оцінки відповідності вимогам стандартів.

УДК 004.021

Є.О. Волошин, ст. 2 курсу
voloshynieo@gmail.com
Науковий керівник: В.С. Гермак. викладач
Центральноукраїнський національний технічний університет, м. Кропивницький

ОГЛЯД МЕТОДІВ ТЕСТУВАННЯ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

В наш час велике значення мають генератори псевдовипадкових послідовностей, які широко застосовуються у різних сферах: моделювання, чисельний аналіз, програмування, криптографія. Тому задача створення якісних генераторів та ефективних методів їх оцінки становить великий інтерес. До криптографічних генераторів застосовуються особливі вимоги: хороші статистичні властивості, великий період послідовності, неможливість передбачити попередні значення генератора маючи згенерований фрагмент, генеровані послідовності чисел повинні бути незалежні. Для перевірки якості генераторів застосовують різні тести. Наразі немає жодного набору критеріїв, який би оцінював, наскільки дані послідовності застосовні саме для конкретної галузі. Існують різні тести, які оцінюють, наскільки досліджувана послідовність «схожа» чи «не схожа» на справді випадкову. Можна виділити такі основні групи методів, які використовуються для оцінки якості генераторів псевдовипадкових послідовностей:

1) Графічні методи. Характеристики послідовностей зображуються за допомогою графіків та діаграм. Сюди відносяться гістограма розподілу елементів, розподіл на площині, перевірка на монотонність. Гістограма дозволяє оцінити рівномірність розподілу чисел у послідовності та визначити частоту повторення кожного символу; розподіл на площині застосовується для визначення залежності між елементами послідовності; перевірка на монотонність дозволяє визначити рівномірність, виходячи з аналізу незростаючих та неспадних підпослідовностей; автокореляційна функція використовується для оцінки кореляції між зсунутими копіями послідовностей та окремих підпослідовностей.

2) Статистичні методи. Статистичні властивості згенерованої послідовності визначаються числовими характеристиками. На основі оціночних критеріїв робляться висновки про ступінь близькості до випадкової послідовності. На відміну від графічних тестів, де результати інтерпретуються людиною, що привносить суб'єктивність у трактування результатів, статистичні тести видають чисельну характеристику, яка дозволяє зробити однозначний висновок, пройдено тест чи ні. Для тестування послідовностей на випадковість існує велика кількість алгоритмів, серед них найбільш поширені тести NIST STS, DIEHARD, CRYPT-X, тести Д. Кнута та інші.

Один із перших наборів статистичних тестів було запропоновано Д. Кнутом. Тести ґрунтуються на статистичному критерії хі-квадрат (критерій згоди Пірсона). Значення статистики хі-квадрат порівнюється з табличними результатами, і в залежності від ймовірності появи такої статистики робиться висновок про її якість. Серед переваг цих тестів — невелика їх кількість та існування швидких алгоритмів виконання. Недолік – невизначеність у трактуванні результатів.

Тести Diehard – набір статистичних тестів для вимірювання якості набору випадкових чисел, були розроблені Джорджем Марсалья і разом розглядаються як один із найсуворіших існуючих наборів тестів.

Набір статистичних тестів Crypt-X розроблений дослідниками науково-дослідного центру з інформаційної безпеки в технологічному університеті Квінсленду є комерційним пакетом програмного забезпечення. Тести використовуються залежно від типу алгоритму генератора, підтримуються потокові та блокові шифри і генератори потоку ключів. У набір включені такі тести: частотний, на послідовність однакових бітів, лінійна складність, складність послідовності, двійкова похідна, зміна точки.

У 1999 р. фахівцями NIST у рамках проекту AES було розроблено набір статистичних тестів «NIST STS» та запропоновано методику проведення статистичного тестування генераторів, орієнтованих на використання у задачах криптографічного захисту інформації. Пакет NIST STS включає 15 статистичних тестів, які розроблені для перевірки гіпотези про випадковість двійкових послідовностей довільної довжини. Усі випробування спрямовані на виявлення різних дефектів випадковості.

Універсальний статистичний тест Маурера заснований на ідеї, що неможливо істотно стиснути послідовність бітів, якщо вони дійсно випадкові. Інакше кажучи, якщо вдалося значно стиснути послідовність біт, згенеровану генератором, він має дефект. Універсальність тесту Маурера в тому, що він може виявити будь-який з основних класів дефектів.

Таким чином, незважаючи на велику кількість тестів, проблема тестування випадкових і псевдовипадкових послідовностей, що застосовуються у криптографії, залишається актуальною на сьогоднішній день. Дослідження у цій галузі тривають, і постійно з'являються ще ефективніші методи оцінки якості генераторів випадкових послідовностей.

УДК 004.021

В.С. Гермак, викладач, К.О. Буравченко, ст. викладач, к.т.н.
germak_vs@ukr.net, buravchenkok@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТРИК MAPE (SMAPE), MSE (RMSE), MAE, R^2 ТА ЇХ ЗАСТОСУВАННЯ ДЛЯ ВИМІРЮВАННЯ ПОХИБКИ ПРИ ПРОГНОЗУВАННІ

Прогнозування, як процес передбачення можливого майбутнього на основі вихідних даних, застосовується у багатьох сферах: демографія, екологія, генетика, метеорологія, соціологія, економіка. Незалежно від сфери застосування, використання прогнозування зумовлено необхідністю приймати різного роду управлінські рішення, щоб уникнути виникнення несприятливих факторів, помилок або для досягнення бажаних, запланованих показників.

Вибір методу прогнозування ґрунтується насамперед на необхідності забезпечення функціональної повноти, достовірності та точності прогнозу, а також необхідності зменшити витрати часу та кошти на здійснення процесу прогнозування. Він залежить від таких факторів: мета прогнозу, його завдання; період, на який формується прогноз; специфіка об'єкта прогнозування; достовірність, повнота та характер вихідної інформації про об'єкт прогнозування; обмежувальні фактори прогнозування; вимоги до результатів прогнозування.

Проблема вибору полягає в тому, що методів дуже багато і важко підібрати ті, які повністю задовольняють цілі складання прогнозу. Тому під час розробки систем прогнозування важливим є вибір правильної метрики якості моделі. Результат вибору безпосередньо впливає на цінність рішення. У прогнозуванні є одна проста істина, висока точність апроксимації даних не гарантує високу точність прогнозів, тому без оцінювання якості побудованої моделі не обійтись.

Однією з найпоширеніших метрик для вимірювання точності прогнозування моделі є MAPE, що означає середню абсолютну помилку у відсотках. Хоча MAPE широко використовується і легко інтерпретується, вона має певні недоліки: вона буде невизначеною, якщо якесь із фактичних значень дорівнює нулю і MAPE не слід використовувати з даними невеликого обсягу. Основна проблема цієї метрики - нестабільність. Цей коефіцієнт можна легко складати для різних рядів, але через поділ на фактичні значення цей коефіцієнт виявився чутливим до масштабу. Також коефіцієнт не є симетричним, завищені прогнози дають більшу похибку, ніж занижені.

Середня квадратична метрика MSE застосовується у випадках, коли треба обрати модель, яка дає менше грубих помилок прогнозу, які стають помітнішими за рахунок піднесення до квадрату. Суть методу полягає в тому, щоб мінімізувати суму квадратів відхилень фактичних значень від розрахункових. Якщо отриману суму розділити на кількість спостережень, то отримаємо MSE. Середньоквадратична помилка підходить для порівняння двох моделей або контролю якості під час навчання, але не дозволяє зробити висновків про те, наскільки добре дана модель вирішує задачу.

Середня абсолютна помилка MAE вимірює середню величину помилок у наборі прогнозів без урахування їх знаку. Модулі у формулі дозволяють позбутися знаків та отримати деяку оцінку відстані від фактичних до розрахункових значень, яку потрібно буде потім мінімізувати. Безперечною перевагою MAE є те, що модулі не збільшують у рази відхилення, тому ця оцінка є більш робастною, ніж MSE. Але цей коефіцієнт складно інтерпретувати і за його допомогою можна порівняти моделі лише по одному ряду даних. А на практиці часто стоїть задача зрозуміти, як поводиться модель за кількома рядами даних у порівнянні з іншими моделями, але отримані з різних рядів MAE складати один з одним не можна.

Коефіцієнт детермінації R^2 вимірює відсоток дисперсії, викликаної моделлю. Фактично це нормована середньоквадратична помилка. Якщо вона близька до одиниці, то модель добре прогнозує дані.

Корінь із середньої квадратичної похибки RMSE це метрика, симетрична щодо знаку, чутлива до великих відхилень від середнього значення фактичної величини. Її часто використовують, коли хочуть уникнути великих помилок у прогнозі. Має таку ж проблему як і MAPE, оскільки кожне відхилення підноситься до квадрату, будь-яке невелике відхилення може суттєво вплинути на показник помилки.

SMAPE або симетрична MAPE. Коефіцієнт повинен бути симетричним, але демонструє зсув у бік завищених прогнозів: завищені прогнози призводять до меншої помилки, ніж занижені.

Таким чином можна прийти до висновку, що найуживаніші метрики MAE і RMSE мають спільну проблему, вони не інтерпретовані, тому самі собою не представляють цінності для оцінки точності моделі. Варто зазначити, що для інтерпретованої та коректної оцінки якості прогнозів рекомендується використовувати набір метрик для прийняття більш обґрунтованих рішень.

УДК 004.421.5

О. О. Майданик, Є. В. Мелешко, А. М. Мацуй
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ГЕНЕРАТОР ПСЕВДОВИПАДКОВИХ ЧИСЕЛ ДЛЯ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОГО РАДІОКАНАЛУ ЗВ'ЯЗКУ БЕЗПІЛОТНОГО ЛІТАЛЬНОГО АПАРАТУ

На сьогоднішній день безпілотні технології все більше займають місце у сфері людської діяльності. Вже велика кількість безпілотників, як наземних так і в повітрі, замінюють важку та небезпечну роботу людини.

Спочатку безпілотні літальні апарати (БПЛА) розроблялися для військових місій, небезпечних для людей. Але з удосконаленням технологій управління та зниженням вартості, їх використання знайшло застосування і в багатьох невійськових сферах.

Найбільш розповсюджена задача для БПЛА – це аерофотозйомка та актуальна на даний час аеророзвідка. Безпілотник, який використовується в таких цілях, має на своєму борту як корисне навантаження відеосистему з камери (або декількох камер), а в деяких випадках денну та нічну (тепловізійну камеру) змонтованих на окремій системі гіростабілізації камер.

В той же час БПЛА вразливі до інформаційних атак, які можуть здійснюватися з різними цілями, основна звичайно перехват інформації, яку він збирає для оператора пристрою. Це все зумовлює необхідність розробки дієвих методів інформаційного захисту дронів від кібератак та перехоплення керування, що є актуальною задачею.

Найчастіше при генерації ключів шифрування використовують генератори псевдовипадкових чисел (ГПВЧ). Дуже часто ГПВЧ є найбільш слабким місцем у системах шифрування. Послідовності, отримані в результаті роботи ГПВЧ повинні бути непередбачуваними та мати довгий період, щоб їх можна було використовувати в криптографічних системах захисту інформації [1].

Для захисту від перехвату керування необхідно застосувати нетипове шифрування даних керування та телеметрії, які передаються по радіообміну. Нетипове шифрування на основі генераторів псевдовипадкових чисел гарно підходить для таких задач, а саме шифрування на основі більярда Сіная, який має дуже високу хаотичність чисел, що генеруються, для ключа шифрування.

Більярд Сіная – це моделювання руху кулі (точки) по полю. Поле має форму опуклих кривих (також може мати іншу форму). Точка частинки рухається по більярду з постійною швидкістю. Коли вона досягає кордону поля більярда, зазнає пружного зіткнення з дзеркальним відображенням відповідно до закону відбиття – кут падіння дорівнює куту відбиття. Між двома зіткненнями точка рухається прямолінійно [2].

На рис. 1 зображено візуалізацію руху математичної точки в деякому більярді Сіная при двох ітераціях.

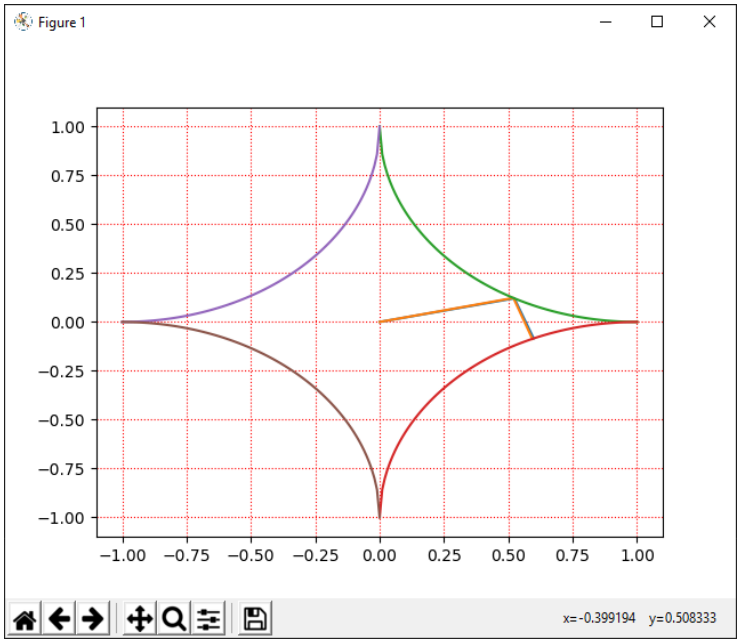


Рис.1 Приклад візуалізації руху математичної точки в більярді Сіная при 2-х ітераціях під час генерації ключа шифрування

З рис. 1 видно, що математичні частинки починають свій рух від центру поля в бік початкового кута, який задається на початку розрахунків. Далі частинка в точці перетину відбивається по закону кутів та рухається до іншої точки перетину. Саме в цих точках перетину ї є число, яке генерується як випадкове.

На даний час проблемою є перехвати керування та заглушення радіозв'язку. ГПВЧ в першу чергу стане корисним для шифрування вже існуючих протоколів зв'язку. Це зручніше реалізувати на рівні радіомодуля (прийомо-передавача). Таким чином, буде забезпечена сумісність з наявними польотними системами. Тобто для забезпечення зв'язку використовується два радіомодуля, які працюють в режимі «прозорий COM-порт».

На початку роботи радіомодулі синхронізуються передачею початкових даних для роботи ГПВЧ (початкові координати на полі та кут першого руху точки) або вже згенерованою таблицею псевдовипадкових чисел. Початкова передача стартових параметрів для ГПВЧ є більш вигідною для надійності шифрування, тому як ключ для кожного пакету даних генерується окремо на приймачі та передавачі і не передаються відкритим радіоканалом. Також ключі не будуть повторюватися, як у випадку попередньо згенерованої скінченої таблиці чисел. По прийому пакету приймаючий модуль використовує поточні координати ГПВЧ (перераховані в ключ) для розшифрування пакету, а наступний пакет з наступним кроком генерації псевдовипадкового числа. Початкові значення координат та кута можна вирахувати (привести до виду числа координат та кута) зчитуюючи значення поточної температури мікроконтролера, серійного номера чіпу чи незаземленої ніжки аналого-цифрового перетворювача (АЦП). Таким чином початкові значення кожного разу не повторюватимуться. На рис. 2 зображено блок-схему модулів зв'язку для реалізації захищеного радіоканалу.

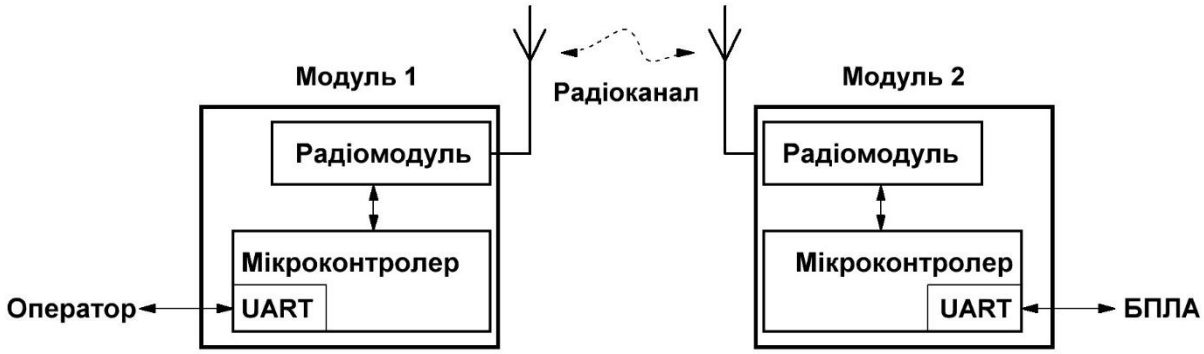


Рис. 2. Блок-схема модулів зв'язку

На основі ГПВЧ також будують системи з плаваючою несучою частотою радіопередачі. Тобто це системи з псевдовипадковою перестройкою радіочастоти (ППРЧ). Наразі такі системи використовують попередньо згенеровані таблиці значень частоти і перед використанням їх необхідно оновлювати та синхронізувати з усіма модулями зв'язку, які використовуються в передачі даних. ГПВЧ на основі більярду може значно посилити надійність зв'язку в таких системах.

Як правило, стандартний зв'язок з БПЛА виконується на відкритих частотах та з поганим шифруванням, а частіше зовсім без шифрування. Тому літальний апарат можливо перехопити по керуванню, а це гарантована втрата дороговартісного обладнання та корисної інформації, яку в собі накопичив дрон при аерофотозйомці (аеророзвідці). Тому у даній роботі для вирішення цієї проблеми пропонується використати шифрування Вермана на основі ключів, згенерованих за допомогою хаотичного більярда Сіная. Пропонується генерувати нові ключі для кожного нового польоту дрона. Запис нових ключів здійснювати перед кожним польотом синхронно у систему дрона та систему управління ним.

Отже, наразі проблема підвищення кібербезпеки дронів стоїть дуже гостро та потребує вирішення. Шифрування даних допоможе уникнути перехоплення керування, а радіоканал на основі ППРЧ значно знизить можливість як перехоплення, так і заглушення зв'язку.

Список літератури

1. Sinai Y.G. Dynamical systems with elastic reflections // Mathematical Surveys. – 1970. - vol. 25, no. 2, pp. 137-189.

2. Собінов О.Г. Простий генератор псевдовипадкової послідовності // Інформаційні технології та комп'ютерна інженерія: зб. тез доп. наук.-практ. конф., м. Кіровоград, 4 груд. 2014 р. – Кіровоград: КНТУ, 2014. – С. 184.

УДК 004.021

К.Є. Крюков, ст. 2 курсу
kriucovkie@gmail.com
Науковий керівник: В.С. Гермак, викладач
Центральноукраїнський національний технічний університет, м. Кропивницький

ПОРІВНЯЛЬНИЙ АНАЛІЗ КРИПТОГРАФІЧНО СТІЙКИХ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

Випадкові числа знайшли застосування у багатьох галузях науки і техніки: наукові дослідження з формуванням випадкових вибірок, комп'ютерне моделювання фізичних процесів, чисельний аналіз, комп'ютерні та азартні ігри. Одна з найактуальніших сфер застосування – криптографія. Випадкові числа в прикладній криптографії можуть використовуватися для генерації ключів, одноразових випадкових чисел в протоколах аутентифікації, генерації одноразових шифроблокнотів, в якості солі в схемах цифрового підпису. Для отримання таких чисел використовуються генератори псевдовипадкових чисел.

Вимоги до криптостійких генераторів дуже жорсткі, вони повинні володіти такими властивостями: довгий період, що гарантує відсутність зациклювання послідовності в межах розв'язуваної задачі; ефективність (швидкість роботи алгоритму та малі витрати пам'яті); відтворюваність (можливість заново відтворити раніше згенеровану послідовність чисел будь-яку кількість разів) портативність (однакове функціонування на різному устаткуванні та операційних системах); швидкість отримання наступного елемента послідовності знаючи попередній (для поділу послідовності на кілька потоків); криптографічна стійкість; хороші статистичні властивості, завдяки яким послідовність неможливо відрізнити від істинно випадкової; реалізація генератора має бути ефективною по відношенню до апаратних ресурсів (через необхідність реалізації на пристроях з вкрай обмеженими апаратними ресурсами – смарт-картах, електронних ключах тощо).

Виділяють три класи методів, які використовуються у криптографії: засновані на криптографічних алгоритмах; засновані на односторонніх функціях (математично складних задачах); спеціальні реалізації.

У першому випадку генерація може бути реалізована одним із наступних способів: застосуванням безпечного блокового шифру до деякого випадкового ключа (шифрування вихідного тексту блоками, причому вміст кожного блоку не впливає на результат шифрування інших блоків); застосуванням криптографічностійкої хеш-функції до вихідного секретного випадкового числа; застосуванням потокових шифрів, які, у свою чергу, самі працюють базуючись на генерації псевдовипадкових біт які комбінуються деяким чином з бітами відкритого тексту (наприклад RC4). На блокових шифрах ґрунтуються такі алгоритми: DES (має розмір блоку 64 біт та ключ 56 біт; довжина ключа стала очевидною вадою алгоритму, її виявилося недостатньо, щоб протистояти атакам повним перебором; існує покращена версія алгоритму, яка називається Triple DES); AES (алгоритм, який замінив собою шифр DES, розмір блоку становить 128 біт та розмір ключа 128, 192 і 256 біт); ГОСТ 28147-89 (шифр заснований на 32-раундовій мережі Фейстеля з 256-бітним ключем). Криптографічна хеш-функція повинна володіти такими властивостями: опір пошуку першого прообразу, опір пошуку другого прообразу, стійкість до колізій, стійкість до подовження прообразу. Приклади безпечних хеш-функцій: VSH, MD4 і MD5, SHA-1, SHA-2, SHA-3, SHA-256 (використовується в алгоритмах хешування біткоїну та інших криптовалют, вихідний хеш становить 256 біт, ентропію можна визначити як множину значень від 1 до 2^{256}, що робить злом вкрай трудоємним процесом, що зводиться до послідовного перебору).

В другому класі алгоритмів для генерації використовуються односторонні функції з ключем, якими може служити функція піднесення до квадрату, експоненціювання, множення простих чисел у скінченому полі. Ці алгоритми використовують складність розв'язання деяких задач для отримання псевдовипадкових чисел, захищених від криптоаналізу. Прикладами є алгоритми Блюма-Блюма-Шуба (недоліком якого є його низька швидкість, що робить його малозастосовним для використання в потокових шифрах), Блюма-Мікалі, Мікалі-Шнорра та Шаміра.

Третій клас алгоритмів ґрунтується на використанні стохастичних методів перетворення чисел. Існує ціла низка генераторів які розроблялися з урахуванням криптографічної стійкості: алгоритм Ярроу, алгоритм Fortuna (спадкоємець алгоритму Ярроу), функція CryptGenRandom.

Таким чином, існує дуже багато різнотипних криптографічностійких генераторів псевдовипадкових чисел. Деякі з цих генераторів здаються практично незламними в реальних умовах, проте будь-яка надійність відносна, а алгоритми атаки не стоять на місці, не кажучи вже про людський фактор, який здатний зруйнувати навіть найзахищенішу систему. А тому, незважаючи на складність і якість обраного для використання генератора, він буде справді гарним інструментом лише в комплексно побудованій системі інформаційної безпеки.

УДК 004.8 656.02

В.С. Лебеденко, магістрант, О.А. Кислун, доцент ,
kyslun@gmail.com, lebedenkovitalik0@gmail
Центральноукраїнсь.соткий національний технічний університет, м.Кропивницький

ОГЛЯД МЕТОДІВ РОЗВ'ЯЗАННЯ ЛОГІСТИЧНИХ ЗАДАЧ ПОШУКУ ОПТИМАЛЬНИХ МАРШРУТІВ

Задачі логістики - це повсякденні задачі сучасного світу, які корінням входять в глибоке минуле, тож за попередні роки накопичено величезний досвід в їх розв'язанні. Широке використання обчислювальної техніки для пошуку оптимальних рішень привело до появи різноманітного програмного забезпечення в даній сфері, проте еволюційний розвиток потребує все нових програмних продуктів, тим більше, що час від часу виникають якісь нові, раніше не передбачені існуючими рішеннями, умови та специфічні вимоги. Отже, виникає задача аналізу набуття отриманого при розв'язанні логістичних задач, при цьому сам же огляд обмежено лише зацікавленістю в області оптимізації доставки.

Задачі доставки та їх рішення:

- транспортна задача (мінімізації витрат на розподілення одного товару або ресурсу з множини вузів базування до множини пунктів призначення відповідно до планів поставки);

- задача пошуку евклідового мінімального кістякового дерева (визначення на графі дерева з мінімальною сумою відстаней ребер, що поєднують всі існуючі вершини графа) в якості базового розв'язання використовуються алгоритми Прима та Крускала;

- задача комівояжера (знаходження мінімального шляху повного обходу вказаних пунктів з поверненням на початок маршруту при зазвичай наявній умові одноразового відвідування пунктів маршруту) загалом гарантовано точно може бути вирішена лише шляхом повного перебору всіх існуючих варіантів маршруту, проте в залежності від кількості пунктів маршруту наявна дуже велика складність перебору, а існуючі ж евристичні та комбіновані методи (найближчого сусіда, гілок та границь, ...) загалом знаходять непогані розв'язки, проте вони можуть бути гіршими за оптимальні;

- задача про найкоротший шлях (пошук найкоротшого шляху, що загалом може мати різновиди: пошук найкоротших шляхів з одного входу, пошук найкоротших шляхів з одним виходом, пошук найкоротшого шляху між парою пунктів) - базова задача, яку і покладено в основу розробки алгоритмів оптимізації маршрутизації (алгоритми: Дейкстри, Беллмана-Форда, пошуку A*, Флойда-Воршелла, Джонсона, ...).

Знаходження найкоротших шляхів з одного входу до всіх інших пунктів алгоритм Дейкстри проводить як покрокову мінімізацією шляху від пункту при почерговому оціненні всіх можливих переміщень з пункту та визначенні з фіксацією оптимальнішого переміщення (початкові шляхи до пунктів довизначаються потенційно більшими за можливі, початковий шлях з входу визначається нулем, переміщення оптимізаційного пошуку з пункту проводиться по напрямку довжини шлях, що відшукується, з виключенням відпрацьованого пункту, завершення пошуку шляху визначається відсутністю пунктів для пошуку нового оптимальнішого кроку шляху).

Знаходження найкоротшого шляху з одного входу до всіх інших пунктів алгоритм Беллмана-Форда проводить як покрокову мінімізацією шляху до пункту при почерговому оціненні всіх можливих переміщень з кожного з пунктів та визначенні з фіксацією оптимальнішого (початкові шляхи до пунктів визначаються потенційно більшими за можливі, початковий шлях з входу визначається нулем, завершення пошуку шляху визначається відсутністю оптимальнішого шляху до жодного з множини пунктів).

Алгоритм пошуку A* - це варіант алгоритму Дейкстри з евристичною оцінкою відстані між вузлом і фінішем для надання переваги вибору відкриття вузлів з метою усунення аналізу потенційно довших шляхів.

Алгоритм Флойда-Воршелла (інші назви: Флойда, Роя–Воршелла, Роя–Флойда или WF) використовується для знаходження оптимального шляху між парою (двома пунктами), що побудовано на покроковій мінімізації шляху між парою через третю точку (завершення алгоритму пошуку оптимального шляху виявляється відсутністю мінімізаційного шляху між будь-якою парою.

Алгоритм Флойда-Воршелла Джонсона призначено для знаходження найкоротших шляхів між усіма парами вершин та базується на зміні ваги та використанні алгоритмів Беллмана-Форда та Дейкстри.

Оскільки кінцевою метою дослідження в рамках, якого наводиться огляд, є розробка програмного забезпечення для пошуку оптимальних маршрутів за певних специфічних умов постачання, то, відповідно, саме алгоритми пошуку найкоротшого шляху лягають в основу подальшої розробки та потребують адаптації, що враховуватиме додаткову затратність накладених умов.

УДК 004.021

О.А. Меркулов, ст. 2 курсу
merculovoa@gmail.com
Науковий керівник: В.С. Гермак, викладач
Центральноукраїнський національний технічний університет, м. Кропивницький

КЛАСИФІКАЦІЯ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

Випадкові числа використовуються дуже давно і мають широку сферу застосування: соціологічні та наукові дослідження (підготовка випадкових вибірок під час збору даних, при перевірці математичних теорем чи дослідженні фізичних явищ); моделювання (комп'ютерне моделювання фізичних явищ та математичне моделювання); криптографія та інформаційна безпека (в криптографічних алгоритмах шифрування); ухвалення рішень в автоматизованих експертних системах; оптимізація функціональних залежностей; ігри (генерування 3D-моделей, текстур, світів, створення варіативності поведінки в іграх).

Генератори випадкових чисел за способом одержання чисел поділяються на апаратні, табличні та алгоритмічні.

Табличні генератори як джерело випадкових чисел використовують заздалегідь підготовлені таблиці, що містять перевірені некорельовані числа. Недоліки такого способу: використання зовнішнього ресурсу для зберігання чисел, обмеженість послідовності, заздалегідь визначені значення.

Апаратні генератори істинно випадкових послідовностей повинні мати джерело ентропії. Розробка таких генераторів досить складне завдання. Джерелами ентропії можуть бути: підкидання монети; часові затримки між моментами випромінювання частинок у процесі радіоактивного розпаду; теплові шуми під час роботи напівпровідникового діода чи резистора; частотні відхилення вільно працюючого генератора частот; фотоефект - випромінювання електронів речовиною під впливом світла; звук від мікрофона чи відео з підключеної камери; стан деяких блоків пам'яті.

Алгоритмічний генератор є комбінацією фізичного генератора та детермінованого алгоритму. Такий генератор використовує обмежений набір даних, отриманий з виходу фізичного генератора, для створення довгої послідовності чисел шляхом перетворення вихідних чисел. Даний вид генераторів представляє найбільший інтерес через його очевидні переваги над генераторами випадкових чисел інших видів. Через дорожнечу апаратних генераторів випадкових чисел у більшості випадків як джерело ентропії використовуються ресурси обчислювальної машини, на якій виконується програма генерації. Як джерело ентропії можуть використовуватися: стан системного годинника; час затримок між натисканнями клавіш клавіатури або рухами мишки; вміст буферів введення/виведення; значення, що отримуються під час роботи системи (час завантаження системи, мережева активність тощо). Перевагами алгоритмічних генераторів є швидкодія, компактність реалізації. Основний недолік – низька якість «випадковості» – такі послідовності, як правило, не проходять більшість тестів на випадковість.

Таблиця 1

Основні алгоритмічні методи генерації псевдовипадкових чисел

п/п	Назва алгоритму	Особливості алгоритму
.	Метод серединних квадратів	Недолік цього методу — наявність кореляції між числами послідовності, а в деяких випадках випадковість взагалі може бути відсутньою.
.	Лінійний конгруентний метод	Перевагою методу є простота реалізації і швидкість. Однак такі генератори не можна використовувати в криптографії, тому що їх легко «зламати». Тим не менш, ці методи є корисними для некриптографічних застосунків, у моделюванні або в ігрових програмах.
.	Адитивний генератор.	При великих запізненнях даний метод стає неефективним через високі вимоги до пам'яті.
.	Інверсний конгруентний генератор.	Основний недолік інверсних генераторів - це трудоємність операції обернення елемента кінцевого поля.
.	Вихор Мерсенна.	Метод забезпечує рівномірний розподіл псевдовипадкових чисел, однак генератор не призначений для отримання криптостійких послідовностей випадкових чисел.
	Рандомізація перемішуванням.	Важливий клас методів, заснованих на комбінуванні генераторів випадкових чисел. Однак методи перемішування мають серйозний недолік - вони змінюють порядок слідування чисел, але не самі числа.
	Криптографічно стійкі генератори	Методи засновані на криптографічних алгоритмах; методи засновані на односторонніх функціях (математично складних задачах); спеціальні реалізації.

Отже, обширність сфери застосування випадкових чисел породжує значну кількість алгоритмів для їх генерації. При виборі оптимального алгоритму треба враховувати галузь, в якій буде застосовуватись даний алгоритм та обчислювальні ресурси, які можна використовувати при реалізації поставленої задачі.

УДК 519.85

Є.С. Прокопенко, студент 3-го курсу
Науковий керівник: О.С. Улічев, к.т.н.,
prokopenko19712002@gmail.com, askin79@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ОПТИМІЗАЦІЯ РОЗКРОЮ ЛІНІЙНОГО СТЕРЖНЯ

Економія матеріальних ресурсів є найважливішим фактором підвищення ефективності суспільного виробництва. Одним з найважливіших засобів, що забезпечують економію різних видів ресурсів, є їхній раціональний розкрій. За статистикою Світової організації торгівлі тільки виробничі підприємства пов'язані з виробництвом паперу та виробів з паперу щороку на відходах втрачають близько 180 млн. дол., окрім прямих збитків неефективність розкрою призводить до ряду непрямих наслідків, як то - збереження екології, економії природних ресурсів тощо.

Завдання розкрою-упаковки викликають широкий інтерес як у виробництві, так і в теоретичних дослідженнях. Класичне завдання одновимірного розкрою (one-dimensional cutting stock problem, 1DCSP) розглядається в багатьох публікаціях [2,3-4]. Це завдання полягає в мінімізації кількості ідентичних прутків матеріалу, що використовуються для розкрою певного набору заготовок.

Для розв'язання задачі запропоновано ряд евристик різної складності. Найпростіші це перший підходящий (First Fit, FF), перший підходящий з упорядкуванням (First Fit Decreasing, FFD) та їх варіації. Більшої старанності при розробці та дослідженні вимагають такі методи, як жадібний алгоритм та його варіація, метод послідовного уточнення оцінок (sequential value correction method SVC), генетичні алгоритми[6], методи часткового перебору. Деякі з евристичних схем підключають безперервну релаксацію з генерацією стовпців для обчислення нижньої межі та отримання округленого рішення. Існують різні оцінки якості евристик, наприклад ймовірнісні характеристики чи докази асимптотичної точності на певному класі задач.

Для найпростішого випадку – лінійного розкрою стержня (труба, прут, арматура, дошка і т.д.) формальна постановка задачі виглядає наступним чином.

Маємо вектор деталей:

$$x=(x_1, x_2, x_3, \ldots\ldots x_m) \quad (1)$$

кожна деталь характеризується:
l_i - довжина деталі;
k_i – кількість шт. деталі.
Деталі нарізуються із заготовок довжини L
Необхідно скласти такі підмножини $|x_i|$, що включають по n елементів, щоб задовольнити умови:

$$\sum_{i=1}^{n} li \le L \quad , \quad (2)$$

кількість підмножин (розбивок) Z, яка фактично визначає кількість необхідних заготовок, повинна бути мінімальною.

І мінімізувати функцію, що визначає відходи:

$$\mu(x) = \sum_{i=1}^{z}\left(L - \sum_{i=1}^{m} li\right) \quad (3)$$

Для реалізації можна запропонувати простий алгоритм, що базується на евристиці з послідовним поліпшенням плану. З використанням додаткових обробок даних на різних етапах алгоритму. Вибір даного підходу доречний у випадку, коли план розкрою оцінюється в межах декількох сотень деталей. Так як в основі алгоритму використовуються перебори (пошук найкращого варіанту укладки) то для великої кількості деталей в плані час роботи алгоритму буде суттєво збільшуватись.

На первинному етапі масив деталей сортується за зменшенням довжини. Першими упаковуються найдовші деталі, далі проводиться зсув масиву (вилучення найдовших деталей) і пошук можливого укладання решти.

На наступному кроці залишок заготовки аналізується на найкращий спосіб укладання туди деталей, що залишилися, починаючи з найменших довжин з перевіркою після чергової ітерації чи покращився план.

Уникнути довгих переборів дозволяють допоміжні масиви з розміщенням довжин не упакованих деталей. Використання цих масивів дає можливість швидше знаходити претендентів.

Запропонований алгоритм зручно інтегрується в Excel-таблицю, якщо реалізувати сріпт оптимізації розкрою на VBA.

Такий підхід буде досить зручним для використання і не потребуватиме додаткового навчання персоналу. Окрім того формування плану у вигляді таблиці Excel спрощує імпортування даних, пост обробку та аналіз.

Архітектура програмного додатку, реалізованого на базі MS Excel і вбудованої мови VBA, матиме наступний вигляд (рис. 1).

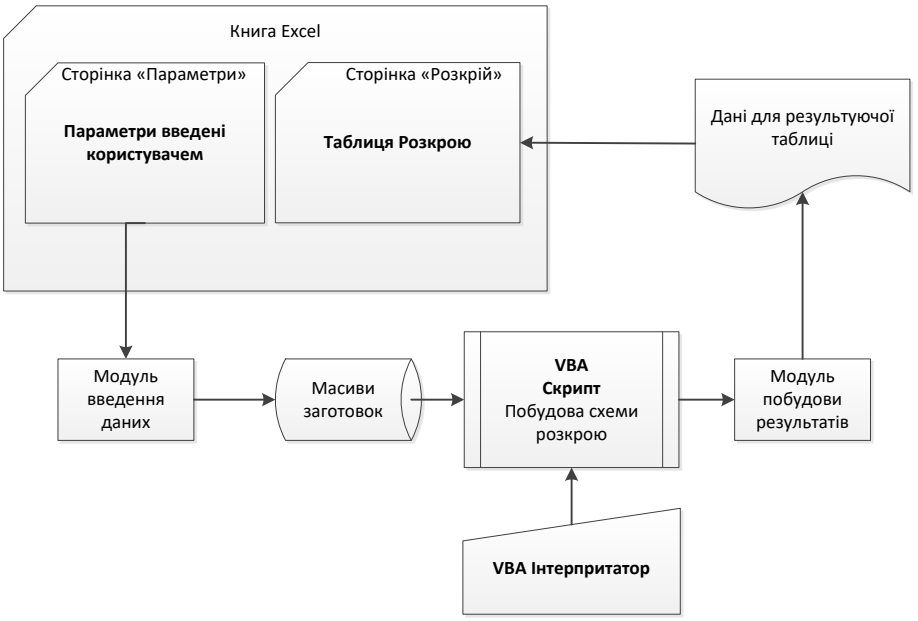


Рис. 1. Архітектура програмного додатку

Висновки

Як показують дослідження, задача розкрою на сьогоднішній день ще не вирішена остаточно. Хоча запропоновано багато різних методів, кожен із них має свої недоліки.

Основним недоліком точних методів є високі обчислювальні вимоги, пов'язані з великою розмірністю задачі. В класі точних методів, в більшості випадків, розмірність задачі експотенціально зростає зі збільшенням кількості вхідних даних (кількості різних деталей, що присутні в плані розкрою).

Більш швидкі методи мають нижчу точність порівняно з точними, водночас вибір правильного евристичного підходу дозволяє отримати прийнятні результати.

Список літератури

1. P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem (Part I). Oper. Res., 9. 1961. -P.849-859.

2. Березовський В.І., Березовська Ю.І. Методи оптимізації розкрою матеріалів // Вісник Харківського національного технічного університету сільського господарства імені Петра Василенка. Серія: Технічні науки. – 2014. – Вип. 152. – С. 195-202.

3. Калашніков О.В., Максимов І.О. Оптимізація розкрою заготовок на машинобудівних підприємствах // Технологічні системи. – 2017. – Вип. 2(76). – С. 47-56.

4. Канторович Л.В. Вибрані праці: у 2-х т. Т. 2: Математична теорія економічної динаміки. – К.: Наукова думка, 1990. – 592 с.

5. Кравченко В.І. Алгоритми оптимізації розкрою // Проблеми програмування. – 2012. – Вип. 2. – С. 130-139.

6. Хіміч В.І., Бакумова І.В. Оптимізація розкрою матеріалів на базі генетичних алгоритмів // Науковий вісник Національного університету біоресурсів і природокористування України. Серія: Технічні науки. – 2016. – Вип. 235. – С. 41-48.

УДК 004.7

Г. М. Дрєєва[1], О. М. Дрєєв[1], Є. В. Мелешко[1], І. В. Миронець[2]
elismeleshko@gmail.com
[1]*Центральноукраїнський національний технічний університет, м. Кропивницький*
[2]*Черкаський державний технологічний університет, м. Черкаси*

ПРОГРАМНА ІМІТАЦІЙНА МОДЕЛЬ КОМП'ЮТЕРНОЇ МЕРЕЖІ З ФРАКТАЛОПОДІБНИМ ТРАФІКОМ

У роботі було розроблено програмну імітаційну модель комп'ютерної мережі з симуляцією мультифрактального трафіку на основі ланцюга Маркова для тестування алгоритмів марштуртизації. Для генерації структури комп'ютерної мережі розроблено метод на основі теорії складних мереж [1-3] та удосконаленого методу Барабаші-Альберт [4]. Для симуляції мережевого трафіку розроблено метод генерації мультифрактальної бінарної послідовності з використанням ланцюга Маркова [5-6].

Комп'ютерна мережа у розробленій моделі представлена повнозв'язним неорієнтованим зваженим графом, в якому вузлами є маршрутизатори, а ребрами – мережеві зв'язки між ними (рис. 1). Вага ребер – величина обернена до пропускної спроможності каналу зв'язку. Вузли містять у собі черги, в яких розміщуються прийняті пакети перед визначенням маршруту їх відправлення та відправкою на наступний вузол. Час у моделі представлений дискретними ітераціями. Маршрутизація здійснюється на основі тих алгоритмів, які необхідно протестувати на моделі.

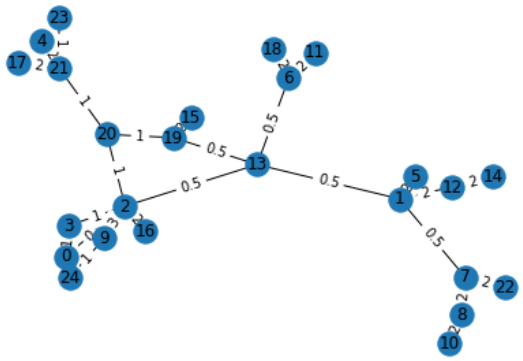


Рисунок 1 – Приклад структури комп'ютерної мережі, згенерованої у розробленій програмній імітаційній моделі

Для симуляції мережевого трафіку у розробленій програмній імітаційній моделі було запропоновано метод генерації бінарної мультифрактальної послідовності на основі ланцюгів Маркова із стохастичним автоматом, який дозволив керувати фрактальною розмірністю бінарного ряду на різних масштабах.

З причини того, що трафік в мережах сегментується пакетами, було вирішено моделювати трафік на рівні дискретизації не за байтами, а за пакетами. В результаті отримано дискретні джерела пакетів, де стан "0" та "1" відповідають за відсутність передачі та передачу пакетів. Для генерації фрактального трафіку було запропоновано використовувати генератор трафіку *G* (рис. 2), який містить два стани «1», «0» та ймовірності залишити стан незмінним p(«0»→«0»)=p0 та p(«1»→«1»)=p1. Генератор видає значення поточного стану дискретно по події відсутності або здійсненню зміни стану.

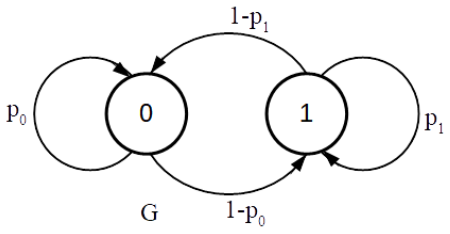


Рисунок 2 – Генератор фракталоподібної бінарної послідовності

Для такого фрактального генератора інтенсивність трафіку можна визначити за формулою:

$$\lambda = \frac{1 - p_0}{2 - (p_0 + p_1)}. \quad (1)$$

Для генерації мультифрактального трафіку пропонується на базі генератора *G* використовувати наступну каскадну модель генератора бінарного трафіку, зображену на рис. 3.

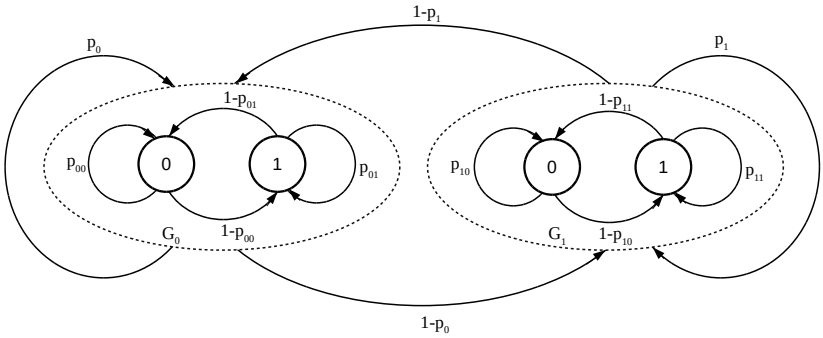


Рисунок 3 – Генератор мультифрактальної бінарної послідовності

Для такого мультифрактального генератора інтенсивність трафіку можна визначити за формулою:

$$\lambda = \frac{1-p_1}{2-(p_0+p_1)} \cdot \frac{1-p_{00}}{2-(p_{00}+p_{01})} + \frac{1-p_0}{2-(p_0+p_1)} \cdot \frac{1-p_{10}}{2-(p_{10}+p_{11})}. \quad (2)$$

Приклад згенерованого трафіку наведено на рис. 4. З метою візуально оцінити утворені ряди на рис. 6 показано бінарний ряд, який агреговано по 25 відліків.

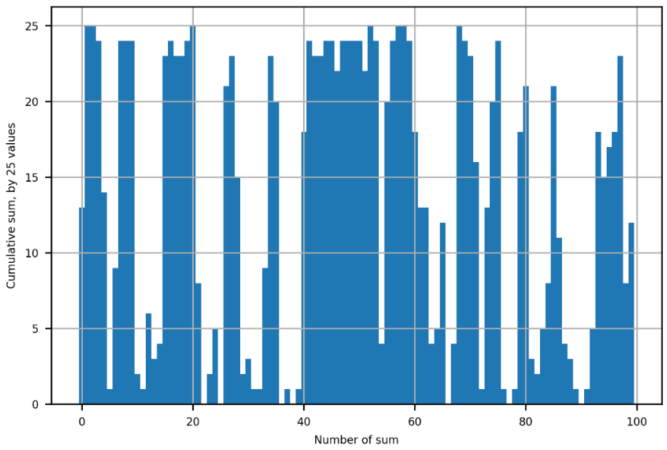


Рисунок 4 – Приклад згенерованого каскадним генератором мультифрактального трафіку, агрегованого для наочності по 25 відліків

В результаті числового експерименту, встановлено факт можливості регулювання показника Херста на заданому часовому масштабуванні. Отримані часові ряди за допомогою каскадного генератора бінарної послідовності мають мультифрактальні властивості. Тобто каскадний генератор має більше можливостей до підгонки до реальних прикладів бінарного трафіку. Використання каскадного генератора є можливим лише при можливості його легкого налаштування, що фактично означає можливість отримати значення фрактальної розмірності d та ймовірностей p_0, p_1, p_{00}, p_{01}, p_{10}, p_{11} з прикладів реального трафіку.

Таким чином розроблено метод генерації трафіку з мультифрактальними властивостями у вигляді бінарного часового ряду на рівні "пакет є" – "пакету немає" та метод програмного імітаційного моделювання комп'ютерної мережі для тестування алгоритмів маршрутизації.

Список літератури

1. Barabási A.-L. Network science // Cambridge University Press. – 2018. – 475 p. – URL: http://networksciencebook.com/

2. Traag V.A. Algorithms and Dynamical Models for Communities and Reputation in Social Networks // Springer International Publishing. – 2014. – P. 229. – URL: https://doi.org/10.1007/978-3-319-06391-1

3. Barabási A.-L., Albert R. Emergence of scaling in random networks (англ.) // Sci-ence, Vol. 286, No. 5439. – 1999. – P. 509-512. – URL: https://doi.org/10.1126/science.286.5439.509

4. Drieieva H.M., Smirnov O.A., Drieiev O.M., Smirnova T.V. A Fractal Analysis of a Self-similar Traffic Generator Based on a Markov Chain // Central Ukrainian Scientific Bulletin, Engineering sciences, Vol. 2(33). – 2019. – pp. 161-172, URL: http://mapiea.kntu.kr.ua/eng/archive/33/33_Drieieva.html

5. Drieieva H., Drieiev O., Meleshko Ye., Yakymenko M., Mikhav V. A method of determining the fractal dimension of network traffic by its probabilistic properties and experimental research of the quality of this method // CEUR-WS, Vol. 3171, Gliwice, Poland. – 2022. – P. 1694-1707, URL: http://ceur-ws.org/Vol-3171/paper120.pdf

УДК 629.7.02+004.3

О. О. Майданик, Є. В. Мелешко, А. М. Мацуй, С. В. Шимко
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ МЕТОДІВ СТАБІЛІЗАЦІЇ ВІДЕО ДЛЯ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ

Метою даної роботи було дослідження будови гіростабілізованих підвісів відеокамер для дронів, що застосовуються для механічної стабілізації відео при відеоспостереженні або візуальній навігації. Від якості стабілізації відео залежить і якість отриманого зображення з дрону, а отже і кількість одержаної корисної інформації.

Існує дві основні групи методів стабілізації відео [1-5]: *оптико-механічна* та *цифрова стабілізація*.

Для максимального покращення якості зображення з відеокамери безпілотного літального апарату (БПЛА) та зведення до мінімуму наслідків дрижання камери, необхідно в першу чергу виконувати механіко-оптичну стабілізацію відео, а потім за необхідності доповнювати її цифровою стабілізацією.

Тільки цифрова стабілізація без механіко-оптичної виконується лише з метою здешевлення дрону. Оптико-механічна стабілізація зазвичай базується на показах гіроскопу. Невеликі дрони, що літають низько, зазвичай мають фіксовані камери без механічної стабілізації, хоча можуть мати цифрову стабілізацію відео [1, 2]. Відео та зображення з фіксованих камер не є гладкими через вібрацію пропелера та вплив навколишнього середовища. Отримані нестабільні відео часто потрібно згладжувати для подальшої роботи з ними. Цифрова стабілізація може дещо виправити становище, але найбільш якісне зображення дозволяє отримати механічна стабілізація на основі гіростабілізованих підвісів відеокамери, а також поєднання механічної та цифрової стабілізації [3]. Механічна стабілізація зазвичай базується на показах гіроскопа [3-5].

Цифрова стабілізація включає такі основні елементи: виявлення зміщення для поточного кадру, компенсація даного зсуву і періодичне оновлення фону, щодо якого відбувається стабілізація.

Виявлення зміщення поточного кадру. Базовий спосіб ґрунтується на кореляційному підході для визначення зміщення, можна коротко описати так:

1) Береться центральна частина фону на базовому зображенні. Центральна частина не повинна бути надто маленькою, інакше у кореляційної функції не вистачатиме даних для стабільної роботи.

2) На поточному кадрі вибирається частина такого ж розміру, але зміщена щодо центру картинки.

3) Для кожного зміщення розраховується деяка метрика, що описує кореляцію центральної частини фону та поточного зображення. Для цього може бути використана, наприклад, сума квадратів різниці для кожної точки цих двох зображень або, наприклад, сума абсолютної різниці для кожної точки.

4) Зміщення, для якого кореляція максимальна за обраною метрикою і буде шуканим зміщенням.

Повний перебір всіх варіантів можливого зміщення зображень щодо один одного займає досить багато часу (складність алгоритму O(n^2), де n – число точок зображення). Тому необхідно обов'язково використовувати оптимізацію, найчастіше для автоматизації використовують градієнтний спуск.

Компенсація зміщення. Для компенсації зсуву зміщуємо поточне зображення на знайдений зсув з протилежним знаком та заповнюємо фоном порожні області біля краю. Для підвищення якості цифрової стабілізації можна використовувати також визначення та компенсацію субпіксельного зсуву.

Оновлення фону. Як фон можна використовувати просто будь-який попередній кадр. Однак, якість стабілізації помітно покращується, якщо як фон використовувати усереднене по багатьох кадрах зображення. Фон бажано періодично оновлювати, щоб компенсувати можливі зміни освітленості на сцені. При оновленні фону потрібно переконатися, що фонове значення є досить контрастним і неоднорідним. В іншому випадку кореляційна функція не матиме чіткого максимуму, що сильно знизить точність роботи стабілізатора. Також дуже небажано, щоб на тлі були присутні об'єкти, що рухаються.

Оптико-механічна стабілізація зазвичай базується на показах гіроскопу, наприклад, на основі 2-х чи 3-х осьових гіростабілізованих підвісів з одним мікроконтролером або гіростабілізованих підвісів з енкодерами та декількома мікроконтролерами.

Гіростабілізовані підвіси – це пристрої для стабілізації відеокамери в просторі, які змонтовані на дроні (літак, квадрокоптер), який через свою фізику польоту знаходиться постійно в русі, тобто камера змонтована на 2-х або 3-х осьовому карданному підвісі (рідше на 1-но осьовому). На кожній з осі знаходиться двигун, який в залежності від зворотнього зв'язку від гіроскопа, що знаходиться на камері, повертає на заданий кут вісь для підтримання камери в горизонті або на кут, який задається пілотом. Вісі називаються як і в авіації: тангаж, крен та рискання (Pitch, Roll, Yaw).

На рис. 1 зображено конструкцію 3-х осьового гіростабілізованого підвісу, який є наймасовішим і найпростішим. Така будова підвісу має свою перевагу в простоті конструкції. Кожна з осей змонтована через двигун, який відхиляє вісь в потрібному напрямку. Керування виконується одним мікроконтролером та окремим гіроскопом, який змонтовано на одній і тій же вісі з камерою.

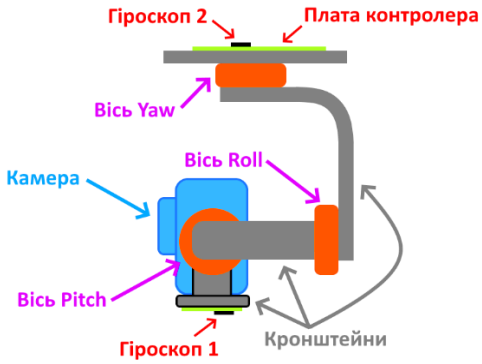


Рис. 1. Схема будови 3-х осьового гіростабілізованого підвісу відеокамери

Гіроскопи потрібні для орієнтації в просторі. Мікросхема гіроскопа може бути оснащена акселерометром, тому що сам по собі гіроскоп у вигляді мікросхеми має дуже багато погрішностей. Тобто, на кристалі кремнію витравлено складний підвіс, який рухається всередині корпусу мікросхеми, а вимірювання відхилень відбувається за рахунок вимірювання ємності. Тому дані гіроскопи при рухах накопичують дуже велику похибку. Для мінімізації цього використовують акселерометр, який вимірює прискорення. Завдяки акселерометру вираховується похибка положення гіроскопа в даний момент часу. Саме таким методом можна отримати положення гіроскопа в просторі. Тобто гіроскоп показує відхилення від горизонту (початкового положення) по 3-х осях (мікросхема має 3-х осьовий гіроскоп та 3-х осьовий акселерометр). Мікроконтролер постійно зчитує положення гіроскопа в просторі та вираховує, на який кут необхідно повернути ту чи іншу вісь, гіроскоп виконує функцію зворотного зв'язку. Другий гіроскоп, який стоїть на платі з мікроконтролером потрібен для початкового калібрування положення гіроскопів.

2-х осьові підвіси з осями Pitch, Roll, як і 3-х осьові, використовуються на квадрокоптерах, рідше на літаках. Підвіси Pitch, Yaw використовуються на літаках. Одноосьові підвіси використовуються дуже рідко на літаках з віссю Roll.

Описані вище системи механічної стабілізації мають ряд недоліків, а саме те, що обчислення положення осей виконує один мікроконтролер відносно гіроскопа та акселерометра, які в цій системі є зворотнім зв'язком. Саме по собі зчитування та фільтрація даних з гіроскопа та акселерометра є затратним, з точки зору обчислень, а для нормальної роботи треба зчитувати значення з високою частотою. Тому стабільність такої системи невисока. При тому, що дані гіроскопа та акселерометра потребують ресурсів треба рахувати вектори положень двигунів осей. Це також не проста задача, яка потребує обчислень з плаваючою точкою та більш потужного мікроконтролера, який буде заздалегідь дорожче коштувати. Для вирішення цієї проблеми використовують систему на основі розділених мікроконтролерів зі зворотнім зв'язком відносно енкодера змонтованого на кожній вісі. Енкодер – це пристрій, який дає значення положення повороту осі (як потенціометр, але енкодер може здійснювати повні оберти на 360°).

Тож, для максимального покращення якості зображення з відеокамери дрона та зведення до мінімуму наслідків дрижання камери, необхідно в першу чергу виконувати механіко-оптичну стабілізацію відео, а потім за необхідності доповнювати її цифровою стабілізацією. Тільки цифрова стабілізація без механіко-оптичної виконується лише з метою здешевлення дрону. Повна відсутність стабілізації застосовується лише в найдешевших дронах для задач, що не вимагають високої точності результатів.

Список літератури

1. Aswini, N., Uma, S.V. Video Stabilization for Drone Surveillance System // Data Science and Computational Intelligence (ICInPro 2021), Communications in Computer and Information Science, vol. 1483. Springer, Cham. – 2021. – P. 468-480. – URL: https://doi.org/10.1007/978-3-030-91244-4_37

2. Kowal D. Considerations for opto-mechanical vs. digital stabilization in surveillance systems // Proceedings, vol. 9451, Infrared Technology and Applications XLI, 94510B, Event: SPIE Defense + Security, 2015, Baltimore, Maryland, United States – 2015. – URL: https://doi.org/10.1117/12.2178123

3. Aguilar W.G., Angulo C. Real-Time Model-Based Video Stabilization for Microaerial Vehicles // Neural Process Lett 43. – P. 459-477. – 2016. – URL: https://doi.org/10.1007/s11063-015-9439-0

4. Zhou X., Zhang H., Yu R. Decoupling control for two-axis inertially stabilized platform based on an inverse system and internal model control // Mechatronics, Vol. 24, Issue 8. – 2014. – P. 1203-1213. – URL: https://www.sciencedirect.com/science/article/pii/S0957415814001317

5. Канченко В.Я., Карнаушенко Р.В., Ключников О.О., Мариношенко О.П., Чепур М.Л. Безпілотні літальні апарати радіаційної розвідки і сільськогосподарського призначення: монографія // Чорнобиль: Інститут проблем безпеки атомних електростанцій. – 2015. – 180 с. – URL: http://www.ispnpp.kiev.ua/wp-content/uploads/2017/mono/khan-2015.pdf

УДК 004.7

Є. В. Мелешко, Г. М. Дрєєва, О. М. Дрєєв, М. С. Якименко
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

МЕТОД МАРШРУТИЗАЦІЇ ФРАКТАЛОПОДІБНОГО ТРАФІКУ У КОМП'ЮТЕРНИХ МЕРЕЖАХ

У роботі було досліджено основні принципи маршрутизації трафіку у комп'ютерних мережах. Виявлено, що існуючі методи маршрутизації можна покращити на основі використання прогнозування завантаженості маршрутизаторів або ймовірності втрати пакетів [1-2]. Також дослідження показало, що комп'ютерний трафік має фрактальні властивості [3-4], що можна використати при розробці методів прогнозування завантаженості мережевих пристроїв. На основі проведеного дослідження запропоновано удосконалений метод маршрутизації фракталоподібного трафіку з додатковою метрикою – ймовірність втрати IP-пакетів. Запропонований метод маршрутизації використовує прогнозування ймовірності втрати пакетів на основі фрактального аналізу. Також розроблено програмну модель комп'ютерної мережі для тестування запропонованого методу маршрутизації.

Генерація структури комп'ютерної мережі відбувалася на основі теорії складних мереж. Для генерації трафіку використано марківські процеси та фракталоподібні часові ряди. На розробленій програмній імітаційній моделі комп'ютерної мережі було проведено серію експериментів для визначення якості роботи розробленого методу маршрутизації фракталоподібного трафіку.

Фрактальна розмірність для часового ряду змінюється в діапазоні (1, 2), а її значення можна інтерпретувати наступним чином:

– значення менші 1.5 – трафік антиперсистентний – будь-яка тенденція прагне змінитись протилежною, чим менша фрактальна розмірність, тим сильніше проявлена антиперситентність, чим ближче до 1,5 – тим більш випадковий процес.

– значення 1.5 – трафік повністю випадковий.

– значення більші 1.5 – трафік персистентний, тобто зберігає свій тренд, чим більша фрактальна розмірність, тим сильніше зберігається тренд, чим ближче до 1,5 – тим більш випадковий процес.

Аналізуючи результати експерименту можна зробити наступні висновки:

– при великих завантаженнях мережі найменше втрачених пакетів, коли трафік антиперсистентний, найбільше, коли трафік випадковий або персистентний.

– при великих завантаженнях мережі запропонований метод маршрутизації значно зменшує втрату пакетів, при чому значне покращення відбувається при випадковому та персистентному трафіку, а при антиперсистентному трафіку покращення також є, але незначне. Результати експериментів запропонованого методу на програмній моделі показали, що:

– при великих завантаженнях мережі найменше втрачених пакетів, коли трафік антиперсистентний, найбільше, коли трафік випадковий або персистентний.

– при великих завантаженнях мережі запропонований метод маршрутизації значно зменшує втрату пакетів, зокрема, в середньому при антиперсистентному трафіку на 1,03%, при випадковому трафіку на 7,85%, при персистентному на 8,00% порівняно зі стандартним методом маршрутизації на основі алгоритму OSPF. При чому в залежності від параметрів мережі та характеристик трафіку використання запропонованого методу може знизити втрати мережевих пакетів під час пікових навантажень на мережу від 1,01 до 2,57 разів.

Таким чином запропонований метод маршрутизації фракталоподібного трафіку дозволяє зменшити ймовірність втрати мережевих пакетів, що підвищує якість обслуговування у комп'ютерній мережі.

Список літератури

1. Svyrydov A., Kovalenko A., and Kuchuk H., "The pass-through capacity redevelopment method of net critical section based on improvement ON/OFF models of traffic," Advanced Information Systems, vol. 2, no. 2, pp. 139-144, 2018, doi: https://doi.org/10.20998/2522-9052.2018.2.24.

2. Tang F., Fadlullah Z. M., Mao B. and Kato N., "An Intelligent Traffic Load Prediction-Based Adaptive Channel Assignment Algorithm in SDN-IoT: A Deep Learning Approach," in IEEE Internet of Things Journal, vol. 5, no. 6, pp. 5141-5154, Dec. 2018, doi: https://doi.org/10.1109/JIOT.2018.2838574.

3. Molnar S. and Terdik G., "A general fractal model of Internet traffic," Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks, Tampa, FL, USA, 2001, pp. 492-499, doi: https://doi.org/10.1109/LCN.2001.990828.

4. Chakraborty D., Ashir A., Suganuma T., Mansfield Keeni G., Roy T. K., Shiratori N., "Self-similar and fractal nature of Internet traffic," Network Management, vol. 14, no. 2, pp. 119-129, 2004, doi: https://doi.org/10.1002/nem.512. Available: https://onlinelibrary.wiley.com/doi/10.1002/nem.512

УДК 004.9+004.7

В. В. Міхав[1], Є. В. Мелешко[1], А. О. Лавданський[2]
elismeleshko@gmail.com
[1]Центральноукраїнський національний технічний університет, м. Кропивницький
[2]Черкаський державний технологічний університет, м. Черкаси

ДОСЛІДЖЕННЯ ПРИНЦИПІВ РОБОТИ ОДНОРАНГОВИХ ДЕЦЕНТРАЛІЗОВАНИХ СТРУКТУРОВАНИХ КОМП'ЮТЕРНИХ МЕРЕЖ

Метою даної роботи було дослідження принципів побудови однорангових децентралізованих структурованих комп'ютерних мереж та створення списку узагальнених принципів їх роботи.

Як показало дослідження, методи організації P2P мереж можна розділити на наступні види [1-4]:

– Централізовані однорангові мережі, наприклад, BitTorrent [5].

– Неструктуровані децентралізовані однорангові мережі, цей тип мереж використовує алгоритм пошуку Flooding, типовий представник – Gnutella [6].

– Структуровані децентралізовані однорангові мережі, найчастіше будуються на основі розподіле-них хеш-таблиць, наприклад, на основі алгоритму Kademlia [7] або Chord [8].

Було проведено дослідження принципів роботи однорангових децентралізованих структурованих комп'ютерних мереж. Найчастіше для індексації та маршрутизації вони використовують розподілені хеш-таблиці (Distributed hash table, DHT). Для простоти будемо називати далі такі мережі – P2P DHT мережі.

Узагальнені принципи роботи P2P DHT мереж наступні [1-4, 7, 8]:

1. Ідентифікатори комп'ютерів. Комп'ютери мають ідентифікатори h ∈ H, що змінюються у діапазоні [0, 2m-1]. Вони визначається деякою хеш-функцією або залежить від черговості підключення комп'ютерів до мережі. Наприклад, 1-й комп'ютер у мережі отримує ідентифікатор h0=0. і-тий комп'ютер одержує ідентифікатор hi=i або один з вільних ідентифікаторів, що звільнилися внаслідок виходу з мережі деяких комп'ютерів, що раніше були приєднані. Оскільки в децентралізованих мережах комп'ютери можуть приєднуватися та від'єднуватися від мережі, тобто, учасники постійно змінюються, задача виявлення уже невикористовуваних індексів і надання їх новим комп'ютерам є важливою та по-різному вирішується у різних алгоритмах. Як правило, не тільки при приєднанні нового комп'ютера до мережі, а й з деякою періодичністю відбувається перевірка доступності вузлів та оновлення таблиць маршрутизації.

2. Ідентифікатори файлів. Файли також мають ідентифікатори h ∈ H, що змінюються у такому ж діапазоні як і у комп'ютерів [0, 2m-1]. Вони визначаються деякою хеш-функцією. Використовується так зване узгоджене хешування, що на відміну від лінійного передбачає зміну в середньому тільки K/n ключів, де K – число ключів, а n – число слотів. Популярним є застосування алгоритму хешування SHA-1 в P2P DHT мережах, в такому разі m=160, що дозволяє створити досить велику множину ідентифікаторів.

3. Зберігання файлів. Файли чи їх частини (при розподіленому зберіганні) або шляхи до них (при розподіленому індексуванні) зберігаються на комп'ютерах таким чином, щоб ідентифікатор комп'ютера та ідентифікатор файлу (його частини) співпадали. Або, при відсутності у мережі комп'ютера з потрібним ідентифікатором, вони повинні бути максимально близькі з наявних варіантів за деякою метрикою відстані. Надалі для простоти будемо розглядати розподілене зберігання файлів. Для забезпечення надійності зберігання інформації кожен файл дублюють на декілька комп'ютерів, для цього обирається q наявних у мережі комп'ютерів, ідентифікатори яких найближчі за обраною метрикою відстані до ідентифікатора файлу. При зміні складу учасників мережі, за необхідності, відбувається перерозподіл частини файлів між комп'ютерами.

4. Таблиця маршрутизації. Зберігання таблиці маршрутизації також є розподіленим. Кожен комп'ютер містить інформацію потрібну для доступу (мережеві адреси) найближчих N сусідів X = {xi-n, … xi+n} рис. 1. Визначення множини X, а також розмір N цієї множини, у кожному конкретному методі здійснюється по-різному. Наприклад, в алгоритмі Chord [11] таблиця маршрутизації комп'ютера містить 2r+1 записи, тобто інформацію про сам комп'ютер, а також про r комп'ютерів, що мають найближчі менші номери серед наявних у мережі комп'ютерів, та про r комп'ютерів, що мають найближчі більші номери серед наявних комп'ютерів. Комп'ютери об'єднуються в кільце, тобто, останній комп'ютер посилається на перший. Що викликає можливі проблеми з неправильним замиканням кільця, виникненням декількох кілець та розривом кільця при помилках в оновленні таблиць маршрутизації. В алгоритмі Kademlia [10] таблиця маршрутизації зберігається у вигляді так званих K-bucket-тів. У кожному K-bucket-ті вузла зберігається інформація про K вузлів мережі, чия відстань до нього знаходиться в межах інтервалу [2^i, 2^(i+1)), у якості метрики відстані використовується операція XOR. Якщо, наприклад, для створення ідентифікаторів застосовується хеш-функція SHA-1, то кількість K-bucket-тів на кожному вузлі буде 160. K – загальносистемне число, наприклад 20. Кожен K-bucket – це список, що містить не більше K-записів; тобто для мережі з K=20 кожен вузол матиме списки, що містять до 20 вузлів для певної відстані від себе. На практиці виходить, що кожен вузол зберігає інформацію про вузли, з якими будь-коли взаємодіяв, протягом певного часу (наприклад, час життя запиту 24 години). Таким чином розмір таблиці маршрутизації змінюється динамічно, а деякі K-bucket-ти будуть порожніми в деякий момент часу.

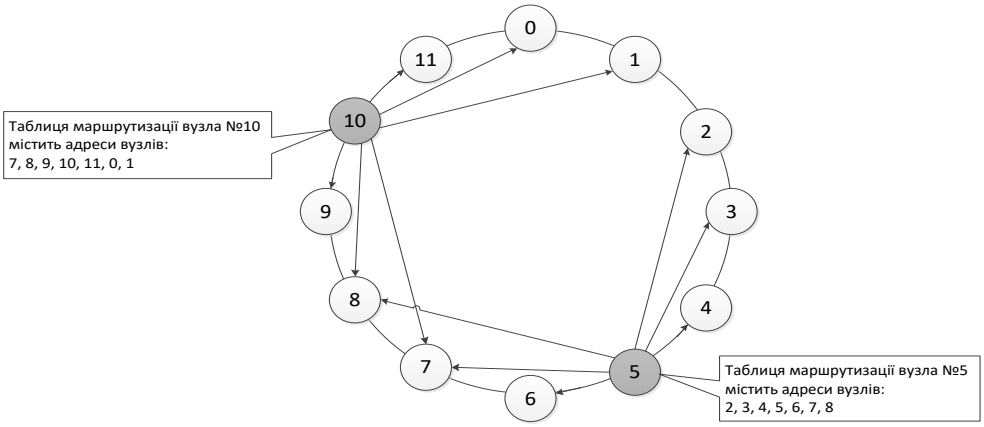


Рис. 1. Приклад принципу наповнення таблиць маршрутизації в однорангових децентралізованих структурованих комп'ютерних мережах: таблиці маршрутизації показані для комп'ютерів №5 та №10 при зберіганні 6 найближчих сусідів у мережі з 12 вузлів за принципами алгоритму Chord

5. Пошук файлів. Файли у мережі шукаються за їх хеш-значеннями. Комп'ютер, який здійснює пошук файлу, вибирає в своїй таблиці маршрутизації j комп'ютерів, ідентифікатори яких найближчі за обраною метрикою відстані до ідентифікатора шуканого файлу. Кожен комп'ютер з множини вибраних S = {s0, s1, … sj} перевіряє чи міститься безпосередньо на ньому шуканий файл, і якщо ні, то пересилає пошукове повідомлення уже j найпідходящим комп'ютерам зі своєї таблиці маршрутизації, і так поки файл не буде знайдений. Чим більше значення s, тим менша ймовірність відмови та вищий рівень інформаційної безпеки, але й вище навантаження на мережу. Можна було б пересилати запит тільки на один комп'ютер з найближчим до шуканого ідентифікатором, але може виявитися, що він на даний час відсутній у мережі. Або ж цим комп'ютером володіє зловмисник, що перенаправить запит не на пошук та завантаження потрібного файлу, а на завантаження вірусу, в той же час при направленні запиту на декілька комп'ютерів можна буде обрати ту відповідь, яка прийде від більшості, що значно зменшить ймовірність інформаційної атаки.

6. Основні запити до мережі. Методи роботи P2P DHT мереж повинні містити також наступні функції:

– Перевірка наявності сусідніх вузлів у мережі для оновлення таблиць маршрутизації;

– Пошук вузлів за ідентифікатором;

– Пошук файлів за ідентифікатором.

Зокрема, в протоколі алгоритму Kademlia для їх реалізації наявні наступні 4 типи запитів: 1) PING – необхідний для перевірки існування конкретного вузла у мережі; 2) STORE дає змогу розмістити інформацію на заданому вузлі; 3) FIND_VALUE – дозволяє знайти значення за ключем; 4) FIND_NODE – використовується для пошуку найближчих K вузлів до заданого ідентифікатора (схожий на FIND_VALUE, тільки ніколи не повертає значення, завжди вузли).

Список літератури

1. Riposo Ju. Diffusion on the Peer-to-Peer Network // LAP LAMBERT Academic Publishing. – 2022. – 100 p.

2. Koo S.G.M. Multimedia Content Distribution Using Peer-to-Peer Overlay Networks: The Design and Analysis of the Next Generation Peer-to-Peer Networks // VDM Verlag Dr. Müller. – 2008. – 88 p.

3. Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D. Information retrieval techniques for peer-to-peer networks // Computing in Science & Engineering, Vol. 6, No. 4, pp. 20-26. – 2004. – DOI: 10.1109/MCSE.2004.12

4. Zeinalipour-Yazti D. Information Retrieval in Peer-to-Peer Systems // M.Sc Thesis, Dept. of Computer Science, University of California Riverside. – 2003. – URL: http://alumni.cs.ucr.edu/~csyiazti/papers/msc/html/

5. The BitTorrent Protocol Specification – 2017. – URL: http://www.bittorrent.org/beps/bep_0003.html

6. Gnutella Protocol Development – 2003. – URL: https://rfc-gnutella.sourceforge.net

7. Kademlia: A Design Specification – 2010. – URL: https://xlattice.sourceforge.net/components/protocol/kademlia/specs.html

8. Stoica I., Morris R., Karger D.R., Kaashoek M.F., Balakrishnan H. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications // ACM SIGCOMM Computer Communication Review, Vol. 31(4). – 2001. DOI: 10.1145/964723.383071

УДК 004.92

О.С. Ткаченко, студент, Є.В. Мелешко, д.т.н., проф.
alex.tranduil@gmail.com, elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ БІБЛІОТЕК МОВИ ПРОГРАМУВАННЯ PYTHON ДЛЯ ВИВЕДЕННЯ ГРАФІКІВ ФУНКЦІЙ

Python – це високорівнева мова програмування, що відрізняється ефективністю, простотою та універсальністю використання. В Python існує безліч бібліотек з різноманітними функціями практично для будь-якої задачі, яка може виникнути у програміста, що є однією з її найбільших переваг порівняно з багатьма іншими мовами програмування.

Метою даної роботи було дослідження та порівняльний аналіз бібліотек мови програмування Python для роботи з комп'ютерною графікою, що дозволяють виводити графіки математичних функцій.

Було проведено дослідження та порівняння наявних переваг та недоліків наступних бібліотек.

1. **Matplotlib** – вважається найпопулярнішою бібліотекою Python для візуалізації даних. Її плюси: чітко відображені властивості даних – при аналізі даних у користувача є можливість швидко переглянути розподіл, що може бути дуже корисно. Мінуси: складно побудувати або підігнати складні графіки, щоб зробити їх більш-менш презентабельними, адже Matplotlib має низькорівневий інтерфейс. Отже за допомогою Matplotlib можна створити будь-що, але для складних графіків може знадобитися набагато більше коду, ніж при використанні інших бібліотек.

2. **Seaborn** – побудована на базі Matplotlib, але більш високорівнева, що полегшує її використання. Плюси: Seaborn зазвичай будує графіки, аналогічні Matplotlib, але з меншою кількістю коду і більш красивим дизайном. Мінуси: більш обмежена і її колекція графіків набагато менша ніж у Matplotlib. Отже, і з цього випливає, що незважаючи на невелику колекцію графіків, створені за допомогою Seaborn графіки при меншій кількості коду виглядатимуть візуально приємніше.

3. **Plotly** – ця бібліотека спрощує створення інтерактивних графіків. Плюси: простота створення інтерактивних графіків – якщо навести на такий графік курсор, то можна його наближати та віддаляти, виділяти ділянки, за наведенням курсору на точку отримувати докладну інформацію, повертати картинку у вихідне положення, а за необхідності «скриншотити» та зберігати як файл; за допомогою Plotly легко робити складні графіки; графіки можна показувати і на сайті (якщо він на Python, або якщо заздалегідь вивантажити все необхідне). Тож, Plotly ідеально підходить для створення інтерактивних та якісних графіків за допомогою всього кількох рядків коду.

4. **Altair** – ця бібліотека ідеально підходить для графіків, що потребують великої кількості статистичних перетворень. Плюси: проста граматика візуалізації; простота перетворення даних; можливість створювати зв'язки між графіками. Мінуси: треба вказувати стиль, бо інакше прості діаграми не будуть оформлені стилістично так само добре як у Seaborn або Plotly; набори даних з більш ніж 5000 екземплярами важко обробляються Altair. Отже, Altair ідеально підходить для створення складних графіків для відображення статистики, але не може обробляти дані, що перевищують 5000 екземплярів, і деякі прості діаграми у ньому поступаються за стилем Plotly і Seaborn.

5. **Bokeh** – це потужна бібліотека з відкритим вихідним кодом, яка дозволяє візуалізувати дані для веб-додатків. Плюси: Bokeh можна використовувати як із високорівневим, так і низькорівневим інтерфейсом; також у Bokeh можна досить легко зв'язувати графіки. Мінуси: має інтерфейс середнього рівня, вимагає більше коду для створення того ж графіка, ніж Seaborn, Altair або Plotly; у Bokeh потрібно вручну налаштовувати параметри, щоб зробити графік презентабельним. Отже Bokeh – єдина бібліотека, інтерфейс якої варіюється від низького до високого, що дозволяє легко створювати як універсальні, так і складні графіки, але для створення якісніших графіків потрібно більше коду, ніж іншим бібліотекам.

Отже, при виборі бібліотеки для побудови графіків у Python, треба трохи почитати особливості цієї бібліотеки, і чітко оцінити, що саме буде демонструвати ваш графік і як ви хочете його зробити, адже будувати простий графік на Plotly, або графік для презентації на Matplotlib буде максимально не зручно, та і результат мабуть не влаштує.

Список літератури

1. Top 6 Python Libraries for Visualization: Which one to Use? – [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/top-6-python-libraries-for-visualization-which-one-to-use-fe43381cd658

2. Matplotlib: Visualization with Python – [Електронний ресурс]. – Режим доступу: https://matplotlib.org/

3. Plotly Open Source Graphing Library for Python – [Електронний ресурс]. – Режим доступу: https://plotly.com/python/

УДК 004.9

Я. П. Шуліка, Є. В. Мелешко, О. К. Конопліцька-Слободенюк
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Часові ряди – це послідовний набір замірів значень деякої змінної, проіндексований в хронологічному порядку. Найчастіше часовий ряд є послідовністю, взятою на рівновіддалених точках в часі, які йдуть одна за одною. Прикладами часових рядів можуть бути: значення температури повітря у певній географічній точці, значення курсу певної валюти, висоти океанських припливів, кількості сонячних плям тощо. Часто виникає задача прогнозування значень часових рядів для систем прийняття рішень.

Метою даної роботи було дослідження можливості застосування нейронних мереж з різною архітектурою для прогнозування значень часових рядів.

Нейронні мережі можуть виявляти складні залежності між минулими і майбутніми значеннями часового ряду і здатні знаходити закономірності та ознаки, що дозволяють виконувати прогнозування [1].

Як показало дослідження, для прогнозування часових рядів, можна використати нейронні мережі з наступною архітектурою:

– Багатошарові нейронні мережі прямого поширення – мережі, у яких з'єднання між вузлами йдуть тільки в одному напрямку – від входу до виходу. Можуть навчатися за допомогою алгоритму зворотного поширення помилки. Можуть використовуватися для прогнозування часових рядів [2].

– Рекурентні нейронні мережі – мають пам'ять і можуть аналізувати послідовності. Показують високу ефективність у прогнозуванні часових рядів з нестаціонарною структурою та динамікою [3].

– Згорткові нейронних мережі – використовують локальний з'єднуючий шар і фільтри для отримання ознак із даних. Знаходять довгострокові залежності в часових рядах та швидко навчаються [4].

– Глибокі рекурентно-згорткові нейронні мережі – це комбінація рекурентних та згорткових мереж яка може моделювати складні нелінійні залежності та знаходити високорівневі ознаки у часових рядах [5].

Різні типи нейронних мереж мають свої переваги і недоліки для прогнозування часових рядів.

Багатошарові нейронні мережі прямого поширення прості і універсальні, але не можуть ефективно моделювати складні нелінійні залежності і не мають довгострокову пам'ять для аналізу часового ряду.

Рекурентні нейронні мережі є потужними засобами для обробки послідовностей даних, але вони складні у побудові і тренуванні і схильні до перенавчання і зникання градієнтів.

Згорткові нейронних мережі ефективні для знаходження закономірностей в сигналах та часових рядах, але вони потребують багато параметрів і обчислювальних ресурсів і не завжди добре підходять для одновимірних часових рядів, адже основне їх призначення – робота з зображеннями.

Глибокі рекурентно-згорткові нейронні мережі є комбінацією рекурентних нейронних мереж і згорткових нейронних мереж, що може поєднати переваги обох типів мереж і подолати їх недоліки. Але такі мережі складні у побудові і тренуванні і потребують налаштування багатьох гіперпараметрів.

Тому при виборі нейронної мережі для прогнозування часових рядів слід враховувати характеристики даних, ціль задачі і наявність обчислювальних можливостей. Також слід проводити експерименти та порівняльний аналіз різних типів нейронних мереж на конкретних даних та оцінювати якість прогнозування за допомогою різних метрик.

Список літератури

1. Чорноиська Ю. Прогнозування за допомогою нейронних мереж, 2010. – URL: https://wiki.tntu.edu.ua/Прогнозування_за_допомогою_нейронних_мереж

2. Руденко О.Г., Безсонов О.О., Романюк О.С. Про один алгоритм навчання нейронної мережі в задачі прогнозування часових рядів, 2018. – URL: http://bionics.nure.ua/article/download/252449/249707/579672

3. Бідюк П.І., Гуць Є.В., Гавриленко В.В., Рудоман Н.В. Прогнозування цін акцій з використанням рекурентної нейронної мережі LSTM // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2021. – Т. 3 (65). – С. 64-68. – URL: http://journals.nupp.edu.ua/sunz/article/view/2390

4. Каштальян А.С., Каштальян О.В. Прогнозування часових рядів розширеною згортковою нейронною мережею // Вісник Хмельницького національного університету, Вип. 6, 2019. – С. 155-160. – URL: http://journals.khnu.km.ua/vestnik/?p=1873

5. Ma M., Mao Z. Deep Recurrent Convolutional Neural Network for Remaining Useful Life Prediction // 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 2019. – pp. 1-4. – URL: https://ieeexplore.ieee.org/document/8819440 https://www.poznavayka.org/uk/nauka-i-tehnika-2/neyronni-merezhi-yih-zastosuvannya-robota/

УДК 004.021

А.М. Токар, ст. 2 курсу
Науковий керівник В.С. Гермак, викладач
tokaram@gmail.com
Центральноукраїнський національний технічний університет,м. Кропивницький

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

Вибір – це те, із чим ми стикаємося щодня. Системи рекомендацій – це набір програмних засобів та методів, покликаних допомогти конкретному користувачеві у здійсненні вибору серед альтернативних об'єктів.

Область застосування рекомендаційних систем дуже різноманітна. Найбільшу роль вони відіграють звичайно для інтернет-магазинів, але також широко використовуються і на сайтах, які виробляють чи публікують контент. Вдало підібрана рекомендаційна система дозволяє скоротити час пошуку потрібного товару чи послуги, утримати користувача на сервісі, підвищити ймовірність здійснення супутніх цільових дій, наприклад, придбання підписки.

На сьогодні виділяють чотири основні підходи до побудови рекомендаційних систем: колаборативна фільтрація; фільтрація заснована на контенті; фільтрація заснована на знаннях; гібридний підхід.

Основна ідея алгоритмів колаборативної фільтрації полягає у побудові рекомендацій для конкретного споживача на основі певних даних про його профіль (уподобання, історію переглядів, оцінок) або оцінки групи однодумців. «Схожістю» користувачів вважається кореляція векторів їх оцінок, яка може розраховуватися багатьма способами. Щоб мати можливість використовувати методи колаборативної фільтрації, потрібно щоб інтереси користувачів були представлені оцінками, які вони виставляють об'єктам після придбання, перегляду тощо. Ніяка додаткова інформація про самих користувачів та оцінювані об'єкти не потрібна. Перевагою методу є універсальність підходу, що зазвичай дає кращий результат, ніж рекомендації на основі контенту. Недоліком є проблема холодного старту, коли відсутні оцінки користувача.

У разі застосування методу фільтрації заснованої на контенті рекомендації формуються, виходячи з "схожості" об'єктів. Алгоритми передбачають наявність інформації про характеристики об'єктів. Перевагою є те, що схожість оцінюється за ознаками вмісту об'єктів. Недоліками є сильна залежність від предметної галузі та обмежена корисність рекомендацій.

Фільтрація заснована на знаннях насправді включає цілий ряд алгоритмів. Під фільтрацією на знаннях розуміють алгоритми, які вибирають рекомендований об'єкт, спираючись на його конкретні властивості. Фільтрацію засновану на знаннях використовують, коли є точна впевненість в ефективності використання певної ознаки або властивості для формування рекомендацій. Даний тип рекомендаційних систем потребує серйозних зусиль для розробки, які зазвичай окупаються ефективністю.

Кожен із названих методів має свої переваги та недоліки. Мінімізувати негативні сторони окремих алгоритмів часто дозволяє використання гібридного підходу, заснованого на поєднанні методів колаборативної та контентної фільтрації. Насправді практично кожна рекомендаційна система представляє композицію кількох методів. Існують різні архітектури гібридних рекомендаційних систем. Так при побудові рекомендаційної системи можуть паралельно використовуватись різні підходи, зокрема незалежно один від одного обчислюються рекомендації використовуючи колаборативну фільтрацію та фільтрацію на контенті. Отримані рекомендації поєднуються за певним правилом і складаються остаточні рекомендації. Такий підхід дозволяє врахувати переваги обох методів. З мінусів можна відзначити збільшені витрати для реалізації та функціонування системи. Також гібридна система може складатися з послідовної комбінації алгоритмів. Наприклад, спочатку здійснюється кластеризація користувачів і потім для кожної групи застосовується алгоритм колаборативної фільтрації. Такий підхід буде доречний при великому обсязі користувачів. На першому етапі в результаті кластеризації будуть утворені групи користувачів. Далі на кожній окремій групі колаборативна фільтрація дасть точніші рекомендації користувачам, ніж на всій множині одразу.

Насправді рекомендаційні системи зазвичай є композицію серії алгоритмів. При цьому вибір алгоритму залишається справою творчою, оскільки безпосередньо залежить від основних задач, які ставляться перед рекомендаційною системою в конкретній галузі. Від задач залежить також вибір критеріїв порівняння та оцінки якості роботи алгоритмів. Для деяких реалізацій немає жодного сенсу гнатися за новизною моделі, але натомість нові моделі можуть допомогти там, де у старих перевірених підходів не вистачає гнучкості для адаптації до поставлених цілей.

УДК 004.021

А.К. Шевченко, ст. 2 курсу
Науковий керівник В.С. Гермак, викладач
shevchenkoak@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

АНАЛІЗ ПРОБЛЕМ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ТА ШЛЯХИ ЇХ ВИРІШЕННЯ

Рекомендаційна система - це невід'ємна частина великих онлайн-платформ, вона допомагає користувачам знаходити релевантний контент у величезному обсязі інформації. Різноманітність таких систем можна проілюструвати основними характеристиками: предмет рекомендації; ціль рекомендації; контекст рекомендації; джерело рекомендації; ступінь персоналізації; формат рекомендації; прозорість рекомендації. У центрі побудови таких систем лежить матриця вподобань. У цій матриці одна із осей відповідає користувачам, друга об'єктам рекомендації. Складність побудови моделей рекомендаційних систем полягає у необхідності одночасно задовольняти різним критеріям якості, а також у складності підготовки та інтерпретації наявних даних.

Основні проблеми, що зустрічаються при побудові рекомендацій: різноманітність типів зворотної реакції (тривалість взаємодії, вподобайки, кліки, додавання до обраного) та її неоднозначність (Клік, але не вподобайка і не покупка це що? Подобається результат користувачеві чи ні?); різноманітність типів контенту (наприклад відео та зображення, для яких тривалість взаємодії принципово не можна порівнювати); холодний старт користувача; холодний старт контенту; короткостроковість трендів та старіння контенту; глобальна популярність; масштабованість, швидкість видачі рекомендацій, навантаження на сервери; оновлення рекомендацій (швидке реагування на останні дії, новий контент); регулярне оновлення моделей; облік послідовності взаємодії (смаки змінюються); облік сезонності (на Новий рік та на 14 лютого у тренді будуть різні фільтри); включення різних типів інформації (щоб враховувати не тільки взаємодії, а й сам контент - зображення та текст, а також метадані користувача - геолокацію, платоспроможність та інше).

Одна з найскладніших – це проблема холодного старту, коли потрібно передбачати, що сподобається користувачам, які тільки прийшли на платформу і не спожили жодного контенту, не ставили оцінок, або рекомендувати товари чи послуги, які люди ще не бачили і нічого про них не знають. Один із методів вирішення цієї проблеми це отримання інформації про користувачів чи продукти з інших джерел: дані з інших, подібних, сервісів; опитування при реєстрації (в обмін на якусь вигоду, наприклад); інформація про товар із його змісту.

Ще одна важлива проблема полягає у тому, що, попри всю схожість задач у сфері побудови рекомендацій, в кожного сервісу є своя специфіка. На музичних сервісах композиції можуть слухати багаторазово, а фільм, навіть найцікавіший, мало хто буде дивитися двічі. У Tiktok мільйони роликів і вони старіють з жахливою швидкістю. Рекламні рекомендації у соціальній мережі потрібно оптимізувати з урахуванням геотаргетингу (навіть якщо ви дуже любите суші, і це справді крутий ресторан, то ви все одно не замовите доставку з Києва до Кропивницького). А в Airbnb необхідно оптимізувати не тільки вдоволення туристів від хостів, а й хостів від туристів (тобто реалізувати двобічну оптимізацію).

Важливо точно визначати критичні точки рекомендаційної системи. Однією з таких критичних точок є метрика якості системи. В ідеалі ми хотіли б прямо виміряти, наскільки користувача влаштовує сервіс, але це неможливо. Доводиться задовольнятися опосередкованими метриками, деякі з них можуть конфліктувати. У багатьох навіть простих метрик є свої підводні камені. Так, орієнтування на кількість кліків підвищує рейтинг рекомендацій сміттєвого контенту і не працює для стрічок, які просто читають гортаючи, без кліків. Якщо говорити безпосередньо про бізнес-метрики (ретеншн, час на сесію, кількість дій за сесію, середня активність за період часу), то з ними також все неоднозначно. Одна метрика, зростаючи, може тягнути іншу на дно. До того ж, використовувати їх у моделях безпосередньо практично неможливо. Одним із методів вирішення проблем при розробці рекомендаційних систем є комбінування вхідних даних. Існує два основні види вхідних даних для рекомендацій - це колаборативні особливості (будь-які дані про взаємодію користувачів та об'єктів) та метадані об'єктів або сам контент. Колаборативні особливості стають потужною основою для рекомендацій, а метадані дозволяють зробити рекомендації точнішими (до того ж допомагають при холодному старті).

Таким чином, завдяки ряду вдосконалень, багато з перевірених часом алгоритмів досі прекрасно функціонують. Проте по мірі збільшення обсягів даних та ускладнення задач повсякчас з'являються нові підходи, відмінні від простих колаборативних моделей, які дозволяють оптимізувати процес побудови рекомендацій.

СЕКЦІЯ 3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ, МЕДИЦИНІ ТА ОСВІТІ

УДК 004.388.4 004.413 004.415.2

Я.Я. Григорчук, студент групи КМ-21,
Науковий керівник О.А. Кислун, доцент ,
grugorkcuk@gmail.com, kyslun@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

РОЗРОБКА ГРИ, ЯК МОЖЛИВИЙ СТАРТАП СТУДЕНТА-АЙТІШНИКА

Перед кожним фахівцем-початківцем постає питання з працевлаштування, можна знайти роботу в існуючій компанії або взяти долю у власні руки та спробувати створити власну компанію. Такий шлях не обходить IT сферу. Одним із найпростіших способів є стартап ігрової індустрії. Зазначена діяльність являє собою створення гри та її комерційне розповсюдження (продаж).

Отже, реалізація стартапу, передбачає наступні етапи; створення ідеї гри; її реалізація безпосередньо в програмному додатку; комерційне розповсюдження розробленого продукту з урахуванням комплексу заходів, спрямованих на підтримку розробки, та її захист від несанкціонованого використання; проектування подальших перспектив використання розробки, як в цілому так, і частково отриманих рішень, шляхом модифікацій, створення продовжень, реалізації (комерціалізації) об'єктів авторського права (прикладом може бути використання персонажу в рекламних цілях) тощо.

Починається процес розробки зі сценарію гри. Для цього обирається жанр, описується ідея гри та її кінцева мета, опис складових елементів гри (наприклад: для бродилки це час, місце, герої, звукові схеми з їх описами), лінії розвитку гри, їх опис, якщо є їх перетин/збіг з їх усіма характеристиками та динамікою розвитку в процесі гри (наприкладі бродилки, що розглядається, це схеми локацій, їх вигляди із звуковим супроводом), фіналізація (по можливості комплектується подальшим розвитком гри на створення нового релізу гри).

Для створення гри обираються інструменти розробки. Зазвичай використовують такі програми створення: Game Maker (програма-конструктор для створення 2D і 3D ігор, що не вимагає ніяких спеціальних знань в області програмування), Unity 3D (ігровий рушій, що, використовуючи інтерфейс візуального програмування для створення ігр та для більш масштабних проектів, може використовувати такі моив програмування, як JavaScript або C#), Clickteam Fusion (програма-конструктор для створення 2D 2D ігор за допомогою інтерфейсу drag'n'drop, що має режим тестування, в якому наявний режим гри можна перевірити на помилки), Construct 2 (заснований на HTML5 конструктор 2D ігор, дозволяє швидко створювати ігри способом Drag-and-drop з використанням візуального редактора та логічної системи, заснованої на принципі поведінки та реакції, та має, крім стандартних наборів плагінів, поводжень та візуальних ефектів, можливість написання плагінів поведінки і ефекти мовою JavaScript), CryEngine (ігровий рушій для створення тривимірних ігор, що має дуже великий набір інструментів для розробки ігор із створенням ескізів моделей і алокацій), Game Editor (дещо спрощений конструктор ігор,для створення ігор і на комп'ютери, і на телефони), Unreal Development Kit. (ігровий рушій для розробки 3D ігор для різноманітних платформ, що дає можливість створювати ігри, не використовуючи мов програмування, а просто задаючи готові події об'єктів). 3D Rad (безкоштовна програмаа, навіть для комерційного використання, для створення 3D ігор на комп'ютер, в якій можна створити мультиплеєрну гру або ж гру по мережі і навіть налаштувати ігровий чат), та інші [1]

Реалізація сценарію гри в залежності від вибраних інструментів та уподобань розробника може включати програмування або ні, проте обов'язково має включати захист від несанкціонованого використання. Сама ж реалізація захисту повинна передбачати можливість розповсюдження продукту, тобто легалізації несанкціонованих клонів.

Програмна продукція вимагає реалізації на ринку. Основними нормативно-правовими актами, які регулюють дане питання, є Цивільний кодекс, закони України «Про авторське право і суміжні права» та «Про розповсюдження примірників аудіовізуальних творів, фонограм, відеограм, комп'ютерних програм, баз даних». Не вдаючись до розгляду правових норм, зазначимо, що комерційне поширення комп'ютерної програми через інтернет автором або правовласником не суперечить законодавству України, тож реалізацію ігор, як програмної продукції, обмежено лише розглядом самого комерційного поширення, як способу. Оскільки, вибраний спосіб комерційного поширення комп'ютерної програми є поширення через інтернет, то наявні можливі варіанти: скористатися послугами спеціалізованих сайтів, які розміщують нові ігри чи програми за винагороду; скористатися власним сайтом, де є платний доступ до розробки; скористатися умовно безкоштовною копією з вільним доступом на певний проміжок часу або безкоштовною копією з обмеженим функціоналом, а після цього отримати винагороду [2]..

Підсумовуючи, можна зазначити, що розробка ігор може бути однією з можливостей власного стартапу студента-айтішника.

УДК 004.056, 004.75

І.А. Лисенко
min_max@i.ua
Центральноукраїнський національний технічний університет, м. Кропивницький

ВИКОРИСТАННЯ ТАБЛИЦЬ РІШЕНЬ ДЛЯ ВДОСКОНАЛЕННЯ РОБОТИ ЕЛЕКТРОННОЇ БАНКІВСЬКОЇ СИСТЕМИ

Для підвищення ефективності роботи банківської системи, збільшення її швидкості та якості обробки запитів, зменшення кількості помилок під час звернень до системи з боку користувачів, необхідно провести якомога ширший та ґрунтовний аналіз роботи системи. Зробити це можливо за допомогою таблиць рішень, що використовуються для проектування тестових наборів.

Банк замовляє розробку ПЗ HomeDirect, що має забезпечити клієнтам банка повний набір електронних банківських послуг. Клієнтський додаток HomeDirect повинен бути доступний всім клієнтам через Internet і його ПЗ повинно забезпечити реалізацію трьох основних типів послуг [1]:

- довідка, що має містити оновлення залишку на рахунку, перегляд списку операцій та завантаження протоколу операцій з рахунком;
- послуги сплати, що забезпечують можливість користувачам сплачувати рахунки через Internet;
- операційні послуги, що міститимуть такі функції як перерахування коштів, а для інвестиційних рахунків – можливість купувати та продавати акції.

При цьому необхідно створити окремий механізм для функцій адміністрування HomeDirect. Крім повного доступу до електронних банківських функцій, відповідно перерахованому вище набору послуг, функції адміністрування повинні дозволяти створювати нові рахунки, закривати рахунки, дозволяти проводити операції окремому клієнту, вручну коректувати залишок на рахунку і, за вимогою власника рахунку, відміняти майбутні платіжні операції.

Електронні банківські послуги HomeDirect мають працювати на основі існуючих баз даних та іншої банківської інфраструктури розподіленої електронної банківської системи. Зокрема, має бути передбачена можлива інтеграція з існуючою кредитною системою LoansDirect для запиту клієнтами надання кредитів через Internet, а також з системою BillsDirect для відстеження неоплачених рахунків.

З використанням процедури перевірки збитковості і протиріччя таблиць рішень створених для перевірки роботи відповідної банківської системи, що використовуються для проектування тестових наборів, може бути виявлено виконання або невиконання необхідної умови збитковості та протиріччя, достатня умова може не перевірятись.

Таблиця рішень повинна відображати такі сценарії входу в систему: «Головний сценарій випадку використання «Переведення коштів», «Головний сценарій випадку використання «Вхід в систему», «Сценарій випадку використання «Вихід із системи» у зв'язку з тим, що у відповідності до їх матриць масок і рішень у таблиці рішень відсутні неортогональні правила.

Використовуючи процедуру перевірки повноти таблиці рішень «Головний сценарій випадку використання «Переведення коштів», «Головний сценарій випадку використання «Вхід в систему», «Сценарій випадку використання «Вихід із системи», можна зробити висновок про відсутність неврахованих тестових випадків, виходячи з рівності

$$\sum_{i=1}^{k} 2^{P_i} = 2^n,$$

де k - кількість правил у таблиці рішень,

i - кількість умов у таблиці рішень,

P_i - ступінь узагальненості ситуації, описаної у i-му правилі, при цьому P_i відрізняються один від одного хоча б однією умовою.

Таким чином коректність матриць слідування розроблених для таблиці рішень, підтверджується відсутністю у них діагональних одиничних елементів, а також одиничних елементів, симетричних відносно основної діагоналі, при цьому для всіх матриць слідування та відповідних їм таблиць рішень має виконуватись умова їхньої сумісності.

Список літератури

1. Галіцин В.К., Сидоренко Ю.Т., Потапенко С.Д. Технологія програмування і створення програмних продуктів. Київ, Україна: КНЕУ, 2009.

УДК 796:004.38

О.А. Павлюк, 3 курс
Науковий керівник: Ю.І. Хлапонін, д.т.н., професор,
завідувач кафедри кібербезпеки та комп'ютерної інженерії
alexpavluk10@gmail.com
Київський національний університет будівництва і архітектури, м. Київ

ТЕНДЕНЦІЇ РОЗВИТКУ СУЧАСНОГО КІБЕРСПОРТУ

Кіберспорт як явище спортивних змагань набув популярність разом зі зростом онлайн трансляцій в 2000-х роках. Це були ефіри професійних та звичайних гравців таких популярних ігор як Starcraft, League of Legends, DOTA or Call of Duty. Тоді ці події привертали увагу декількох тисяч людей, але з кожним роком кількість глядачів збільшувалась[1].

Порівнюючи з початком ери, сучасний кіберспорт далеко просунувся за кількістю глядачів. Найбільші турніри останніх 5 років збирали мільйонні авдиторії біля екранів(рис.1).

Рис. 1 – Кількість глядачів найбільших турнірів за останні 5 років[2]

Із набуттям популярності мобільних ігор в постковідний період, вони теж зайняли своє місце в кіберспортивній галузі. Наприклад, PUBG Mobile, Free Fire та Mobile Legends: Bang Bang своїми турнірами 2020-2022 років привертали увагу більше двох мільйонів глядачів[2].

Разом із популярністю кіберспортивних події зростав і обіг ринку кіберспорту. Кіберспортивний ринок за 6 років представлений на рисунку 2.

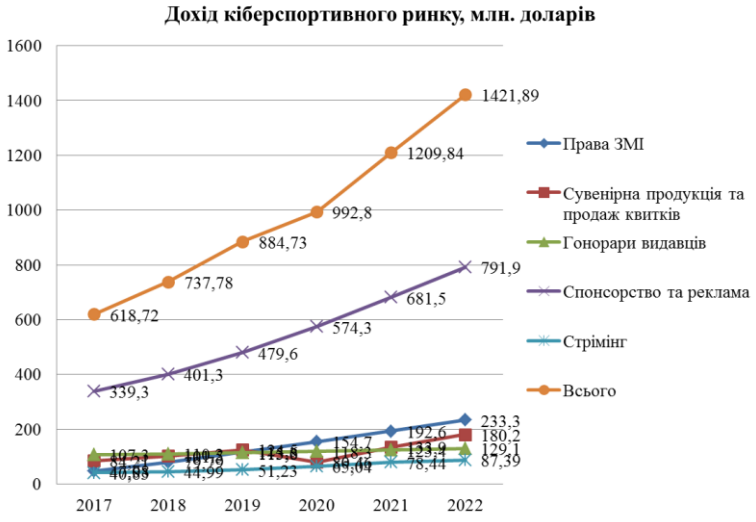


Рис. 2 – Дохід кіберспортивного ринку[3]

Також стрімко зростали призові фонди. Рекордсменами є The International 10 of Dota 2 2021 року (40 мільйонів доларів США), The International 2019 (34,34 мільйонів доларів США), Fortnite World Cup 2019 Finals

(30 мільйонів доларів США). Рекордсмен за середньою кількістю глядачів League of Legends має постійні призові у розмірі 2,250 млн. доларів США для чемпіонатів світу[2].

Український кіберспорт розпочався з перемоги команди гравців в Dota 2 Natus Vincere в чемпіонаті The International 2011 року (1 млн. доларів США за перше місце). Хоча подібних успіхів більше не було, це дало надхнення українцям розпочати свій шлях в цій сфері.

Сучасний кіберспорт в Україні популяризує і розвиває Федерація кіберспорту України, що була створена в 2017 році, а визнана національною – 2018. З тих часів федерація проводить турніри для української молоді.

Україна займає дванадцяте місце за кількістю виграних призових(рис. 3).

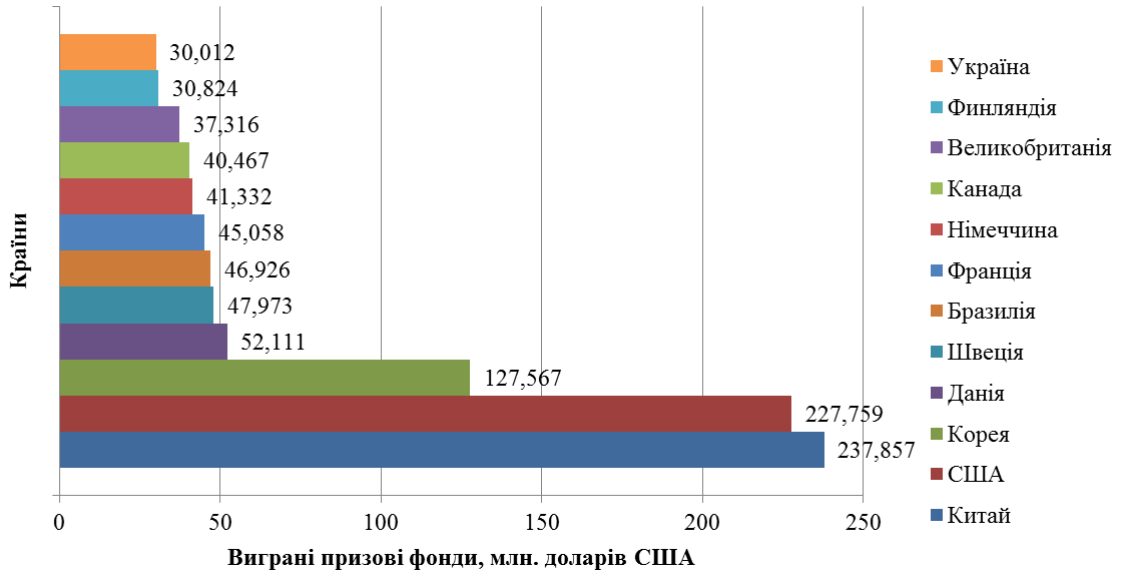


Рис. 3 – Виграні призові фонди[4]

Крім того, багато вихідців з України є членами успішних команд по всьому світові. Наприклад, Михайло Гармаш «Kira», Дмитро Аненко «Anechek», Олександр Севостьянов «Unho1y» – гравці в League of Legends; Володимир «No[o]ne» Міненко, Роман «Resolut1on» Фомінок – гравці в Dota 2.

Висновки

Аналізуючи зріст числа глядачів, призових фондів та ринкового обігу в кіберспорті, можна стверджувати, що ця галузь інформаційного світу має яскраве та успішне майбутнє. Крім того, в Україні існує організація, яка підтримує та допомагає молодим вітчизняним кіберспортсменам, що свідчить про перспективи розвитку кіберспорту в країні. Разом з державними інвестиціями це дозволить українським гравцям займати більш активну позицію на світовій арені та розвивати цю галузь в Україні.

Список літератури

1. Paul Tassi 2012: The Year of eSports [Електронний ресурс] / Paul Tassi – Режим доступу: https://www.forbes.com/sites/insertcoin/2012/12/20/2012-the-year-of-esports/?sh=478c35b97e11

2. Esports Charts [Електронний ресурс] – Режим доступу: https://escharts.com/

3. Advertising & Media Market Insights eSports – Worldwide [Електронний ресурс] / ESM.ONE, Inc. – Режим доступу: https://www.statista.com/outlook/amo/esports/worldwide

4. Highest Earnings By Country [Електронний ресурс] / Esports Earnings – Режим доступу: https://www.esportsearnings.com/countries

5. League of Legends Esports Wiki [Електронний ресурс] / FANDOM Games Community – Режим доступу: https://lol.fandom.com/wiki/League_of_Legends_Esports_Wiki

УДК 004.8

К.О. Довгенко, студент групи КМ-21,
Науковий керівник: О.А. Кислун, доцент ,
ofigenno235@gmail.com, kyslun@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ШТУЧНИЙ ІНТЕЛЕКТ ТА ЙОГО РОЛЬ У СУЧАСНОМУ СВІТІ

Робочі процеси стають все більш об'ємними та потребують все більше людиногодин. Отже, зайнятість людей зростає, спеціалістів та часу не вистачає. Вирішенням проблеми є штучний інтелект (ШІ).

Головна риса ШІ - постійна можливість навчання, збір інформації та успішне застосування її на практиці, що дозволяє досягати таких же результатів, які можна отримати від людського мислення. Штучний інтелект здатний думати і реалізовувати завдання, від найпростіших до найскладніших, таких як: готування коктейлів, керування транспортом або ставлення діагнозів людям. [1, 5]

Серед основних технологій штучного інтелекту виділяються три напрями: машинне навчання, нейронні мережі та генетичні алгоритми.

Машинне навчання - це інноваційна технологія, що дозволяє комп'ютерам самостійно вчитися на основі інформації та удосконалювати свої алгоритми без явного програмування. Вона використовується для часткової або повної автоматизації різних складних аналітичних процесів. [4]

Нейронні мережі є комп'ютерними системами, що наслідують функціонування людського мозку. Вони складаються з численних штучних нейронів, які з'єднані між собою та спільно працюють для вирішення різноманітних задач. [3]

Для вирішення задач оптимізації та пошуку рішень застосовуються генетичні алгоритми. Базуючись на імітації природного еволюційного процесу, де кращі рішення "виживають" та "розмножуються", генетичні алгоритми створюють нові та кращі рішення. За допомогою генетичних алгоритмів створюється популяція випадкових рішень, які є потенційними рішеннями задачі, проводиться відбір найкращих рішень, які залишаються для наступної ітеграції. Також здійснюється мутація, тобто невеликі випадкові зміни в кращих рішеннях з метою створення ще кращих. [2]

Беззаперечно ШІ, як технологія, надає значні переваги її користувачам, проте існують і недоліки її застосування, а саме:

- наявність потенційних помилок, яка обумовлена залежністю штучного інтелекту від навчальних даних, які, в свою чергу, можуть бути не достатньо представницькими або належно структурованими, що може призвести до некоректної роботи штучного інтелекту;
- проблеми з використанням штучного інтелекту в злочинних цілях, виникають при використанні ШІ шахраями, шпигунами, та ін. особами, для досягнення особистих цілей, що можуть призвести до великих проблем;
- заміна робочих місць, що в свою чергу, може призвести до безробіття та економічних проблем. [7]

Штучний інтелект має безліч застосувань, і його роль у різних галузях продовжує зростати. Шляхом поліпшення ефективності та точності багатьох процесів, штучний інтелект робить їх більш автоматизованими та ефективними. Наприклад, у медицині, використання штучного інтелекту може допомогти у виявленні різних захворювань та розробці більш ефективних методів лікування. У фінансах, використання ШІ може допомогти зменшити ризики та витрати, покращити роботу банківських та фінансових установ. У виробничій сфері, використання штучного інтелекту може допомогти підвищити продуктивність та ефективність виробничих процесів, контролюючи якість виробів, зменшуючи відходи та покращуючи безпеку на робочому місці. [6]

Оскільки, штучний інтелект все більше і більше інтегрується в різні сфери людського життя, то відповідно постануть завдання розробки етичних та юридичних норм для регулювання використання штучного інтелекту, а також навчання нових спеціалістів професійним навичкам, які стануть актуальними у світлі автоматизації та зміни ролі людини у виробничому процесі.

Список використаних джерел

1. https://aiconference.com.ua/uk/news/printsipi-raboti-iskusstvennogo-intellekta-i-perspektiva-ego-ispolzovaniya-92238
2. http://free-review.net.ua/shtuchnyj-intelekt-genetychnyj-algorytm-i-jogo-zastosuvannya/
3. https://www.poznavayka.org/uk/nauka-i-tehnika-2/neyronni-merezhi-yih-zastosuvannya-robota/
4. https://www.ibm.com/topics/machine-learning
5. https://www.ibm.com/cloud/learn/what-is-artificial-intelligence
6. https://aiconference.com.ua/uk/news/oblasti-primeneniya-iskusstvennogo-intellekta-92253
7. https://sashave.medium.com/про-загрози-і-ризики-розвитку-штучного-інтелекту-f44067001913

УДК 004.9

Рисований М. бакалаврант
maximofficial@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ПЕРЕВАГИ ТА НЕДОЛІКИ РОЗРОБКИ ІГОР ТА ПЗ НА UNITY

Unity – це міжплатформове середовище розробки ПО та комп`ютерних ігор. Двигун Unity дозволяє створювати програми, що працюють під більш ніж 20 різними операційними системами, що включають персональні комп'ютери, ігрові консолі, мобільні пристрої, інтернет-програми та інші. Випуск рушія відбувся у 2005 році і з того моменту триває постійний його розвиток. На ньому написано більш ніж тисяча ігор, програм та застосунків, які охоплюють велику кількість жанрів. Варто зауважити, що Unity використовується не тільки великими компаніями, а також і маленькими незалежними студіями.

Основною перевагою двигуна є кросплатформовість: Unity дозволяє розробляти ігри та ПЗ для різних платформ, таких як Windows, MacOS, Linux, Android, iOS, Xbox, PlayStation та інших. Це значно зменшує витрати на розробку та підтримку продукту.

Наступною перевагою є наявність візуального середовища розробки. Цей фактор включає в себе інструментарій візуального моделювання, інтегроване середовище, ланцюжок складання та розробки, а також створення прототипів застосунку та його тестування. Ці складові допомагають розробнику облегшити його роботу та виконати її належним чином.

Велика спільнота користувачів: Unity має велику та активну спільноту користувачів, що надає безкоштовну та платну допомогу вирішення проблем та навчання розробки на цій платформі.

Швидкість розробки: Unity надає широкий набір інструментів, що дозволяє розробникам швидко створювати та тестувати ігри та ПЗ. Багато процесів, таких як створення графіки та фізики, можуть бути автоматизовані, що прискорює процес розробки.

Наступна перевага – це мова програмування C#. Більшість розробників використовує саме цю мову для написання застосунків. Ця мова є високорівневою і дозволяє програмісту легко увійти в розробку ігор. Також можна використовувати мову JavaScript, але вона є менш зручною у даному випадку. У середовище Unity також є чудовий інтегрований редактор рівнів із підтримкою цих двох мов програмування.

Як першим недоліком можна виділити що розробка більш складної гри або програми може обійтися дорого, тому що базових безкоштовних бібліотек ніколи не вистачить, тому варто пам`ятати про бюджет при розробці більш великих застосунків.

Другим недоліком є відсутність підтримки Unity посилань на зовнішні бібліотеки, через роботу з якими програмістам доводиться налаштовувати власноруч, в результаті це може ускладнювати розробку в групах. Також Unity-проекти забирають велику кількість пам'яті, це може спричинити помилки на мобільних пристроях та в подальшому проблеми з налагодженням та оптимізацією, а розробка великої AAA-гри потребує значної кількості оптимізації.

Вартість: Незважаючи на те, що Unity безкоштовна, вона має платні версії з більшим набором інструментів та функцій. Крім того, підтримка Unity може виявитись досить дорогою.

Отже, Unity є повноцінним середовищем розробки, яке об'єднує різні програмні інструменти, що робить процес розробки дуже простим та комфортним. Незважаючи на те, що вона має деякі недоліки, вона все ще є привабливим варіантом для розробників, оскільки дозволяє створювати мультиплатформні ігри та ПЗ. Це робить Unity одним з найкращих варіантів для початківців, які збираються розпочати свій шлях у світі розробки ігор.

Список літератури

1. Mike Geig, Unity Game Development in 24 Hours, Sams Teach Yourself, 2018
2. Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#, 2017.
3. Unity Game Development Cookbook: Essentials for Every Game, 2017.

УДК 004.94

О.В. Рудський, А.М. Копп
oleksandr.rudskyi@cs.khpi.edu.ua, andrii.kopp@khpi.edu.ua
Національний технічний університет «Харківський політехнічний інститут», м. Харків

ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ОБРОБКИ ПРИРОДНОЇ МОВИ ДЛЯ АНАЛІЗУ ВІДПОВІДНОСТІ МОДЕЛЕЙ БІЗНЕС-ПРОЦЕСІВ ЇХ ТЕКСТОВИМ ОПИСАМ

Моделі бізнес-процесів зарекомендували себе як ефективний засіб візуалізації та вдосконалення складних організаційних операцій. Однак створення моделей бізнес-процесів є трудомісткою справою, яка потребує значних ресурсів, тому можуть виникати ситуації в яких модель бізнес-процесу не відповідає її текстовому опису. Це може призвести до втрат у часі та значних грошових втрат. Таким чином, актуальною є задача аналізу відповідності моделей бізнес-процесів їх текстовим описам.

Для того, щоб обробляти тексти з метою аналізу відповідності моделей бізнес-процесів їх текстовим описам, пропонується застосовувати наступні технології обробки природної мови або NLP-технології (Natural Language Processing):

– токенізація;
– пошук стоп-слів;
– стемінг.

Токенізація – це перший крок у будь-якому процесі NLP. Токенізатор розбиває неструктуровані дані та текст природною мовою на фрагменти інформації, які можна розглядати як дискретні елементи. Це відразу перетворює неструктурований рядок (текстовий документ) на числову структуру даних, придатну для машинного навчання [1].

Після розділення тексту на токени, часто стає ясно, що не всі слова несуть однакову кількість інформації, якщо взагалі будь-яку інформацію для завдання прогнозного моделювання. Загальні слова, які несуть мало значної інформації, називаються стоп-словами. Стоп-слова – це слова будь-якою мовою, які не додають великого сенсу реченню. Їх можна сміливо ігнорувати, не жертвуючи змістом речення. Для деяких пошукових систем це деякі з найпоширеніших, коротких функціональних слів, таких як "the", "is", "at", "which" та "on" [2].

Стемінг є однією з найпоширеніших операцій попередньої обробки даних, яка виконується майже у всіх проектах обробки природної мови (NLP). Стемінг – це процес скорочення слова до основи шляхом відкидання допоміжних частин, таких як закінчення чи суфікс. Результати стемінгу іноді дуже схожі на визначення кореня слова, але його алгоритми базуються на інших принципах. Тому слово після обробки алгоритмом стемінгу може відрізнятися від морфологічного кореня слова [3].

Таким чином, для розв'язання задачі аналізу відповідності моделей бізнес-процесів їх текстовим описам з отриманими текстами T_1 та T_2 необхідно виконати наступні дії (рис. 1) [4]:

1) розбити отримані на вхід тексти T_1 та T_2 на окремі слова (токенізувати), отримавши відповідні мультимножини слів:

$$(W_1, m_1) = \{(t_i^1, m_1(t_i^1), t_i^1 \in W_1 \wedge i = \overline{1,n}\},$$
$$(W_2, m_2) = \{(t_j^2, m_2(t_j^2), t_j^2 \in W_2 \wedge j = \overline{1,q}\},$$

де W_1 – множина слів, отриманих в результаті токенізації тексту T_1; W_2 – множина слів, отриманих в результаті токенізації тексту T_2; $t_i^1 \in W_1$, $i = \overline{1,n}$ – слово, отримане в результаті токенізації тексту T_1; $t_j^2 \in W_2$, $j = \overline{1,q}$ – слово, отримане в результаті токенізації тексту T_2; $m_1(t_i^1)$ – відображення $m_1 : W_1 \to \{1,2,3,...\}$, яке для кожного слова $t_i^1 \in W_1$, $i = \overline{1,n}$ встановлює кількість його повторювань в тексті T_1; $m_2(t_j^2)$ – відображення $m_2 : W_2 \to \{1,2,3,...\}$, яке для кожного слова $t_j^2 \in W_2$, $j = \overline{1,q}$ встановлює кількість його повторювань в тексті T_2; n – кількість слів, отриманих в результаті токенізації тексту T_1; q – кількість слів, отриманих в результаті токенізації тексту T_2;

2) видалити стоп-слова з множин W_1 та W_2, отримавши множини лише змістовних термінів, що стосуються предметної області бізнес-процесу:

$$stop : \{W_k, k = \overline{1,r}\} \to \{W_k', k = \overline{1,r}\},$$

де W_k, $k = \overline{1,r}$ – множина слів, отримана в результаті токенізації вихідного тексту, що також містить стоп-слова; W_k', $k = \overline{1,r}$ – множина слів, отримана в результаті токенізації вихідного тексту, з якої були

видалені стоп-слова; *stop* – відображення, яке для кожної множини W_k, $k=\overline{1,r}$, яка містить стоп-слова, ставить у відповідність множину W_k', $k=\overline{1,r}$, яка не містить стоп-слова; r – кількість множин слів, що обробляються, $r=2$;

3) виконати стемінг слів у множинах W_1' та W_2', що залишились після видалення стоп-слів:

$$stemm:\{W_k', k=\overline{1,r}\}\to\{W_k'', k=\overline{1,r}\},$$

де W_k'', $k=\overline{1,r}$ – множина слів, отримана в результаті стемінгу слів, що залишились після видалення стоп-слів; $stemm$ – відображення, яке для кожної множини W_k', $k=\overline{1,r}$, з якої були видалені стоп-слова, ставить у відповідність множину W_k'', $k=\overline{1,r}$, слова в якій, що залишились після видалення стоп-слів, були скорочені до основи.

Таким чином, в результаті виконання попередній дій, будуть отримані множини слів W_1'' та W_2'':

$$W_1''\cup W_2''\subseteq\{W_k'', k=\overline{1,r}\}.$$

Обчислити подібність цих двох множин слів W_1' та W_2' можна за допомогою коефіцієнта Жаккара:

$$K_J=\frac{|W_1''\cap W_2''|}{|W_1''|+|W_2''|-|W_1''\cap W_2''|}=\frac{|W_1''\cap W_2''|}{||W_1''\cup W_2''|}.$$

Відповідно, отримане значення коефіцієнту Жаккара можна інтерпретувати як ступінь відповідності моделі бізнес-процесу її текстовому опису.

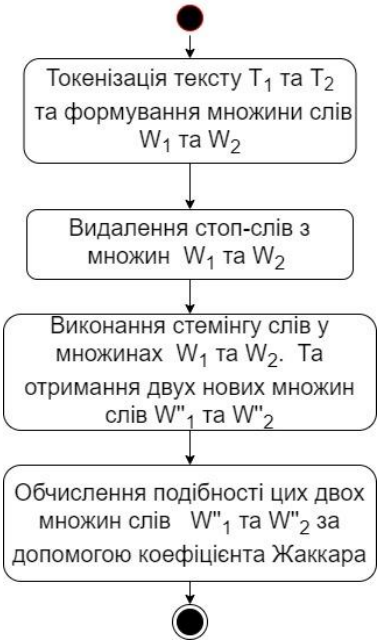


Рис. 1. Алгоритм розв'язання задачі аналізу відповідності моделей бізнес-процесів їх текстовим описам

Список літератури

1. Tokenization in NLP: Types, Challenges, Examples, Tools. URL: https://neptune.ai/blog/tokenization-in-nlp (дата звернення: 10.03.2023).

2. Rajaraman A., Ullman J. D. Mining of Massive Datasets. *Cambridge University Press*. Cambridge, 2011. P. 18–19.

3. Jongejan B., Dalianis H. Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. Singapore, 2009. P. 145–153.

4. Kopp A. M., Orlovskyi D. L. The approach and the software tool to calculate semantic quality measures of business process models. *Bulletin of the National Technical University "KhPI" : System analysis, control and information technology*. 2022. No. 1 (7). P. 66–69.

УДК 519.683.8

Є.А. Деркач, магістрант, О.В. Коваленко, д.т.н., доцент,
derkachea@gmai.com ,clashav@gmail.com
Центральноукраїнський національний технічний університет,м. Кропивницький

АКТУАЛЬНІСТЬ ВИВЧЕННЯ ПІДХОДІВ REVERSE ENGINEERING ЗДОБУВАЧАМИ ВИЩОЇ ОСВІТИ ІТ–СПРЯМОВАНОСТІ

Вивчення технологій зворотного проектування (reverse engineering) є актуальна тема [1-2] для здобувачів освіти закладів вищої освіти ІТ–спрямованості, особливо напрямів кіберзахисту. Зворотне проектування – це процес розбирання та аналізу програмної системи, щоб зрозуміти, як вона працює без допомоги оригінальних проектних документів та вихідних кодів програм. Розглянемо найактуальніші причини та існуючі підходи зворотного проектування для здобувачів вищої освіти ІТ–спрямованості. Є декілька причин, чому вивчення зворотного проектування є важливим [3-4].

1. **Всебічний розвиток розуміння комп'ютерних систем**, програмного-апаратного забезпечення та архітектури операційної системи. За допомогою зворотного проектування можна дізнатися, як певна система функціонує на низькому рівні, отримуючи уявлення про те, як різні компоненти системи працюють разом для вирішення різних завдань.

2. Зворотне проектування є ключовим інструментом для **виявлення вразливостей та слабких місць** у програмному та апаратному забезпеченні. Аналізуючи вихідний код і системну архітектуру продукту, можна визначити потенційні недоліки безпеки та розробити способи їх подолання.

3. **Розвиток критичного мислення** та навиків вирішення проблем. Аналізуючи складні системи та програмне забезпечення, здобувачі вищої освіти вчаться мислити творчо та знаходити інноваційні рішення проблем. Ці навички є безцінними в ІТ галузі, де необхідно вміти аналізувати складні системи та розробляти рішення для складних проблем.

Розглянемо існуючі підходи зворотного проектування та їх актуальність у сучасних умовах.

1. **Аналіз зловмисного програмного забезпечення** (malware analysis). Дає можливість зрозуміти, як працює зловмисне програмне забезпечення, і розробити методи виявлення та запобігання його поширенню.

2. **Безпека Інтернету речей** (Internet of Things security). В сучасних умовах всебічного розповсюдження різноманітного обладнання Інтернету речей без технічної документації та з закритим вихідним кодом reverse engineering стає все більш важливим для розуміння вразливостей безпеки пристроїв і розробки ефективних заходів протидії.

3. **Протидія методам обфускації вихідного коду**. Обфускація коду – це техніка, яка використовується для ускладнення розуміння програмного коду систем. Використання технік зворотного проектування використовується для розробки методів обходу обфускації вихідного коду та аналізу основного коду.

4. **Системи захисту від шахрайства** (anti-cheat systems): зворотне проектування використовується при розробці систем захисту від шахрайства онлайн-ігор. Аналізуючи код гри, розробники можуть виявити та запобігти шахрайству та іншим формам нечесної гри.

5. **Зворотне проектування апаратного забезпечення.** Зворотне проектування не обмежується програмним забезпеченням. Він також використовується для аналізу та розуміння внутрішньої роботи апаратних пристроїв, таких як мікрочіпи, друковані плати та інші електронні компоненти.

Як висновок вивчення технологій зворотного проектування має вирішальне значення для здобувачів вищої освіти ІТ напрямку, оскільки воно дає їм цінні навички та знання, які користуються великим попитом у конкурентній ІТ-індустрії. Розуміючи, як системи працюють на найнижчому рівні, дає можливість глибше зрозуміти складні апаратні та програмні системи та навчитися розробляти інноваційні рішення для подолання викликів, з якими вони стикаються. Навчальним закладам вищої освіти важливо включати підходи зворотного проектування в свої навчальні дисципліни освітньо-професійних програм, щоб озброїти здобувачів освіти необхідними навичками та знаннями для процвітання в динамічній сфері ІТ, що постійно розвивається.

Список літератури

1. Bruce Dang, Alexandre Gazet, Elias Bachaalany Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation. Wiley. 2014. 384 p.

2. Jitender Narula Implementing Reverse Engineering. BPB Publications. 2021. 494 p.

3. Reginald Wong Mastering Reverse Engineering: Re-engineer your ethical hacking skills. Packt Publishing. 2018. 436 p.

4. Abhinav Mishra Mobile App Reverse Engineering. Packt Publishing. 2022. 166 p.

УДК 378:004

А.С. Коваленко к.т.н., доцент, О.В. Коваленко, д.т.н., доцент, Є.А. Деркач, магістрант,
annasun911@gmail.com, clashav@gmail.com, derkachea@gmai.com
Центральноукраїнський національний технічний університет,м. Кропивницький

АКТУАЛЬНІСТЬ ВИКЛАДАННЯ ТА РОЗВИТКУ SOFT SKILLS У НАВЧАЛЬНИХ ДИСЦИПЛІНАХ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ ЗВО ІТ–СПРЯМОВАНОСТІ

На сучасному високо конкурентному ІТ ринку праці знання Hard skills, безсумнівно, є вирішальними для ІТ фахівців, які прагнуть увійти в індустрію ІТ технологій. Однак важливість Soft skills навичок неможливо переоцінити, оскільки вони можуть значно вплинути на успіх в кар'єрі. Викладання та розвиток Soft skills у закладах вищої освіти (ЗВО) здобувачам освіти є найактуальнішим сучасним трендом.

Соціально-комунікативні навички чи гнучкі навички (Soft skills) в ІТ – це особисті якості, які дозволяють ІТ фахівці ефективно спілкуватися, співпрацювати та адаптуватися до змін на робочому місці.

Сучасні підходи розробки програмного забезпечення стають все більш орієнтованими на командну розробку, а здатність працювати разом з різноплановими спеціалістами стає все більш критичною. Програмісти та інші ІТ фахівці повинні вміти ефективно працювати з іншими членами команди та ефективно спілкуватися, щоб досягти бажаних результатів.

Окрім ефективної роботи в команді, ІТ фахівці також повинні бути адаптованими та здатними швидко вивчати нові технології. У технологічній індустрії інструменти програмного забезпечення та мови програмування можуть швидко змінюватися, і ІТ фахівці повинні мати можливість швидко адаптуватися до нових технологій і методів. Це вимагає здатності бути гнучким і адаптованим.

Крім того, ІТ фахівці також повинні мати сильні навички критичного мислення. Вони повинні вміти аналізувати проблеми та швидко знаходити ефективні рішення. Крім того, вони повинні мати чудові навички управління часом та організаційні навички, щоб ефективно керувати проектами та дотримуватись термінів. Ці навички спілкування необхідні для того, щоб проекти були завершені вчасно та в межах бюджету.

Розглянемо актуальні напрями Soft skills для ІТ фахівців з визначенням ключових навиків, на розвитку яких необхідно зосередитися, наведемо приклади в яких напрямах та навчальних дисциплінах це можна застосувати.

Проведені дослідження дозволили виділити наступні актуальні у 2023 році напрями Soft skills: Тайм-менеджмент; Адаптивність; Вирішення проблем; Командна робота; Креативність в роботі; Лідерство; Навички міжособистісного спілкування; Трудова етика; Увага до деталей під час реалізації поставлених завдань. Розглянемо першочергові напрями.

Адаптивність – це перший найважливіший навик ІТ фахівців. Технологічна галузь постійно розвивається, і програмісти повинні мати можливість швидко адаптуватися до нових технологій, мов і робочого середовища. Здатність навчатися новому та адаптуватися до нових ситуацій може допомогти програмістам залишатися попереду та процвітати в індустрії технологій. Зміна місця роботи, напрямку роботи чи основного інструментарію значною мірою вводить ІТ фахівця у негативний емоційний стан фрустрації що значно зменшує його продуктивність та призводе до вигорання.

Критичне мислення та **вміння розв'язувати проблеми**. ІТ фахівці, які вміють критично мислити та ефективно вирішувати проблеми, є цінними активами для будь-якої команди. Вони можуть допомогти виявити проблеми та швидко запропонувати ефективні рішення. Крім того, сильні навички вирішення проблем можуть допомогти дотримуватись термінів виконання проекту (тайм-менеджмент) та покращувати якість їхньої роботи.

Лідерство. Хоча ІТ фахівці наприклад програмісти не завжди можуть займати керівну позицію, здатність ефективно керувати малою чи середньою командою може допомогти їм виділитися серед своїх колег та значно підвищити свій статус. Лідерські навички включають здатність мотивувати та надихати інших, делегувати завдання та ефективно спілкуватися.

Співпраця та спілкування навичка спілкування є критично важлива для програмістів-початківців. Уміння ефективно працювати з іншими членами команди та ефективно спілкуватися може допомогти досягти кращих результатів за менший час, що дуже високо ціниться на високо конкурентному ІТ ринку праці. Це також може допомогти створити позитивну робочу атмосферу та підвищити моральний дух команди. Крім того, програмісти, які вміють добре спілкуватися, можуть краще сформулювати свої ідеї та рішення, що може призвести до більш значних можливостей кар'єрного зростання.

Проведено дослідження та огляд галузі знань 12 «Інформаційні технології», а саме стандартів вищої освіти України першого (бакалаврського) рівня: 122 «Комп'ютерні науки»; 123 «Комп'ютерна інженерія»; 125 «Кібербезпека» показало що вивчення Soft skills необхідно проводити в перших семестрах навчання.

З забезпеченням загальних компетентностей.

122 «Комп'ютерні науки»: «ЗК6. Здатність вчитися й оволодівати сучасними знаннями»; «ЗК8. Здатність генерувати нові ідеї (креативність)»; «ЗК9. Здатність працювати в команді»; «ЗК10. Здатність бути критичним і самокритичним»; «ЗК11. Здатність приймати обґрунтовані рішення».

123 «Комп'ютерна інженерія»:«Z2. Здатність вчитися і оволодівати сучасними знаннями»; «Z6. Навички міжособистісної взаємодії»; «Z8. Здатність працювати в команді».

125 «Кібербезпека»: «КЗ 2. Знання та розуміння предметної області та розуміння професії»; «КЗ 4. Вміння виявляти, ставити та вирішувати проблеми за професійним спрямуванням».

З забезпеченням спеціальних (фахових) компетентностей.

122 «Комп'ютерні науки»: «СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника».

123 «Комп'ютерна інженерія»: «Р11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів»; «Р15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення».

125 «Кібербезпека»: «КФ 2. Здатність до використання інформаційнокомунікаційних технологій, сучасних методів і моделей інформаційної безпеки та/або кібербезпеки».

Найактуальніші напрями вивчення та розвитку Soft skills у ЗВО є вивчення навчальних дисциплін з викладанням: вступу до спеціальності; першої мови програмування (C,C++, C#, python); сучасних методологій розробки програмного забезпечення; іноземної та Української мови (за професійним спрямуванням); Вищої математики (теорія ймовірності, дискретна математика).

Як висновок можна сказати що розвиток Soft skills здобувачами освіти ЗВО виходять далеко за межі успіху в кар'єрному зростанні. Ці навички також можуть покращити особисте та соціальне життя людини, покращуючи її здатність ефективно спілкуватися, співпереживати іншим і конструктивно вирішувати конфлікти. Soft skills можуть допомогти побудувати міцніші стосунки з колегами та дати можливість стати ефективним лідером думок та учасником різноманітних спільнот. Хоча технічні навички вкрай необхідні, їх недостатньо, щоб гарантувати успіх на сучасному IT ринку праці.

Список літератури

1. Стандарт вищої освіти України першого (бакалаврського) рівня ступеня «бакалавр» за галуззю знань 12 «Інформаційні технології», спеціальністю 122 «Комп'ютерні науки». URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2019/07/12/122-kompyut.nauk.bakalavr-1.pdf (дата звернення: 01.04.2023).

2. Стандарт вищої освіти України першого (бакалаврського) рівня ступеня «бакалавр» за галуззю знань 12 «Інформаційні технології», спеціальністю 123 «Комп'ютерна інженерія». URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/123-kompyuterna-inzheneriya.pdf (дата звернення: 01.04.2023).

3. Стандарт вищої освіти України першого (бакалаврського) рівня ступеня «бакалавр» за галуззю знань 12 «Інформаційні технології», спеціальністю 125 «кібербезпека». URL: https://mon.gov.ua/storage/app/uploads/public/5bb/626/1a8/5bb6261a84776166409164.pdf (дата звернення: 01.04.2023).

УДК 004.4

В.А. Федорієнко[1], О.С. Прокопенко[2]
[1]v.fedoriienko@edu.nuou.org.ua com,
кандидат технічних наук,
начальник науково-дослідного відділу навчально-наукового центру стратегічних комунікацій у сфері забезпечення національної безпеки та оборони,
Національний університет оборони України імені Івана Черняховського, м. Київ
[2]o.prokopenko@edu.nuou.org.ua,
доктор філософії (комп'ютерні науки),
начальник науково-дослідної лабораторії навчально-наукового центру стратегічних комунікацій у сфері забезпечення національної безпеки та оборони,
Національний університет оборони України імені Івана Черняховського, м. Київ

АСПЕКТИ УДОСКОНАЛЕННЯ ТЕХНОЛОГІЇ МОНІТОРИНГУ ІНФОРМАЦІЙНОГО ПРОСТОРУ НА ОСНОВІ АНАЛІЗУ РИНКУ ПРОГРАМНИХ ПРОДУКТІВ

Сьогодні комп'ютерні засоби автоматизованого моніторингу інформаційного простору набувають все більшого поширення. Перш за все це обумовлено потребою побудови стійкого інформаційного простору України, особливо під час широкомасштабної збройної агресії рф проти України. Вчасне виявлення негативного впливу агресора надасть змогу вчасно відреагувати і ефективно протидіяти на визначених рівнях. Загалом, така протидія полягає у нейтралізації викликів і загрозам застосування негативного інформаційно-психологічного впливу у відкритих джерелах інформації, що значно ускладнює процеси реалізації комунікативних заходів для досягнення стратегічних цілей держави. Негативний інформаційний вплив, заснований на використанні змісту ворожих пропагандистських наративів, активно розповсюджуються у глобальному інформаційному просторі, що негативно впливає на соціальну свідомість, нав'язуючи створений противником світогляд, твердження, факти, аргументи, чутки, тощо [1, 2].

Для ефективного реагування, сьогодні гостро постає питання використання сучасних інформаційних технологій моніторингу, збору і обробки інформації з відкритих джерел у інформаційних системах військового призначення. Це пояснюється необхідністю вироблення необхідних заходів інформаційного протиборства проти розгорнутих інформаційних спеціальних операцій противника, які можуть відбуватися комплексно з активізацією, підвищеною динамічністю і напруженістю ведення бойових дій на окремих напрямах фронту. За кількістю, інтенсивністю і масштабністю вкидань інформаційної пропаганди ворога, на основі використання відомих математичних методів аналізу і прогнозування, можливо визначати етапи інформаційно-психологічної спеціальної операції, що надає додаткові можливості для своєчасних адекватних дій з інформаційної протидії.

Питання моніторингу інформаційного простору найбільшої ефективності набуло у програмних продуктах зарубіжних і вітчизняних розробників. До них відносяться як повнофункціональні продукти, де забезпечуються процеси обробки даних: збір даних, аналіз даних, візуалізація і інтерпретація даних, так і окремі сервіси і бібліотеки, за допомогою яких спрощуються процеси розробки програмного забезпечення на високорівневих мовах програмування: Python, C#, C++, PhP, Java та інші.

До першої категорії програмних продуктів належать: IdexAttack, Semantrum, InfoStream, Wisdom Well, Web-Observer, Semantic Force, Multigo, Google Trends, Brandwatch, UAport, GreyLog. Усі зазначені продукти працюють в мережі Інтернет та реалізовані на веб платформах. Доступ здійснюється через веб браузер, за необхідністю для здійснення інтеграції між компонентами систем надається API-конектор [3]. Здебільшого, зазначені програмні продукти, розроблені у вигляді агрегаторів новин, у яких за допомогою відповідних фільтрів відслідковується необхідна інформація з обраних відкритих джерел. Обробка текстових даних веб-сторінок і соціальних мереж здійснюється на основі Web-скрапінгу і парсингу даних. Технології Machine Learning в продуктах автоматизованого моніторингу вирішують задачі класифікації і кластеризації при розпізнаванні характеристик досліджуваного контенту інформації. Зрештою, оброблені дані візуалізують за допомогою спеціалізованих засобів у вигляді графіків, діаграм та інших графічних представлень, що дозволяє легше сприймати великі об'єми даних, швидко отримати візуальне уявлення про те, як дані пов'язані між собою і як вони можуть бути використані для вирішення конкретних проблем.

Проте, використання наведених вище програмних продуктів в інформаційно-аналітичному забезпеченні супроводжується низкою питань, пов'язаних з порядком обробки даних та їх раціональної візуалізації. Технологічні аспекти інформаційно-аналітичного забезпечення моніторингу інформаційного простору, крім вирішення наведених вище задач, мають включати наступний функціонал:

можливості з приймання-передавання, обробки текстових повідомлень про розташування і дії ворога, від населення тимчасово окупованих територій;

технологію пошуку ворожих наративів в інформаційних повідомленнях відкритих джерел інформації, їх взаємозв'язок і прогнозовані ступені ризику на певні складові сектору безпеки і оборони України;

систему підтримки прийняття рішень для оперативного вироблення, прийняття і реалізації необхідних заходів протидії деструктивного інформаційно-психологічного впливу противника.

Існуючі у практиці використання інформаційно-аналітичного забезпечення невідповідності, створюють передумови для розроблення інформаційної технології виявлення і аналізу негативного інформаційно впливу. В сучасних реаліях, зазначене можливо досягти за рахунок низки різноманітних сервісів і спеціалізованих додатків обробки текстових масивів даних, можливості яких дозволяють виявляти та аналізувати інформаційні загрози. До таких відносяться:

сервіси для аналізу тональності тексту – для визначення позитивного, негативного або нейтрального відтінку тексту: Такі сервіси, як TextRazor [4] та Aylien, можуть бути використані для аналізу соціальних мереж, новин та інших джерел інформації. Зазначене здійснюється на основі глибокого аналізу текстових даних для вилучення зв'язків, типізованих залежностей між словами та синонімами, уможливлюючи потужні контекстно-семантичні конструкції.

сервіси для аналізу семантики та структури тексту, наприклад Gensim, FastText та SpaCy – дозволяють відшукувати спільні слова та теми у текстах. Вони можуть бути використані для аналізу новин, блогів, відгуків користувачів та інших джерел інформації. На прикладі бібліотеки машинного навчання для обробки текстів Gensim [5], можливо виокремити такі специфічні функції, як: аналіз семантики та структури тексту, тематичне моделювання, векторне представлення слів. Функції тематичного моделювання Gensim, на основі байєсівської моделі тематичного моделювання (Hierarchical Dirichlet Process), здійснює автоматизоване розпізнавання тематики текстового контенту. А на основі ймовірнісної моделі тематичного моделювання (Latent Dirichlet Allocation) – визначити кількість тематик та їх ключові слова.

Інструментарій для збору та аналізу даних з соціальних мереж, які дозволяють відслідковувати обговорення певної теми в соціальних мережах. Такі інструменти, як: Social Mention, Netvibes та Hootsuite, можуть використовуватися для моніторингу репутації, виявлення інформаційних загроз та аналізу тенденцій у громадській думці.

Наведені вище бібліотеки побудовані на базі платформи з відкритим кодом NLTK (Natural Language Toolkit) [6] для роботи з природними мовами. Вона надає доступ до корпусів текстів та лексичних ресурсів, а також має набір інструментів для обробки текстів, що допомагає вирішувати завдання, пов'язані з обробкою природних мов.

Сукупність наведених вище положень, дозволяє стверджувати про доволі обширні можливості сучасного інформаційного забезпечення, що дозволяє створювати нові інформаційні технології моніторингу інформаційного простору для певної специфіки діяльності. Можливості з обробки тексту, побудованих на основі технологій штучного інтелекту, дозволяють виявляти у джерелах інформації не лише маніпулятивний зміст і ворожі наративи, а також здійснювати:

аналіз якості контенту інформації, за критеріями граматичної та стилістичної правильності, логіки, послідовності, та відповідності мовленнєвому етикету;

аналіз ключових слів у повідомленнях, які пов'язані з конкретною темою, та визначити, які саме аспекти повідомлень є ворожими;

оцінку рівня страху у повідомленнях, на основі аналізу лексики, тону та стилю повідомлень;

оцінку рівня конфліктності, на основі аналізу кількості та ступеню емоційної напруги в повідомленнях, а також за кількістю звернень до агресивних слів та висловлювань;

дослідження джерел (соціальних мережі, тематичні блоги), які поширюють інформацію до інших джерел.

Список літератури

1. Почепцов Г. Сучасні інформаційні війни. Видання третє, доповнене та перероблене. Київ : Видавничий дім "Києво-Могилянська академія", 2016. 504 с.

2. Курбан О. В. Сучасні інформаційні війни у мережевому он-лайн просторі. Навчальний посібник. Київ : ВІКНУ, 2016. 286 с.

3. Коротко про API та його тестування. – URL : https://qagroup.com.ua/publications/korotko-pro-ari-ta-jogo-testuvannia/ (дата звернення: 18.03.2023).

4. TextRazor. – URL : https://sourceforge.net/software/product/TextRazor/ (дата звернення: 18.03.2022).

5. D'Agostino A. How to Train a Word2Vec Model from Scratch with Gensim. – URL : https://towardsdatascience.com/how-to-train-a-word2vec-model-from-scratch-with-gensim-c457d587e031 (дата звернення: 13.03.2022).

6. Documentation. Natural Language Toolkit. – URL : https://www.nltk.org/ (дата звернення: 14.03.2023).

УДК 004.8, 004.94

А.М. Мельник, А.С. Коваленко, к.т.н., доцент
mselnikanna@gmail.com, annasun911@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ОГЛЯД ПІДХОДУ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ СТВОРЕННЯ ДИЗАЙНУ САЙТУ

Зі стрімким розвитком технологій створити веб-сайт, який виділявся б із натовпу, стало дедалі складніше. Дизайн веб-сайту, який є привабливим, функціональним і зручним для користувача, має вирішальне значення для успіху будь-якого онлайн-бізнесу. В останні роки використання нейронних мереж [1-2] для створення дизайну веб-сайтів стає все більш популярним через їх здатність навчатися та вдосконалюватися з часом.

Дизайн веб-сайтів пройшов довгий шлях з перших днів існування Інтернету, де базові макети HTML і CSS були нормою. З появою сучасних веб-технологій і постійно зростаючим попитом на візуально привабливі та інтуїтивно зрозумілі інтерфейси веб-дизайнери постійно шукають нові способи створення приголомшливих дизайнів, які зацікавлять користувачів.

Однією з найновіших тенденцій веб-дизайну є використання нейронних мереж [2-3] для автоматичного створення дизайнів. Нейронні мережі – це тип алгоритму машинного навчання, який можна навчити на великому наборі даних розпізнавати закономірності та робити прогнози. У випадку веб-дизайну нейронну мережу можна навчити на великому наборі даних існуючих веб-сайтів і використовувати ці знання для автоматичного створення нових дизайнів.

Є кілька причин, чому використання нейронної мережі для створення дизайну веб-сайтів актуальне у сьогоденні: це може заощадити час і зусилля дизайнерів; створення дизайну веб-сайту з нуля може бути трудомістким і виснажливим процесом, що включає численні ітерації та перегляди; використовуючи нейронну мережу для автоматичного створення проектів, дизайнери можуть зосередитися на доопрацюванні та налаштуванні дизайнів, створених мережею, а не починати щоразу з нуля. Нейронні мережі можуть створювати унікальні та креативні проекти, які може бути важко придумати за допомогою традиційних методів проектування. Нейронні мережі можуть розпізнавати шаблони в існуючих дизайнах і вчитися на них, що дозволяє їм створювати нові та інноваційні дизайни, натхненні найкращими елементами існуючих дизайнів.

Нейронні мережі можна використовувати для створення дизайну, який є унікальним, візуально привабливим і адаптованим до конкретних потреб бізнесу. Вони можуть аналізувати такі дані, як колірні схеми, типографіка та параметри макета, щоб створювати проекти, які відповідають вимогам користувача.

Використання нейронних мереж у веб-дизайні також пропонує кілька переваг, таких як підвищення ефективності та економічності, а також можливість створювати персоналізовані дизайни, які задовольняють унікальні потреби окремих користувачів.

Хоча все ще існують певні обмеження щодо використання нейронних мереж у веб-дизайні, такі як потреба в обширних навчальних даних і потенціал упереджень, триваючі дослідження та розробки в цій галузі, ймовірно, подолають ці проблеми та створять дизайн на основі нейронних мереж. ще більш цінний інструмент для веб-розробників.

Загалом, використання нейронних мереж у дизайні веб-сайтів є значним кроком вперед в еволюції веб-розробки та, ймовірно, відіграватиме дедалі важливішу роль у формуванні майбутнього онлайн-досвіду. Адже, навчитися генерувати варіанти та прототипи за секунди та хвилини можна завдяки алгоритмам нейронної мережі. Крім того, такі послуги економлять час як розробника, так і клієнта та роблять життя простішим.

Список літератури

1. Lingfei Wu, Peng Cui, Jian Pei Graph Neural Networks: Foundations, Frontiers, and Applications. Springer. 2022. 725 p.

2. Aggarwal C. Neural Networks and Deep Learning, London, Springer International Publishing, 2018, 512 c.

3. Martin T Hagan, Howard B Demuth, Mark H Beale Neural Network Design. Martin Hagan. 2014. 800 p.

УДК 004.04

Д.О. Пархоменко, магістрант Б.Є. Золотухін, О.В. Коваленко, д.т.н., доцент,
dmitryparkhomenko11@gmail.com, b.zolotuxin@gmail. com, clashav@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

СУЧАСНІ ПІДХОДИ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ BIG DATA В МЕДИЧНІЙ ГАЛУЗІ

Останніми роками в галузі медицини спостерігається вибух даних [1-2], які можуть революціонізувати лікування пацієнтів. З появою електронних медичних записів, переносних пристроїв та інших цифрових технологій охорони здоров'я медичним працівникам доступна велика кількість даних. Однак проблема полягає в тому, щоб осмислити ці дані та перетворити їх на практичні ідеї. Саме тут на допомогу приходить технологія великих даних.

Big data [3] - це термін, який використовується для опису великих, складних та швидко зростаючих наборів даних, які не можуть бути ефективно оброблені традиційними методами. У медицині Big data може включати дані з різних джерел, таких як: медичні записи; зображення; генетичні дані; інформацію про спосіб життя та інші джерела даних.

Використання Big data в медицині може допомогти покращити діагностику, лікування та профілактику захворювань, а також підвищити ефективність та якість охорони здоров'я [4].

Розглянемо застосування Big data в медицині, проблеми та перспективи його використання.

1. **Збір та аналіз медичних даних** - це один з основних застосувань Big data в медицині. Збір та збереження медичних даних може бути здійснено з використанням електронних медичних записів, медичних зображень, мобільних додатків та інших джерел. Аналіз цих даних з використанням алгоритмів машинного навчання та штучного інтелекту може допомогти виявити певні закономірності та тенденції, які можуть бути використані для покращення діагностики та лікування захворювань.

2. **Покращення діагностики та лікування**. З використанням аналізу медичних даних можна виявити складні взаємозв'язки між різними показниками здоров'я, що допоможе лікарям приймати кращі рішення щодо діагностики та лікування пацієнтів. Наприклад, за допомогою Big data можна виявити певні медичні ризики та проводити персоналізований підхід до кожного пацієнта.

3. **Попередження захворювань та підвищення якості охорони здоров'я**. З використанням Big data можна виявити певні тенденції та закономірності в розвитку захворювань та ризиків їх поширення. Це може допомогти вчасно реагувати на загрози здоров'ю та запобігати поширенню захворювань. Крім того, Big data може бути використано для покращення якості охорони здоров'я. За допомогою аналізу медичних даних можна виявляти проблемні моменти в системі охорони здоров'я та пропонувати рішення для їх вирішення. Наприклад, це може бути виявлення недостатнього фінансування окремих напрямків медицини або недостатньої кваліфікації медичного персоналу.

Таким чином, використання Big data в медицині є важливим напрямком розвитку медичної галузі. Збір та аналіз медичних даних, використання їх для покращення діагностики та лікування, а також попередження захворювань та підвищення якості охорони здоров'я - це лише деякі з аспектів використання Big data в медицині.

Розглянемо існуючі труднощі використання Big data в медицині.

Незважаючи на те, що використання Big data в медицині має великий потенціал для поліпшення якості лікування та попередження захворювань, воно також стикається з деякими труднощами.

1. **Проблема конфіденційності та захисту даних** є важливою при використанні медичних даних. Великі обсяги медичних даних містять особисту інформацію про пацієнтів, що може бути використана зловмисниками для різних цілей. Тому необхідно забезпечити високий рівень конфіденційності та захисту даних.

2. Необхідно бути уважними при використанні Big data в медицині, оскільки недостатньо точні або невірні результати можуть призвести до **неправильно проаналізовані висновки та помилкові рішення** щодо лікування пацієнтів. Тому необхідно забезпечити якість даних та точність аналізу

3. **Недостатня кількість кваліфікованих фахівців**, які можуть працювати з Big data в медицині. Не всі лікарі та фахівці в галузі медицини мають достатні навички та знання для аналізу великих обсягів даних. Тому необхідно забезпечити зручний та доступний інтерфейс для лікарів та інших фахівців, щоб вони могли легко використовувати великий обсяг даних у своїй роботі та в цілому провести трудомістке впровадження програмних продуктів.

4. Недостатня стандартизація медичних даних. Оскільки дані зібрані з різних джерел, може виникнути проблема з невідповідністю даних, що може призвести до неточностей та спотворених результатів.

Отже, використання Big data в медицині стикається з деякими труднощами, такими як проблема конфіденційності та захисту даних, необхідність забезпечення якості даних та точності аналізу, а також необхідність забезпечити зручний та доступний інтерфейс для фахівців в галузі медицини.

Розглянемо перспективи використання Big data в медицині.

1. Використання Big data в медицині має потенціал для революційних змін в галузі охорони здоров'я. Однією з найважливіших перспектив є розвиток машинного навчання та штучного інтелекту в медицині. Машинне навчання може допомогти в **прогнозуванні захворювань, діагностиці та лікуванні**. Наприклад, машинне навчання може допомогти у виявленні ознак захворювання на ранній стадії та розробці індивідуальних планів лікування.

2. **Побудова індивідуальних планів лікування та профілактики**. Збір та аналіз великої кількості даних про пацієнта може допомогти у побудові індивідуальної стратегії лікування та профілактики. Це дозволить пацієнтам отримувати більш ефективну та індивідуальну медичну допомогу.

3. **Глобальні системи охорони здоров'я**. Збір та аналіз великої кількості даних про стан здоров'я нації може допомогти у виявленні тенденцій та покращенні системи охорони здоров'я. Наприклад, аналіз даних може допомогти виявити популяції зі збільшеним ризиком захворювання та розробити стратегії профілактики.

Використання Big data в медицині має великий потенціал для розвитку галузі та поліпшення якості медичної допомоги. Розвиток машинного навчання та штучного інтелекту, використання Big data для побудови індивідуальних планів лікування та профілактики, а також впровадження Big data в глобальні системи охорони здоров'я може значно поліпшити ефективність медичної допомоги та допомогти зберегти тисячі життів.

Big data в медицині має безліч переваг, серед яких підвищення ефективності та точності діагностики та лікування захворювань, зниження витрат на охорону здоров'я та покращення якості медичної допомоги. Однак, разом із тим, з'являються труднощі, пов'язані із захистом конфіденційності пацієнтів, недостатньою точністю даних, а також недостатньою кількістю кваліфікованих фахівців. Незважаючи на це, перспективи подальшого розвитку застосування Big data в медицині дуже амбіційні. Розвиток машинного навчання та штучного інтелекту, використання Big data для побудови індивідуальних планів лікування та профілактики, а також впровадження Big data в глобальні системи охорони здоров'я можуть значно поліпшити ефективність медичної допомоги та допомогти зберегти тисячі життів.

Окрім того, необхідно вдосконалювати правову базу щодо збору, зберігання та використання медичних даних, а також забезпечувати кваліфіковану освіту та підготовку медичних фахівців для роботи зі складними медичними даними.

Таким чином, використання Big data може значно покращити якість медичної допомоги, сприяти розвитку галузі та дозволити більш ефективно вирішувати проблеми зі здоров'ям населення. Проте, для досягнення цих цілей потрібно вирішити ряд труднощів, пов'язаних із захистом конфіденційності пацієнтів, недостатньою точністю даних, а також недостатньою кількістю кваліфікованих фахівців. На жаль, ці труднощі є складними, тому їх розв'язання вимагає багато зусиль та співпраці між галузевими експертами та владними органами. Однак, з урахуванням потенціалу, який надає використання Big data в медицині, ці зусилля є виправданими та можуть призвести до значних досягнень в медичній науці та практиці.

Список літератури

1. Katherine Marconi, Harold Lehmann Big Data and Health Analytics. Auerbach Publications, 2014. 328 p.

2. Ashish Khanna, Deepak Gupta, Nilanjan Dey Applications of Big Data in Healthcare. Academic Press. 2021. 281 p.

3. Pantea Keikhosrokiani Big Data Analytics for Healthcare. Academic Press. 2022. 354 p.

4. Pardeep Kumar, Yugal Kumar, Mohamed A. Tawhid Machine Learning, Big Data, and IoT for Medical Informatics. Academic Press. 2021. 458 p.

УДК 004.75

Ю.М. Пархоменко, Т.Д. Ахметов, А.І. Козак
parhomenkoym@ukr.net, andilarkin@gmail.com, scpacker19@gmail.com
Центральноукраїнський національний технічний університет,м. Кропивницький

РОЗРОБКА ПІДСИСТЕМИ ДЛЯ УПРАВЛІННЯ ТА АДМІНІСТРУВАННЯ НЕОДНОРІДНОЇ МЕРЕЖІ

Нові виклики в розподілених середовищах протягом тривалого часу змушують адміністраторів інформаційних систем розглядати управління мережею як головну турботу. При цьому системне адміністрування відходить на другий план, а пов'язані з ним інструменти з'являються як самостійні сервіси, чужі платформам і додаткам мережевого управління. Ця інверсія, яка не повністю відповідає логіці функціонування корпоративних інформаційних систем (оскільки мережа виконує лише роль допоміжної інфраструктури), зберігається протягом кількох років.

Оскільки кількість розподілених програм і особливо мережевих баз даних перевищила певну межу, роль системного адміністрування значно зросла. За цим послідував процес інтеграції системного та мережевого керування, який вимає від великих виробників швидкого вдосконалення своїх продуктів.

Таким чином, існує потреба в розробці програмного забезпечення, яке може задовольнити потребу в реалізації сучасних протоколів, враховуючи пластичність, при побудові інфраструктури з неоднорідними з'єднаннями маршрутизації та інтеграції віртуального сегмента у фізичний сегмент. Другим менш важливим фактором є вартість обладнання, яке може використовуватися в системі.

Програмно-апаратне рішення підсистеми повинне бути компактним та наочним, легким для сприйняття, надавати користувачу в процесі роботи з підсистемою необхідну теоретичну допомогу.

Мінімальна конфігурація апаратних засобів, що забезпечать в повному обсязі функціонування програмного забезпечення підсистеми, має наступний вигляд:

- персональний комп'ютер;
- 4 мережеві карти;
- мінімальний об'єм оперативної пам'яті – 4Гб;
- операційна система linux Ubuntu server 20.

Таким чином, визначивши: складові компоненти майбутньої підсистеми; перелік функцій, виконання яких має забезпечити кожен з компонентів; вимоги, обмеження щодо підсистеми та кожного компонента окремо; шляхи реалізації проектних рішень; мінімальну конфігурацію комплексу технічних засобів, маємо всі необхідні дані для побудови функціональної схеми підсистеми.

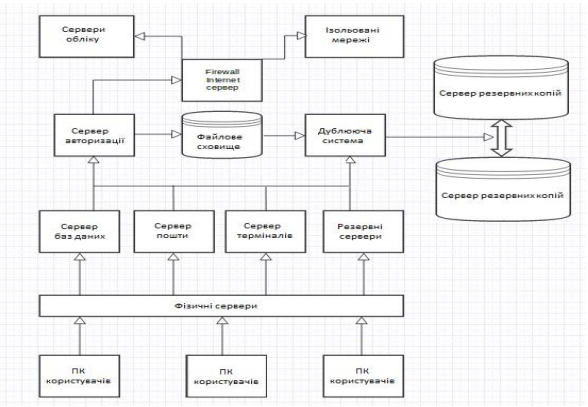


Рис.1 Пропонуєма функціональна схема підсистеми управління та адміністрування неоднорідної мережі.

Розроблена підсистема побудована по модульному принципу, тобто є головний, зв'язуючий модуль та підлеглі модулі, кожен з яких виконує якусь одну функцію. Головним модулем виступає мережний екранний фаєрвол IPTables при розробці правил внесені змінні, для спрощення записів, та записи які формують маршрути на базі ядра, що дає можливість уникнути додаткових скриптів.

Таким чином, розробник може без особливих труднощів допрацювати записи, виправити можливі помилки, додати або вилучити якісь нові вузли мережі не перезавантажуючи систему.

Пропонуєма система, буде виконувати багато різних операцій при роботі з мережею. Тому потрібно чітко розуміти топологію мережі яка підлягає адмініструванню.

УДК 004.04, 004.75

Ю.М. Пархоменко[1], С.О. Бондаревський[2], Д.О. Литвиненко[3]
parhomenkoym@ukr.net, bondarevsky@yahoo.com, dimasik.litvinenko.2014@gmail.com
[1]Центральноукраїнський національний технічний університет, м. Кропивницький

АНАЛІЗ ТА ВИБІР МОДЕЛІ ПОШУКУ КАТАЛОЖНИХ СИСТЕМ

Інформаційний пошук - напрям досліджень, що вивчає питання пошуку документів, обробки результатів пошуку, а також цілу низку суміжних запитань : моделювання, класифікації, кластеризації і фільтрації документів, проектування архітектури пошукових систем і призначених для користувача інтерфейсів, мови запитів, і так далі. Документ - це змістовно закінчена одиниця інформації, представлена на якій-небудь природній мові, яка ідентифікується унікальним чином Документ - це порція інформації, якою оперують інформаційно-пошукові системи. Інформаційно-пошукова система - це комплекс програмних засобів, що забезпечують виборчий відбір за заданими ознаками документів, що зберігаються в електронному (оцифрованому) уявленні. Одним з ключових понять, що характеризує вибір того або іншого методу аналізу текстової інформації, а також реалізацію конкретного варіанту пошуку, є модель пошуку [1].

Класично модель інформаційного пошуку побудована на системі каталогів і картотек. Система каталогів і картотек - сукупність планомірно організованих, доповнюючих один одного взаємозв'язаних каталогів і картотек, які розкривають зміст і структуру фондів в різних аспектах. Вони є важливим засобом розкриття фондів та надання інформації користувачам про наявні в них матеріали, а також дозволяють користувачам з найменшими витратами часу підібрати необхідне враховуючі наявні потреби [2].

Каталоги і картотеки містять про наявний у фонді документ дані, які складені згідно з існуючими правилами бібліографічного опису. Каталоги і картотеки розрізняються по видах відбитих документів, призначенні, охопленні фондів, способі угрупування і формі представлення інформації [3].

Електронний бібліотечний каталог - сукупність програмних і апаратних засобів по забезпеченню діяльності бібліотеки за замовленням, каталогізації, пошуку, видачі книг, рішення різних завдань по звітності і книгозабезпеченні читачів та ін. як в локальній обчислювальній мережі, так і через web-вміст.

І так, в каталозі інформація про зміст документів архіву згрупована по предметах (темам, галузям), розташованих відповідно до прийнятої для цього каталогу схеми класифікації документної інформації. Сукупність різних каталогів складає систему каталогів. Сукупність робіт з підготовки, створення і ведення каталогів називається каталогізацією. Каталогізація включає наступні види робіт : складання схеми класифікації, виявлення і відбір документів, їх опис і систематизація, ведення каталогів. Вибір схеми класифікації визначається складом об'єктів (документів, ...) і завданнями їх використання. Залежно від схеми побудови каталоги підрозділяються на систематичні, тематичні, предметні і їх різновиди (іменний, географічний, об'єктний). Об'єктом опису може виступати документ (група документів, частина документу), одиниця зберігання (одиниця обліку), опис, комплект, фонд (група фондів), що містять інформацію по певній темі (питанню). До складу описової статті входять: назва архіву, індекс, рубрика, підрубрика, дата події, місце події, зміст, номер фонду, назва фонду, номера опису, справи, листів, мова документу, спосіб відтворення, для карткового каталогу також прізвище укладача і дата складання описової статті.

Як правило, найбільш поширені являються систематичний і іменний каталоги. В систематичному каталозі документи класифікується по галузях знань і практичної діяльності, та розташовується в логічній послідовності. У іменному каталозі документи класифікується за абеткою прізвищ осіб, що згадуються в документах або є їх авторами. Подальша систематизація проводиться в хронологічній або логічній послідовності.

Отже при розробці проекту каталожної системи бажано в якості моделей організації пошуку – обирати класичні на базі яких будуються систематичні і іменні каталоги.

УДК 004.9

А.М. Мельник, В.В. Босько, В.А. Резніченко
mselnikanna@gmail.com
Центральноукраїнський національний технічний університет,м. Кропивницький

ДОСЛІДЖЕННЯ СУЧАСНИХ МЕТОДІВ РОБОТИ ТА НАПРЯМКІВ ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ

Використання генетичних алгоритмів значно поліпшує процес розробки різних математичних задач а також навчання та моделювання штучного інтелекту. Дослідження генетичних алгоритмів актуальні для дослідження сучасних методів роботи з ними та їх вдосконалення. Саме тому мене цікавить ця тема тези.

Генетичні алгоритми виникли в результаті спостереження і спроб копіювання природних процесів, що відбуваються в світі живих організмів, зокрема, еволюції, пов'язаної з нею селекції (природного відбору). Алгоритми допомагають вирішенню багатьох задач за допомогою спостережень за природними процесами [1].

Генетичний алгоритм – це еволюційний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом використанням механізмів, що нагадують біологічну еволюцію.

В генетичних алгоритмах використовуються терміни, подібні до тих, які використовується в біології (популяція, мутація, покоління, схрещування, потомство, хромосоми, гени тощо) [9].

Алгоритм виник під час проб копіювання та тестування різних біологічних процесів, які відбуваються в живих організмах окрема, еволюції та пов'язаної з нею селекції (природнього відбору) популяцій живих істот [9].

Генетичні алгоритми також застосовуються при розробці програмного забезпечення в системах штучного інтелекту(для якого раніше використовували лише нейронні мережі, але зараз генетичні алгоритми є альтернативних варіантом) [1].

Приклади задач де застосовуються генетичні алгоритм:

– Екстремальні задачі (пошук точок мінімуму і максимуму). В математиці поширені задачі на пошук максимуму та мінімуму точок і за допомогою ГА це можна вирішити автоматично.

– Настроювання штучної нейронної мережі. Такі обчислювальні системи побудовані на основі біологічних нейронних мереж, які складають мозок тварин, тому звісно ж налаштування нейронної мережі потребує використання генетичних алгоритмів [5, 6].

– Моделювання штучного життя (Artificial life systems). За допомогою створених людиною моделей і пристроїв вивчається життя, живі системи та їх еволюція [3].

– Біоінформатика (згортання білків і РНК). В цій галузі використовуються машинні алгоритми і статичні методи для аналізу біологічних даних [4].

– Різноманітні задачі на графах (задача комівояжера, розфарбування, знаходження парасполучень). Шляхом здійснення підбору, комбінування і варіації окремих параметрів можна розв'язувати різноманітні задачі на графах і багато інших задач [5].

– Навчання штучної нейронної мережі. Як і налаштування штучної мережі - це складний процес який побудований за принципом [5, 6].

– Штучне життя. Створене на основі спостереження за реальними біологічними процесами, та за допомогою різних пристроїв відбувається аналіз та створення штучного життя [3, 9].

Принцип роботи генетичних алгоритмів

Простий генетичний алгоритм випадковим чином генерує початкову популяцію структур. Робота генетичного алгоритму уявляє собою ітераційний процес, що продовжується доти, поки не виконаються задане число поколінь або будь-який інший критерій зупинки. В кожному поколінні генетичного алгоритму реалізується відбір пропорційно пристосованості, одноточковий кросинговер і мутація. Спочатку, пропорційний відбір призначає кожній структурі імовірність $P_s(i)$ рівну відношенню її пристосованості до сумарної пристосованості популяції.

Потім відбувається відбір (із заміщенням) усіх n особин для подальшої генетичної обробки, відповідно до величини $P_s(i)$. При такому відборі члени популяції з більш високою пристосованістю з більшою імовірністю будуть частіше вибиратися, ніж особини з низькою пристосованістю. Після відбору, n обраних особин випадковим чином розбиваються на $n/2$ пари. Для кожної пари з імовірністю P_c може застосовуватися кросинговер. Відповідно з імовірністю P_c кросинговер не відбувається і незмінені особини переходять на стадію мутації. Якщо кросинговер відбувається, отримані нащадки заміняють собою батьків і переходять до мутації. У генетичному алгоритмі зберігається основний принцип природного відбору – чим пристосованіше індивідуум (чим більше відповідне йому значення цільової функції), тим з більшою імовірністю він буде брати участь у схрещуванні. Тепер моделюються мутації – у декількох випадково обраних особинах нового покоління змінюються деякі гени. Потім стара популяція частково або цілком знищується і ми переходимо до розгляду наступного покоління. Популяція наступного покоління в більшості реалізацій генетичних алгоритмів містить стільки ж особин, скільки початкова, але в силу відбору пристосованість у ній у середньому вище [8].

Генетичні алгоритми поділяю за двома функціями: принцип налаштування алгоритмів та стратегіями відбору.

Налаштування генетичних алгоритмів

Генетичний алгоритм робить пошук рішень за допомогою:

– відбору гіперплощин (hyperplane sampling), здійснюваного кросовером, оскільки останній комбінує й сполучає шаблони батьків у їхніх дітях.

– методу hill-climbing, що забезпечується мутацією: особина випадковим образом змінюється – невдалі варіанти вимирають, корисні зміни зберігаються популяцією.

Дослідження показали, що на простих завданнях з малим розміром популяції ГА з мутацією (і без кросовера) знаходять рішення швидше. На складних багатоекстремальних функціях краще використовувати ГА із кросовером, оскільки цей метод більше надійний, хоча й вимагає більшого розміру популяції.

Стратегії відбору

Ранговий відбір (rank selection): для кожної особини її ймовірність потрапити в проміжну популяцію пропорційна її порядковому номеру у відсортованій по зростанню пристосованості популяції. Такий вид відбору не залежить від середньої пристосованості популяції.

Турнірний відбір (tournament selection): з популяції випадковим чином вибирається t особин, і краща з них міститься в проміжну популяцію. Цей процес повторюється N разів, поки проміжна популяція не буде заповнена. Найпоширенішийо варіант при $t = 2$.

Відбір усіканням (truncation selection): при відборі усіканням популяція сортується по пристосованості, потім береться задана частка кращих, і з них випадковим чином N разів вибирається особина для подальшого розвитку [9].

Висновок. Вивчаючи цю тему, можу підсумувати: генетичні алгоритмі побудовані на основі біологіних досліджень, метод. За допомогою алгоритмів можна здійснювати різного виду операції над числа(використовувати в математичних задача), навчати та моделювати штучний інтелект та досліджень біоінформатики. В майбутньому дуже важливо розвивати дослідження та створення нових задач для роботи з генетичними алгоритмами оскільки це допомагає покращити недосконалості роботи.

Список літератури

1. Генетичні алгоритми. Ключові поняття і методи реалізації [Електронний ресурс] – Режим доступу до ресурсу: http://www.znannya.org/?view=ga_general.

2. Генетичні та ройові алгоритми [Електронний ресурс] – Режим доступу до ресурсу: https://www.victoria.lviv.ua/library/students/sss/theme2.html.

3. Штучне життя [Електронний ресурс] – Режим доступу до ресурсу: https://www.wiki-data.uk-ua.nina.az/A-life.html.

4. Біоінформатика [Електронний ресурс] – Режим доступу до ресурсу: 5. https://uk.wikipedia.org/wiki/Біоінформатика.

5. Штучна нейронна мережа [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Штучна_нейронна_мережа.

6. Штучна нейронна мережа [Електронний ресурс] – Режим доступу до ресурсу: https://futurum.today/shtuchni-neironni-merezhi-shcho-tse-take/.

7. Троцько В. В. Методи штучного інтелекту [Електронний ресурс] / В. В. Троцько – Режим доступу до ресурсу: https://library.krok.edu.ua/media/library/category/navchalni-posibniki/trotsko_0001.pdf.

8. Іващук Р. Основний принцип роботи генетичного алгоритму[Електронний ресурс] / Р. Іващук – Режим доступу до ресурсу: https://elartu.tntu.edu.ua/bitstream/123456789/14648/2/Conf_2009v1_Ivashchuk_R-Osnovnii_printsip_roboti_196.pdf

9. Вікіпедія. Генетичні алгоритми [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Генетичні алгоритми

УДК 629.7.02

Д.Г. Бурейко, Р.М. Минайленко, О.К. Конопліцька-Слободенюк, Л.І. Поліщук
aron70@ukr.net
Центральноукраїнський національний технічний університет, м. Кропивницький

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ЗАХИСТУ ЕОМ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ ПРИ ПЕРЕХВАТІ IP-ФУНКЦІЙ

В даний час широкого поширення набули операційні системи сімейства Windows. Вони широко використовуються не тільки як домашні системи, але і як сервери. Ця лінійка ОС відрізняється непоганою захищеністю від шкідливих програм, а також для неї існує велика кількість додаткових систем безпеки (різні антивіруси, фаєрволли).

Встановивши антивірус і фаєрволл багато користувачів думають, що вони повністю захищені, і навіть більшість програмістів вважають, що досить частіше перевіряти свій комп'ютер на підозрілі речі (автозавантаження, процеси і.т.д.) і ніяка шкідлива програма до них не зможе проникнути.

Перехоплення системної функції API полягає у зміні деякої адреси в пам'яті процесу або деякого коду в тілі функції таким чином, щоб під час виклику цієї API функції управління передавалося не їй, а функції, яка замінює системну. Ця функція, замість системної, виконує заплановані користувачем дії, а потім або викликає оригінальну функцію або не викликає її взагалі.

ОС Windows побудовано на системі DLL (бібліотек, які завантажуються динамічно). Система надає додаткам сервісні API функції, за допомогою яких вона може взаємодіяти із системою. Перехоплення API функцій дозволяє обійти багато обмежень системи і робити з нею практично що завгодно.

Основна мова створеного коду програми - Delphi, але матеріал актуальний і для будь-якої іншої мови (C, C++, Асемблер і.т.д.). Єдина умова - мова має бути 100% компілюваною, а також підтримувати роботу з покажчиками та асемблерні вставки.

Розроблене програмне забезпечення дозволяє створити систему захисту ЕОМ від несанкціонованого доступу при перехваті IP-функцій.

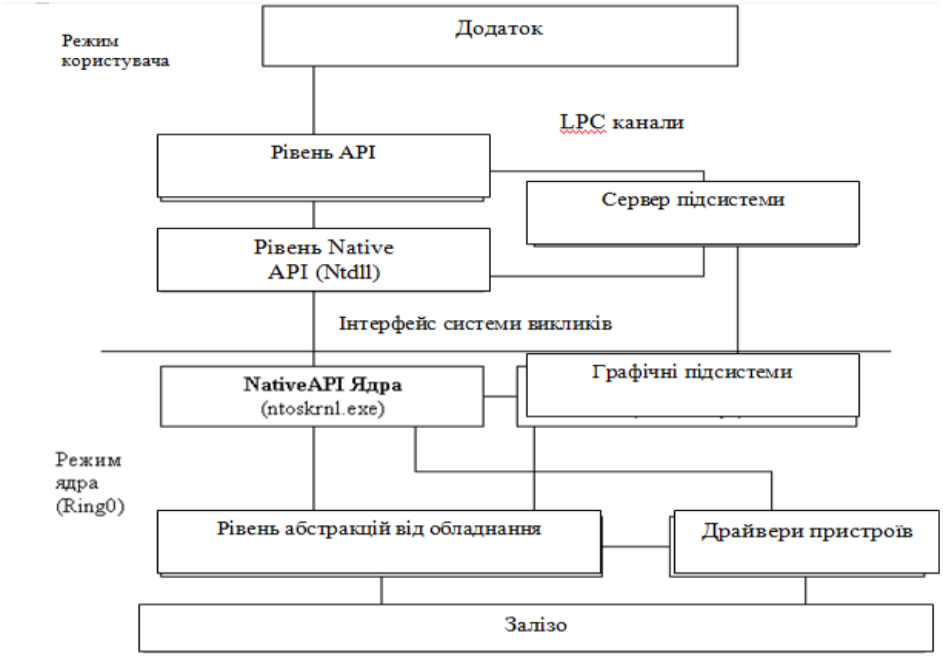


Рис.1. Структурна схема системи

Розроблена програма має інтуїтивно зрозумілий інтерфейс і є простою у використанні. Програма не вимагає для своєї роботи значних комп'ютерних ресурсів, вистачить будь-якого комп'ютера із ОС Windows.

Проведені розрахунки ефективності розробки та впровадження програмного продукту, показують несумнівну економічну доцільність широкого застосування розробленої інформаційної системи.

УДК 004.9

В. Кривохижа,бакалаврант,
vvvetal2003@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ГНУЧКІ МЕТОДОЛОГІЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В УПРАВЛІННІ ІТ-ПРОЕКТАМИ

У сучасному світі стандартизація підходів до управління проектами є важливою вимогою. Управління ІТ-проектами може здійснюватися класичними методами, але також можна використовувати підходи, які дозволяють ефективно працювати з великою кількістю невизначеності. Наразі широко використовуються гнучкі методології управління проектами, зокрема, Agile, до якого належать Scrum, Lean, Kanban, eXtremeProgramming та інші ітераційні методи управління, що мають широкий спектр інструментів для досягнення поставлених цілей.

Scrum - це гнучкий метод управління проектами, який ґрунтується на емпіричній теорії та ітеративний підхід, що дозволяє будувати ІТ-продукт починаючи з найважливіших його частин. Функціонал ІТ-проекту визначається за пріоритетами та планується в ітеративних інтервалах, які називаються спринтами. Команда складається з власника продукту, ScrumMaster і розробників.

Функціональність визначається поверхнево за допомогою історій користувачів (User Stories), що дозволяє команді розробників створювати гнучку імплементацію. Кожна ітерація розробки (спринт) виконується шляхом презентації реалізованих частин функціоналу клієнту та власнику продукту, а також безпосередньою доставкою заготовки продукту. Це дозволяє власнику продукту адаптувати завдання наступної ітерації відповідно до побажань клієнта та безпосередньо контролювати виконання проекту.

Скрам-майстер стежить за дотриманням процесів, що чітко визначені ним, та підтримує команду розробників, допомагаючи їм уникати перешкод та вирішувати проблеми, які виникають під час розробки продукту. В цілому, Scrum дозволяє замовнику максимально контролювати виконання проекту та максимізувати його вартість.

LEAN — це досить абстрактна методологія, але ефективний управлінський підхід, який спрямований на оптимізацію бізнес-процесів шляхом фокусування на потребах клієнтів, включаючи мотивацію працівників, задіяних у проекті. Робота поділяється на невеликі частини і кожна частина розробляється незалежно і окремо. У Лін немає чітких меж етапів, тому можлива паралельна реалізація декількох частин проекту, причому кожна частина може перебувати на різній стадії.

Принципи LEAN:

Визначення вартості. Аналіз споживчої або ринкової вартості та вкладення цінностей у продукт.

Створення карти потоку. Методика створення блок-схеми, нанесення на карту всіх етапів, через які проходить продукт до споживача.

Забезпечення безперервного потоку. Безперервний потік послуг або продукції від початку до кінця проекту шляхом застосування процесів.

Завантаження продукту. Робота на одній стадії процесу починається лише тоді, коли є потреба у використанні результатів на наступній стадії.

Вихідні дані для наступного етапу. Тобто, попит тягне за собою товар чи послугу через потік цінності.

Працювати над вдосконаленням. Максимальне скорочення витрат, щоб усі дії приносили цінність клієнту через безперервне покращення або нові відкриття.

Kanban – методологія, що реалізує принцип «вчасно» (just in time). На відміну від Scrum, Kanban не має чіткого визначення залучених ролей, проте також має роль власника продукту, що забезпечує створення карток завдань для реалізації проєкту. Проте Kanban не обмежує час ітерацій (спринтів). Канбан направлений на візуалізацію роботи та організацію безперервного потоку завдань.

Основні ключові практики методу Kanban:

1) Візуалізація. Побудований на ідеї прозорості поточного стану розвитку проєкту за рахунок використання дошок з колонками етапів реалізації, а також карток завдань проекту, які переміщуються між етапами. Отже, можна побачити завдання та етапи.

2) Обмеження кількісті завдань поетапно. Кожна стадія має ліміт, що налаштовується, на мінімальну і максимальну кількість завдань, які можуть перебувати на даній стадії, може бути на даному етапі. Кожен наступний етап висуває вимоги щодо початку реалізації завдань попереднього етапу. Таким чином, кожен наступний етап буде "тягнути" за собою початок виконання завдань на попередньому етапі реалізації.

3) Регулювання потоку. Переходи між станами постійно відстежуються, вимірюються та звітуються. На основі отриманих даних оновлюється карта виконання проєктів та проводиться аналіз змін в системі.

4) Зрозумілість потоку. Ефективність системи не буде досягнута до тих пір, поки процес не стане зрозумілим для всіх залучених до нього учасників. Потреба обговорення, чітке розуміння та об'єктивний розгляд проблеми.

5) Цикл зворотного зв'язку. Необхідні для впровадження вдосконалення процесів та організації в цілому.

6) Вдосконалення шляхом співпраці та експериментальних розробок.

Назва методології походить від ідеї застосування корисних традиційних методів і практик розробки програмного забезпечення, виводячи їх на новий "екстремальний" рівень. Наприклад, практика перегляду коду, яка полягає в тому, що один програміст переглядає код, написаний іншим програмістом.

Вибір ефективної методології управління проєктом є ключовим етапом в розробці ІТ-проєкту. Він визначає інструменти, процеси та робочі методики, що будуть використовуватися під час реалізації проекту, а також спосіб роботи спеціалістів, які беруть участь у проєкті. Вибір правильної методології дозволяє ефективно управляти процесом розробки, зменшити ризики та забезпечити успішне завершення проекту.

УДК 004.89

К.О. Задорожний, Є.В. Мелешко, М.С. Якименко
kostazadorozhnij9@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ ПЕРЕВАГ ТА РИЗИКІВ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В МЕДИЦИНІ

Штучний інтелект (ШІ) – це одна з найбільш важливих сучасних інформаційних технологій, яка змінює різні сфери нашого життя, включаючи медицину. Завдяки швидкому розвитку ШІ, виникла можливість використовувати інтелектуальні системи для допомоги лікарям у діагностуванні та лікуванні різних захворювань [1-9]. Тому не дивно, що його роль в медицині постійно зростає. ШІ може виявляти певні ознаки захворювань, проводити аналіз результатів тестів, прогнозувати розвиток захворювань та навіть розробляти індивідуальні плани лікування. ШІ може значно полегшити роботу лікарів, допомогти їм приймати важливі рішення та покращити результати лікування пацієнтів [2, 5].

Огляд систем штучного інтелекту для допомоги лікарям

Штучний інтелект у медицині – це новітня технологія, яка революціонізує сферу медицини. Нові системи штучного інтелекту дозволяють лікарям швидше та точніше діагностувати та лікувати хвороби, прогнозувати розвиток захворювань та ефективність лікування. Однією з систем штучного інтелекту для допомоги лікарям Є Merative (колишня IBM Watson Health). Це система штучного інтелекту, розроблена IBM, яка допомагає лікарям в діагностиці та прогнозуванні захворювань. Вона здатна виявити потенційні проблеми з судинною системою, розпізнати рак, визначити, чи схильний пацієнт до утворення тромбів. Merative може блискавично реагувати на те, коли необхідно вивчити нову інформацію і зробити висновки на її основі [3, 4]. Другою важливою системою є Google DeepMind Health. Ця система штучного інтелекту була розроблена для використання в онкології, прогнозування початку гострого ураження нирок та для аналізу сканувань очей, пошуку ранніх ознак захворювань, що призводять до сліпоти. Вона використовує нейромережі та аналіз даних, щоб допомогти лікарям у виявленні різних захворювань та розробці індивідуальних планів лікування [6]. Третьою важливою системою є Babylon Health. Ця система штучного інтелекту пропонує ефективні рішення для надання медичної допомоги. Вона використовує нейромережі та аналіз даних для забезпечення ефективної телемедицини, віртуальних консультацій та діагностування захворювань [7].

Діагностування захворювань за допомогою штучного інтелекту

Діагностування захворювань за допомогою штучного інтелекту є однією з важливих областей медичної науки. Штучний інтелект може виявляти симптоми та характеристики хвороб, що допомагає лікарям точніше та швидше поставляти діагнози. Інформація, що вводиться в систему, включає результати тестів, рентгенівські знімки, електрокардіограми та інші медичні дані. Однією з головних переваг штучного інтелекту в діагностуванні є його здатність до швидкого та точного аналізу великої кількості даних. Наприклад, за допомогою глибокого навчання, машинне навчання може навчитися розпізнавати патології на знімках, що значно спрощує роботу лікарів та зменшує ймовірність помилок у діагнозі. Інший важливий аспект діагностування за допомогою штучного інтелекту – це відсутність впливу людського фактору. Штучний інтелект не ставить перед собою задачі вибрати діагноз в залежності від симпатій до певної патології чи впливу зовнішніх факторів. Це забезпечує більш об'єктивну та точну діагностику. Хоча використання штучного інтелекту в медицині може зменшити роль лікарів у діагностуванні захворювань, вони все ще необхідні для інтерпретації результатів та для прийняття рішень щодо лікування. Тож, ШІ може допомогти лікарям прискорити процес діагностики і забезпечити більш точне та ефективне лікування пацієнтів [1, 5, 9].

Розробка і підбір медикаментів за допомогою штучного інтелекту

Величезний хімічний простір, що містить $>10^{60}$ молекул, сприяє розробці великої кількості молекул ліків. Однак відсутність передових технологій обмежує процес розробки ліків, роблячи його трудомістким і дорогим завданням, яке можна вирішити за допомогою ШІ. ШІ може розпізнавати влучні та провідні сполуки, а також забезпечувати швидшу перевірку цільового препарату та оптимізацію дизайну структури препарату. Незважаючи на свої переваги, штучний інтелект стикається з деякими серйозними проблемами, такими як масштаб, зростання, різноманітність і невизначеність даних. Набори даних, доступні для розробки ліків у фармацевтичних компаніях, можуть включати мільйони сполук, і традиційні інструменти машинного навчання можуть не працювати з такими типами даних. Обчислювальна модель на основі кількісного співвідношення структура-активність (Quantitative structure-activity relationship, QSAR) може швидко передбачити велику кількість сполук або простих фізико-хімічних параметрів, таких як log P або log D [1].

Прогнозування фізико-хімічних властивостей. Фізико-хімічні властивості, такі як розчинність, коефіцієнт розподілу (log P), ступінь іонізації та внутрішня проникність препарату, опосередковано впливають на його фармакокінетичні властивості та сімейство цільових рецепторів і, отже, повинні враховуватися при розробці нового препарату. Для прогнозування фізико-хімічних властивостей можна використовувати різні інструменти на основі ШІ. Наприклад, машинне навчання використовує великі набори даних, отримані під час складної оптимізації, виконаної раніше, для навчання програми. Алгоритми для дизайну ліків включають молекулярні

дескриптори, такі як рядки SMILES, вимірювання потенційної енергії, щільність електронів навколо молекули та координати атомів у 3D, щоб генерувати можливі молекули через DNN і таким чином передбачати їх властивості [1].

Прогноз біоактивності. Ефективність молекул ліків залежить від їх спорідненості з цільовим білком або рецептором. Молекули ліків, які не виявляють жодної взаємодії або спорідненості з цільовим білком, не зможуть забезпечити потрібну терапевтичну дію. У деяких випадках також можливо, що розроблені молекули ліків взаємодіють з небажаними білками або рецепторами, що призводить до токсичності. Отже, спорідненість зв'язування лікарського засобу з мішенню (drug target binding affinity, DTBA) є життєво важливою для прогнозування взаємодії лікарський засіб-мішень. Методи на основі штучного інтелекту можуть вимірювати спорідненість зв'язування лікарського засобу, враховуючи особливості або схожість препарату та його цілі. Взаємодії на основі ознак розпізнають хімічні компоненти препарату та його цілі для визначення ознак. Навпаки, у взаємодії на основі подібності враховується подібність між препаратом і ціллю [1].

Прогнозування структури цільового білка. Під час розробки молекули ліків важливо призначити правильну мішень для успішного лікування. Численні білки беруть участь у розвитку захворювання, і в деяких випадках вони надмірно експресуються. Отже, для вибіркового націлювання на захворювання життєво важливо передбачити структуру цільового білка для розробки молекули ліків. ШІ може допомогти у відкритті ліків на основі структури, передбачаючи тривимірну структуру білка, оскільки конструкція відповідає хімічному середовищу ділянки цільового білка, таким чином допомагаючи передбачити вплив сполуки на мішень разом із міркуваннями безпеки перед їх синтезом або виробництво. Інструмент штучного інтелекту AlphaFold, який базується на DNN, використовувався для аналізу відстані між сусідніми амінокислотами та відповідними кутами пептидних зв'язків для прогнозування 3D-структури цільового білка та продемонстрував чудові результати, правильно передбачивши 25 із 43 структур [1].

Помилки та ризики при використанні штучного інтелекту у медицині

Системи ШІ в медицині можуть нести певні ризики та викликати помилки. Ці помилки можуть бути небезпечними для пацієнта, якщо система рекомендує неправильний препарат, не помічає пухлину при рентгенологічному скануванні або неправильно виділяє лікарняне ліжко одному пацієнту замість іншого, неправильно оцінивши ризик та пріоритетність захворювань. Звичайно, помилки і так трапляються в медицині через людський фактор і без участі ШІ. Але помилки ШІ потенційно відрізняються принаймні з двох причин: 1) реакція суспільства на помилки програмного забезпечення, що призводять до травмування пацієнта, будуть більш агресивними, аніж коли цю помилку допустила б людина; 2) якщо конкретна система ШІ набуде широкого використання у медичних закладах, одна її проблема чи не доопрацювання в алгоритмі може призвести до масовості травмувань через неї, на відміну від помилок одного лікаря [2, 8].

Висновки. Штучний інтелект має значний потенціал для використання в медицині, зокрема, в діагностуванні та лікуванні захворювань. Завдяки своїм можливостям, ШІ дозволяє точніше та швидше виявляти симптоми та характеристики хвороб, а також аналізувати велику кількість медичних даних. Однією з головних переваг використання ШІ в медицині є зменшення людського фактору, що забезпечує більш об'єктивну та точну діагностику. Також машинне навчання дозволяє ШІ навчитися розпізнавати патології на знімках, що спрощує роботу лікарів та зменшує ймовірність помилок у діагнозі. Водночас, використання ШІ в медицині повинно здійснюватися з урахуванням етичних, юридичних та безпекових аспектів. Крім того, ШІ не може замінити роль лікарів, які залишаються необхідними для інтерпретації результатів та для прийняття рішень щодо лікування. Отже, використання ШІ в медицині може бути корисним інструментом для поліпшення діагностики та лікування захворювань, проте воно повинне здійснюватися з обережністю та уважністю з урахуванням всіх можливих ризиків та обмежень.

Список літератури

1. Paul D., Sanap G., Shenoy S., Kalyane D., Kalia K., Tekade R.K. Artificial intelligence in drug discovery and development // Drug Discov Today, 2021 Jan., vol. 26(1), pp. 80-93. doi: 10.1016/j.drudis.2020.10.010 – URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7577280/

2. Землянська, О. В., Страшнова А.С. Наслідки впровадження штучного інтелекту в сучасну медицину // XXVI Всеукраїнська науково-методична конференція «Проблеми охорони праці, промислової та цивільної безпеки», 19 травня 2022 року, м. Київ, Україна. – Київ: КПІ ім. Ігоря Сікорського, 2022. https://ela.kpi.ua/handle/123456789/48619

3. Застосування штучного інтелекту в медицині: ефективна діагностика і створення нових ліків – URL: https://aiconference.com.ua/uk/news/primenenie-iskusstvennogo-intellekta-v-meditsine-effektivnaya-diagnostika-i-sozdanie-novih-lekarstv-92604

4. Merative Blog – URL: https://www.merative.com/blog

5. Artificial intelligence in healthcare (Date retrieved: 9 April 2023 12:45 UTC) – URL: https://en.wikipedia.org/wiki/Artificial_intelligence_in_healthcare

6. DeepMind (Date retrieved: 9 April 2023 12:51 UTC) – URL: https://en.wikipedia.org/wiki/DeepMind

7. Babylon Health (Last updated: Feb. 2023) – URL: https://www.nhsforsale.info/private-providers/babylon-new/

8. Price II W.N. Risks and remedies for artificial intelligence in health-care, 2019 – URL: https://www.brookings.edu/research/risks-and-remedies-for-artificial-intelligence-in-health-care/

9. Дмитрик К. Як штучний інтелект застосовується в медицині: поточні досягнення та перспективи // Журнал Аптека online, № 11(1382), 20 Березня 2023 – URL: https://www.apteka.ua/article/661210

УДК 004.94

Б.О. Удудяк, Д.О. Берестенко, Є.В. Мелешко, М.С. Якименко
elismeleshko@gmail.com
Центральноукраїнський національний технічний університет, м. Кропивницький

ДОСЛІДЖЕННЯ СУЧАСНИХ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ З ВІДКРИТИМ КОДОМ ДЛЯ ГЕНЕРАЦІЇ ТЕКСТІВ

Системи та засоби штучного інтелекту – галузь науки, яка займається теоретичними дослідженнями, розробкою та застосуванням алгоритмічних та програмно-апаратних систем і комплексів з елементами штучного інтелекту та моделюванням інтелектуальної діяльності людини.

Системи штучного інтелекту з відкритим кодом для генерації текстів стали дуже популярними останнім часом, завдяки швидкому розвитку нейронних мереж і глибокого навчання. Ці системи використовуються в різних галузях, як-от медицина, фінанси, транспорт та інші, але їхнє застосування в генерації текстів є найцікавішим.

Системи штучного інтелекту можуть генерувати тексти на основі заданих параметрів, наприклад, на певну тему, із заданим стилем або тоном. Ці системи працюють на основі глибокого навчання, яке дає їм змогу "вчитися" на основі величезної кількості даних, що робить їх дедалі ефективнішими. Системи штучного інтелекту можуть генерувати різні типи текстів, включно зі статтями, розповідями, листами тощо.

Найбільш популярними системами штучного інтелекту з відкритим кодом для генерації текстів є GPT-2, GPT-3 та BERT. Також досить цікавими у цій галузі є моделі Transformer-XL та CTRL.

GPT-2 (Generative Pre-trained Transformer 2) – це мовна модель, розроблена OpenAI, яка привернула широку увагу своєю здатністю створювати високоякісний текст природною мовою. Маючи 1,5 мільярди параметрів, це одна з найбільших доступних мовних моделей, яка була навчена на великому масиві текстових даних із використанням архітектури трансформатор, що є дуже ефективним для обробки послідовних даних, таких як текст. GPT-2 продемонструвала надзвичайну продуктивність у широкому діапазоні мовних завдань, включаючи завершення тексту, створення вмісту та чат-ботів. Здатність цієї моделі генерувати відповідний контексту та зв'язний текст зробила її цінним ресурсом для розробників і дослідників, які цікавляться обробкою природної мови. Однією з найважливіших переваг GPT-2 є її відкритий сирцевий код, який був опублікований OpenAI на GitHub. Це дозволяє розробникам точно налаштувати модель для конкретних завдань або використовувати її як відправну точку для своїх мовних моделей. Попередньо навчені моделі, надані OpenAI, можна використовувати для створення тексту та інших мовних завдань, що робить їх цінним ресурсом для різноманітних програм. Незважаючи на численні переваги, випуск GPT-2 також викликав занепокоєння щодо можливого неправильного використання моделі для створення оманливого або шкідливого вмісту. Щоб вирішити ці проблеми, OpenAI спочатку вирішив не випускати повну версію моделі для громадськості. Однак згодом вони випустили зменшені версії моделі та дослідницьку статтю з описом її можливостей. Підсумовуючи, GPT-2 є значним прогресом у галузі обробки природної мови та має потенціал для використання для широкого кола мовних завдань. Її відкритий сирцевий код і попередньо навчені моделі роблять її цінним ресурсом для розробників і дослідників, зацікавлених у створенні природної мови. Однак можливість неправильного використання моделі підкреслює необхідність відповідального використання та етичних міркувань у розробці та розгортанні систем штучного інтелекту для генерації текстів.

GPT-3 є наступною версією GPT-2, що вже має 175 мільярдів параметрів на відміну від попередниці, та більш якісно генерує тексти природною мовою. GPT-3 також працює на основі технології глибокого навчання, яка дає їй змогу "вчитися" на основі величезної кількості даних. Однією з переваг GPT-3 є її здатність генерувати тексти на основі невеликої кількості даних. Це означає, що система може генерувати тексти на нову тему, навіть якщо їй не були надані дані на цю тему.

Інша популярна система штучного інтелекту з відкритим кодом для генерації текстів – **BERT**, яку розробили в Google, використовується для розв'язання завдань, пов'язаних із природною мовою, таких як пошук даних, генерація текстів, сумаризація текстів та інших завдань. BERT (Bidirectional Encoder Representations from Transformers) – це модель глибокого навчання з відкритим кодом, що використовує контекстуальні векторні представлення слів для досягнення високої точності в розумінні та генерації природної мови в різноманітних завданнях, включно з машинним перекладом, розпізнаванням мовлення і відповіддю на запитання. BERT використовує глибокі нейронні мережі для розуміння природної мови і демонструє високу точність у різних завданнях, пов'язаних із природною мовою.

Transformer-XL – це модель зі штучним інтелектом, яка базується на трансформерах та володіє здатністю до генерації тексту з довшими контекстами порівняно з іншими моделями. Вона була розроблена компанією Google Research та демонструє високу ефективність на багатьох завданнях генерації тексту, таких як машинний переклад, відповіді на запитання та інші. Transformer-XL має кілька унікальних особливостей, включаючи підтримку більш довгих контекстів за рахунок використання механізму пам'яті та здатність до адаптації до нових завдань через зміну вхідних даних. Ця модель зі штучним інтелектом може бути використана для різних завдань, таких як генерація тексту, машинний переклад, розпізнавання мови та інші.

Крім того, вона може бути використана для створення більш складних систем штучного інтелекту з використанням глибокого навчання. Transformer-XL є частиною широкого спектру інструментів з відкритим кодом для генерації тексту та інших завдань штучного інтелекту. Вона може бути використана дослідниками та розробниками для створення нових методів генерації тексту та відповідних застосунків.

CTRL – також як і Transformer-XL є моделю зі штучним інтелектом, яка базується на трансформерах та володіє більшою точністю та кращою здатністю до контролю згенерованого тексту в порівняні з іншими моделями. Як і GPT-2 та GPT-3 була розроблена компанією OpenAI та демонструє високу ефективність в багатьох завданнях генерації тексту, таких як створення діалогових систем, створення тексту для веб-сторінок тощо. Одна з унікальних особливостей CTRL – це здатність до контролю згенерованого тексту. CTRL може бути налаштована на зміну стилю, теми або на генерацію тексту з заданим тоном. Одною із головних відмінностей CTRL від інших моделей генерації природньої мови є те, що вона здатна зберігати та використовувати інформацію про контекст попередніх слів для генерації більш змістовних речень. CTRL може виконувати завдання генерації тексту у багатьох областях, таких як, наприклад, автоматичне підписування електронної пошти, генерація новин, генерація машинних перекладів тощо. Недоліками CTRL можуть бути нестабільність під час навчання, вимоги до потужного обчислювального обладнання та складність розуміння внутрішньої структури моделі. Однак, ці недоліки можуть бути подолані шляхом додаткового дослідження та оптимізації.

Деякі основні переваги, які може надавати використання штучного інтелекту для генерації текстів:

- прискорення процесу написання будь-яких текстів для вивільнення часу на інші завдання;
- допомога авторам блогів і сайтів у створенні контенту;
- створення текстових шаблонів для більш зручної роботи по створенню контенту;
- поліпшення якості корпоративних повідомлень і листів, спрямованих на взаємодію з клієнтами;
- створення автоматичних звітів і документів, що спрощує процес управління бізнесом;
- допомога компаніям у створенні привабливих та ефективних рекламних оголошень;
- допомога у створенні цікавих і зрозумілих навчальних матеріалів для учнів та студентів;
- машинний переклад – моделі генерації природньої мови можуть бути використані для автоматичного перекладу тексту на різні мови;
- створення діалогових систем, які можуть комунікувати з користувачами і відповідати на їх запитання, наприклад, для часткової автоматизації технічної підтримки;
- автоматичний аналіз тексту, наприклад, визначення тону тексту, категоризації тематики і т.д.;
- тощо.

У підсумку можна зробити висновок, що сучасні системи штучного інтелекту з відкритим кодом для генерації текстів можуть допомогти полегшити роботу у багатьох галузях. Системи штучного інтелекту можуть полегшити роботу і скоротити час написання майже будь-яких текстів. На сьогоднішній день існує великий вибір моделей штучного інтелекту для генерації тексту на природній мові і вивчення їх та подальше використання є доцільним.

Список літератури

1. Alammar J. The Illustrated GPT-2 (Visualizing Transformer Language Models), 2019. – URL: https://jalammar.github.io/illustrated-gpt2/
2. GPT-2: 1.5B release, 2019 – URL: https://openai.com/research/gpt-2-1-5b-release
3. GPT-2, 2020 – URL: https://github.com/openai/gpt-2
4. GPT-3: Language Models are Few-Shot Learners, 2020 – URL: https://github.com/openai/gpt-3
5. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019, P. 4171-4186. – doi: https://doi.org/10.48550/arXiv.1810.04805
6. Yang Z., Le Q. Transformer-XL: Unleashing the Potential of Attention Models, 2019. – URL: https://ai.googleblog.com/2019/01/transformer-xl-unleashing-potential-of.html
7. Dai Z., Yang Z., Yang Y., Carbonell J., Le Q. V., Salakhutdinov R. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context, 2019. – doi: https://doi.org/10.48550/arXiv.1901.02860
8. Keskar N.S., McCann B., Varshney L.R., Xiong C., Socher R. CTRL: A Conditional Transformer Language Model for Controllable Generation, 2019. – doi: https://doi.org/10.48550/arXiv.1909.05858
9. Keskar N. S., McCann B., Varshney L., Xiong C., Socher R. CTRL – A Conditional Transformer Language Model for Controllable Generation, 2021. – URL: https://github.com/salesforce/ctrl

УДК 004.056

А.В. Грицак[1], Я.Ю. Яремчук[1], В.М. Білоус[1],
grytsak.a.v@gmail.com,
[1]*Вінницький національний технічний університет, м. Вінниця*

ПІДВИЩЕННЯ СТІЙКОСТІ ВІРТУАЛЬНИХ СЕРВЕРІВ ДО DDOS-АТАК НА ОСНОВІ МАСШТАБУВАННЯ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ КЛАСТЕРА

На сьогодні широкого застосування і популярності набули DDoS-атаки, мета яких – повне припинення роботи сервера. Середні збитки від DDoS-атак оцінюються у світі в 50 тисяч доларів для невеликих організацій і майже у 500 тис. доларів для великих підприємств. Усунення наслідків DDoS-атаки потребує додаткового робочого часу співробітників, відволікання ресурсів з інших проектів на забезпечення безпеки, розробки плану оновлення програмного забезпечення, модернізації обладнання тощо.

Метою роботи є розробка алгоритму для покращення захисту віртуальних контейнеризованих серверів від DDoS-атак.

Об'єктом дослідження є процес захисту віртуальних контейнеризованих серверів від DDoS-атак.

Предметом є дослідження є методи та засоби підвищення стійкості віртуальних контейнеризованих серверів до DDoS-атак.

DDoS-атака – комплекс дій, здатний повністю або частково вивести з ладу інтернет-ресурс. В якості жертви може виступати практично будь-який інтернет-ресурс, наприклад веб-сайт, ігровий сервер або державний ресурс.

Використання маштабованості (scalability) є важливим чинником у захисті від DDoS-атак, оскільки такі атаки можуть залежати від масштабування нападу для досягнення своєї цілі. Масштабованість – здатність пристрою збільшувати свої можливості шляхом нарощування кількості функціональних блоків, що виконують одні й ті ж завдання. У випадку, коли віртуальний сервер не справляється з навантаженням, йому додаватимуться додаткові об'єми потужностей у вигляді додаткових ядер процесора, контейнерів, віртуальних машин, мегабайт оперативної пам'яті тощо. Автоматичне масштабування виключає з цього процесу участь людини, що значно заощаджує час, ресурси, і, як наслідок, – кошти.

Властивість масштабованості актуальна, перш за все тому, що умови сучасного бізнесу змінюються настільки швидко, що унеможливлюють довгострокове планування, яке потребує всебічного і тривалого аналізу вже застарілих даних, навіть для тих організацій, які здатні це собі дозволити. Натомість приходить стратегія поступового, крок за кроком, нарощування потужностей інформаційних систем. З іншого боку, зміни у технології приводять до появи все нових рішень і зниження цін на апаратне забезпечення, що потенційно робить архітектуру інформаційних систем більш гнучкою. Одночасно розширюється міжоперабельність, відкритість програмних і апаратних продуктів різних виробників, хоча поки їх зусилля, спрямовані на відповідність стандартам, узгоджені лише у вузьких секторах ринку. Без урахування цих факторів споживач не зможе скористатися перевагами нових технологій, що не заморожуючи коштів, вкладених у недостатньо відкриті, або такі, що стали безперспективними технології.

Нерівномірне навантаження на сервер – часте явище. Наприклад, вночі сервер може простоювати, а вранці навантаження може зрости.

При включенні автомасштабування проводиться моніторинг зайнятих ресурсів сервера. Зміни доступних ресурсів і крок цих змін залежить від правил, налаштованих користувачем і цілі автомасштабування. Для одних проектів важливіше мати можливість швидко масштабуватися, для інших – заощадити більше коштів.

Масштабованою система може називатися тільки у тому випадку, якщо є можливість розширення ресурсів пропорційно вимогам завдань за рахунок зміни апаратних можливостей.

Зазвичай розглядають два види масштабування:

– вертикальне масштабування – збільшення продуктивності кожного компонента системи з метою підвищення загальної продуктивності;

– горизонтальне масштабування – розбиття системи на більш дрібні структурні компоненти та рознесення їх по окремим фізичним машинам (або їх групам), і (або) збільшення кількості серверів, паралельно виконують одну й ту ж функцію.

Також іноді виділяють прогнозне масштабування – це функція, яка аналізує історичні моделі трафіку і становить на їх підставі прогнози для подальшого планування змін кількості обчислювальних ресурсів у певні моменти часу в майбутньому.

Завдяки функції прогнозного масштабування забезпечується більш швидке, просте і точне виділення ресурсів, що сприяє зниженню витрат і підвищенню швидкості відгуку додатків.

Використовуючи горизонтальне масштабування як таке, що легко імплементується в інфраструктуру без змін вихідного програмного коду і може бути глобально застосовано як частина вирішення проблеми DDoS атак для клієнтів хмарних хостингів.

Корисною властивістю автомасштабування є можливість максимальної економії ресурсів. За

допомогою автомасштабування можна зменшувати або збільшувати обсяг ресурсів доступних серверу.

Запропоновано удосконалений алгоритм, який відслідковує навантаження різних значущих показників операційної системи і, у залежності від результатів моніторингу, вносить зміни до інфраструктури, підвищуючи її стійкість.

Значущими параметрами для моніторингу є такі параметри:

– CPU (Central Processor Unit – центральний процесор сервера);

– RAM (Random Access Memory);

– Load Average (Середнє значення навантаження).

Певна кількість операцій введення-виведення у будь-якій конфігурації системи може варіюватися в залежності від змінних, які тестувальник вводить в програму, в тому числі співвідношення операцій читання і запису, поєднання послідовного і випадкового доступу моделей, кількість робочих потоків і глибини черги, а також розміри блоків даних.

Особливістю удосконаленого алгоритму є контроль за процесом з боку адміністратора. Моніторинг відбувається по трьох основних критеріях, для економії ресурсів скрипт буде запускатись циклічно один раз на декілька хвилин.

У першу чергу буде запускатись дослідження середовища по трьом основним вищеописаним метрикам – скільки віртуальних ядер у контейнері, яка кількість оперативної пам'яті. Кількість ядер процесора чисельно дорівнює максимально допустимому значенню Load Average.

Також CPU навантаження не має досягати значення 100%. При досягненні рівня навантаження у 100 відсотків процесору не вистачатиме потужності та потоків для обробки і опрацювання усіх операцій, тому операції ставатимуть у чергу, що спричинить довге виконання задач обчислювальною машиною.

Щодо параметру операційної пам'яті бажано не допускати перевищення реального значення навантаження більше ніж 70-90%. Саме ці значення будуть взяті до уваги при розробці програмного продукту. При досягненні завантаження більше 90 відсотків серверу доведеться підвантажувати дані з дискового накопичувача або SWAP.

Додатково при виборі розміщення SWAP потрібно враховувати, що не усі файлові системи дозволяють використовувати пряму адресацію блоків SWAP-файлу.

Наступним кроком є послідовні циклічні перевірки і порівняння основних значущих параметрів з реальним показниками продуктивності контейнерів.

Після кожної перевірки відбуваються зміни в деплоймент файл (якщо показники вийшли за межі встановленої норми; якщо ж ні – то зміни не відбуваються), який зберігається у .json форматі. Після збереження змін згідно з деплоймент файлом відбувається відповідне масштабування інфраструктури, яка знаходиться на моніторингу.

Якщо внаслідок автоматичного масштабування кількість контейнерів буде перевищувати один, то навантаження буде балансуватись за алгоритмом Round Robin.

У комп'ютерах балансування навантаження розподіляється між декількома обчислювальними ресурсами, такими як комп'ютерні кластери, мережі, центральні процесори або диски.

Також, для зручності і практичності конфігурування автоматичного масштабування можливе зберігання конфігурації у GitHub репозиторії. У такому випадку, перед кожною перевіркою нові налаштування будуть завантажуватись з репозиторія і масштабування відбуватиметься (чи не відбуватиметься) з їх урахуванням.

Висновки. Отже, розроблений алгоритм автоматичного масштабування серверних кластерних ресурсів для протидії DDoS атакам виконує моніторинг основних критично важливих для функціонування сервера метрик, у разі необхідності виконує масштабування ресурсів без участі людини у процесі, а також повідомляє про свої дії адміністратора системи. На початку атаки показники моніторингу систем контейнерів починають зростати. При досягненні критичних значень система автоматично починає масштабуватись. Наступним кроком є автоматичне повідомлення адміністраторів системи про зміни, які відбулись. За рахунок масштабування адміністратори також мають додатковий час для ідентифікації проблеми, або, як раптовий сплеск навантаження, або як DoS/DDoS атаки на систему і прийняття рішення щодо заходів. Таким чином, завдяки вчасному інформуванню і вживанню заходів життєдіяльність обчислювальної системи може бути швидко, ефективно і з найменшими втратами врятована, атака відбита і загроза ліквідована.

УДК 004.4277.2.056.55

Д.В. Євграфов[1], Ю.Є. Яремчук[1]
ramgraf@bigmir.net
[1]*Вінницький національний технічний університет, м. Вінниця*

АСИМПТОТИЧНО ОПТИМАЛЬНИЙ ПРИЙМАЧ СИГНАЛІВ ПОБІЧНИХ ВИПРОМІНЮВАНЬ З ЕКРАНІВ МОНІТОРІВ НА РІДИННО-КРИСТАЛІЧНИХ СТРУКТУРАХ

Для більшості моніторів електронно-обчислювальної техніки (ЕОТ) виникнення електромагнітного поля (ЕМП) є побічним і небажаним результатом їх роботи. Подібні паразитні електромагнітні поля називають побічними електромагнітними випромінюваннями. Крім того, монітори спричиняють наведення у оточуючих їх предметах, що проводять електричний струм. Існування побічних електромагнітних випромінювань і наведень (ПЕМВІН) робить потенційно можливим перехоплення інформації за допомогою спеціальних технічних засобів розвідки противника (СТЗРП), що «зчитують», наприклад, виведену на екран монітора інформацію. Дуже важливо оцінити потенційні можливості сучасних СТЗРП з виявлення сигналів ПЕМВІН з екрану моніторів ЕОТ, оскільки вони постійно вдосконалюються як розширенням частотного діапазону перехопленої інформації, так і способами її оброблення.

Теорія оптимального приймання сигналів на фоні внутрішніх шумів приймача стверджує, що найкращим виявлювачем сигналів є той, що найбільш погоджений із сигналом ПЕМВІН. В умовах апріорної невизначеності щодо більшості параметрів сигналу витоку подібним приймачем сигналів є асимптотично байєсівський виявляч сигналів ПЕМВІН – приймач, що наближається до оптимального при збільшенні співвідношення сигнал/шум на виході його лінійної частини. Такі приймачі застосовують алгоритми максимальної правдоподібності (АМП), сутність яких полягає у тому, що невідомі параметри сигналу, що приймається, замінюють їх максимально правдоподібними оцінками, в разі, коли кількість невідомих параметрів є кінцевою. Поставимо на меті обгрунтувати структуру СТЗРП, що реалізує АМП. Нехай зображення екрану монітору ЗОТ на рідинно-кристалічних структурах (РКС) є незмінним протягом часу аналізу $T_{\text{а}} \in [T_{\text{а min}}, T_{\text{а max}}]$, $T_{\text{а min}}, T_{\text{а max}}$ – нижня і верхня границі часу аналізу. Тоді на інтервалі часу $t \in [0, T_{\text{а}}]$, $T_{\text{а}} >> T_{\text{к}}$, $T_{\text{к}}$ – період слідування кадрів розгортки монітору, аналізують реалізацію сигналу

$$x(t) = \begin{cases} n(t), & \text{коли сигналу немає,} \\ n(t) + s(t, \mathbf{v}_0), & \text{за наявності сигналу,} \end{cases} \tag{1}$$

де $n(t)$ – гаусівський процес із нульовим середнім і кореляційною функцією

$$B(t, \tau) = M[n(t) n(t + \tau)], \tag{2}$$

$\mathbf{v}_0$ – вектор дійсних значень параметрів сигналу ПЕМВІН $s(t, \mathbf{v}_0)$. Передбачають, що $s(t, \mathbf{v}_0)$ є відомою функцією часу t та вектора з n- параметрами $\mathbf{v}_0 \in \boldsymbol{\Theta}$, $\boldsymbol{\Theta}$ – простір параметрів, у яких інтервали значень кожного параметра кінцеві $\Theta_i \in [\Theta_{i1}, \Theta_{i2}], i = 1...n$, або $\boldsymbol{\Theta} \in [\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2]$; $\boldsymbol{\Theta}_1$ – вектор нижніх значень параметрів; $\boldsymbol{\Theta}_2$ – вектор верхніх значень параметрів.

Рішення про наявність сигналу $s(t, \mathbf{v}_0)$ ухвалюють, коли абсолютний максимум функціоналу співвідношення правдоподібності (ФСП)

$$\sup_{\mathbf{v} \in \boldsymbol{\Theta}} [l(T_{\text{а}}, \mathbf{v})] \geq h, \tag{3}$$

а про те, що сигналу немає – коли $\sup_{\mathbf{v} \in \boldsymbol{\Theta}} [l(T_{\text{а}}, \mathbf{v})] < h$, де h – деякий поріг виявлення, що залежить від критерію оптимальності,

$$\begin{aligned} l(T_{\text{а}}, \mathbf{v}) &= \exp[L(T_{\text{а}}, \mathbf{v})], \\ L(T_{\text{а}}, \mathbf{v}) &= \int_0^{T_{\text{а}}} x(t) V(t, \mathbf{v}) \, \mathrm{d}t - \frac{1}{2} \int_0^{T_{\text{а}}} s(t, \mathbf{v}) V(t, \mathbf{v}) \, \mathrm{d}t, \end{aligned} \tag{4}$$

а $V(t, \mathbf{v})$ – розв'язок інтегрального рівняння:

$$\int_0^{T_{\text{а}}} B(t, \tau) V(\tau, \mathbf{v}) \, \mathrm{d}\tau = s(t, \mathbf{v}), \qquad t \in [0, T_{\text{а}}]. \tag{5}$$

Мета: знайти (4) рішенням інтегрального рівняння (5), та отримати АМП (3) для довільного вектору невідомих параметрів $\mathbf{v}$ сигналу ПЕМВІН і невідомого часу незмінного зображення на екрані монітору $T_{\text{a}} \in [T_{\text{a min}}, T_{\text{a max}}]$.

Об'єктом дослідження є синтез оптимальних алгоритмів виявлення сигналів в умовах суттєвої апріорної невизначеності щодо його параметрів.

Предметом є синтез асимптотично оптимального алгоритму виявлення сигналів ПЕМВІН з екранів моніторів на РКС.

У результаті дослідження знайдена структура сучасного СТЗРП, здатного перехоплювати сигнали ПЕМВІН з екранів моніторів на РКС. Якщо картинка на моніторі статична, то її можна подати у вигляді суми напруг m - періодичних i - х послідовностей сигналів, для коефіцієнтів ряду Фур'є:

$$s(t, \mathbf{v}_0) = \mathbf{1}(T_{\text{a}0} - t) \sum_{k=-\infty}^{\infty} \sum_{i=1}^{m} \frac{U_i}{\pi k} \sin\left(\frac{\pi k \tau_i}{T_{\text{к}}}\right) \cos\left[\frac{2\pi k}{T_{\text{к}}}[t - t_{зi}]\right], \tag{6}$$

$\mathbf{1}(x)$ – одинична функція, $T_{\text{a}0}$ – істинна тривалість незмінного зображення на екрані монітору. Нульову гармоніку у подальшому виключаємо із розгляду, оскільки вона не поширюється простором. Тоді (6) для довільного вектору невідомих параметрів $\mathbf{v}$ сигналу ПЕМВІН можна подати через квадратури амплітуд:

$$a_{kc} = \sum_{i=1}^{m} \frac{2U_i}{\pi k} \sin\left(\frac{\pi k \tau_i}{T_{\text{к}}}\right) \cos\left(\frac{2\pi k t_{зi}}{T_{\text{к}}}\right), \; a_{ks} = \sum_{i=1}^{m} \frac{2U_i}{\pi k} \sin\left(\frac{\pi k \tau_i}{T_{\text{к}}}\right) \sin\left(\frac{2\pi k t_{зi}}{T_{\text{к}}}\right)$$

у вигляді:

$$s(t, \mathbf{v}) = \mathbf{1}(T_{\text{a}0} - t) \sum_{k=1}^{\infty} \left[a_{kc} \cos\left(\frac{2\pi k t}{T_{\text{к}}}\right) + a_{ks} \sin\left(\frac{2\pi k t}{T_{\text{к}}}\right) \right]. \tag{7}$$

Підстановка (1) і (7) у (4) та максимізація у (7) за невідомими a_{kc}, a_{ks} дозволяє отримати ФСП для АМП, в якому подолано невизначеність щодо квадратур амплітуд:

$$L(T_{\text{a}}) = \frac{4}{N_0 T_{\text{a}}} \sum_{k=1}^{K} \left[\left(\int_0^{T_{\text{a}}} x(t) \cos\left(\frac{2\pi k t}{T_{\text{к}}}\right) \text{d}t \right)^2 + \left(\int_0^{T_{\text{a}}} x(t) \sin\left(\frac{2\pi k t}{T_{\text{к}}}\right) \text{d}t \right)^2 \right]. \tag{8}$$

Сумісний алгоритм виявлення ПЕМВІН і оцінювання інтервалу часу, на якому вони не змінюються, полягає у порівнянні абсолютного максимуму від (8) з порогом виявлення h, при якому приймається рішення про виявлення ПЕМВІН з екрану монітору, коли

$$\sup_{T_{\text{a min}} \le T_{\text{a}} \le T_{\text{a max}}} L(T_{\text{a}}) > h,$$

і рішення про те, що сигналу ПЕМВІН немає – коли

$$\sup_{T_{\text{a min}} \le T_{\text{a}} \le T_{\text{a max}}} L(T_{\text{a}}) \le h.$$

Висновки

1. Синтез алгоритму максимальної правдоподібності у вигляді асимптотично баєсівського виявлювача сигналів побічних електромагнітних випромінювань і наведень можливий лише, коли кількість невідомих квадратурних амплітуд сигналу витоку a_{kc}, a_{ks} є кінцевою – $2K$.

2. Кінцева кількість невідомих квадратурних амплітуд під час аналізу алгоритмів виявлення сигналів побічних електромагнітних випромінювань з екрану монітору не дозволяє застосувати відомий апарат однокомпонентних марковських процесів, пригідний для розривних сигналів.

3. Подальший розв'язок задачі аналізу алгоритмів виявлення сигналів побічних електромагнітних випромінювань і наведень в спеціалізованому технічному засобі розвідки противника потребує відшукування розподілень абсолютних максимумів нестаціонарних диференційованих у середньоквадратичному процесів.

УДК 004.4

Д.П. Присяжний[1], П.В. Павловський[1], В.В. Саврацький[1]
dimpris@gmail.com
[1]*Вінницький національний технічний університет, м. Вінниця*

УДОСКОНАЛЕННЯ МЕТОДУ ВИЯВЛЕННЯ ПОРУШЕННЯ КОНФІДЕНЦІЙНОСТІ ФАЙЛІВ ВИХІДНОГО КОДУ ЗА РАХУНОК ЗБІЛЬШЕННЯ ІНТЕРВАЛІВ ОБРОБКИ ТА ЗАСТОСУВАННЯ ТРАНСПОЗИЦІЇ

Захист авторських прав на програмний код як об'єкт інтелектуальної власності у сфері розробки програмного забезпечення нині є серйозною проблемою для розробників, адже існуючі механізми захисту прав на володіння об'єктом інтелектуальної власності не є достатньо адаптованими до особливостей даної предметної області. Як не існує достатньо відповідних цим особливостям юридичних механізмів, так і програмних засобів, що дозволили б чітко визначати факт порушення автентичності файлів вихідного коду програм. Оскільки програмний код з точки зору законодавства у сфері захисту авторського права розглядається як звичайний текст, то відповідно для пошуку ознак порушення автентичності файлів та відповідно доведення факту несанкціонованого копіювання із подальшим присвоєнням авторства використовуються ті ж рішення, що й для порівняння звичайних текстів.

Ще одним аспектом, що є надзвичайно важливим для алгоритмів порівняння файлів вихідного коду, є швидкість проведення перевірки. При проведенні судових чи внутрішніх службових розслідувань нині керуються набором методів та механізмів, що повинні виявляти характерний стиль автора і на основі цих доказів свідчити про автентичність чи неавтентичність файлу. Однак, дані методи орієнтуються на ручне їх виконання, тож час обробки результатів таких досліджень є непристойно великий. Таким чином, існує гостра потреба у пошуку рішення, що дозволило б проводити перевірку автентичності файлів вихідного коду програм за значно менший час аніж існуючі методи та базуватиме свої висновки на певній чіткій метриці, що не даватиме простору для ентропії.

Метою роботи є удосконалення методу виявлення порушення конфіденційності файлів вихідного коду програм.

Об'єктом дослідження є процес захисту від несанкціонованого копіювання порушення автентичності файлів вихідного коду програм.

Предметом є методи та засоби порівняння версій файлів вихідного коду програм.

У результаті дослідження виконано аналіз існуючих систем, що дозволяють проводити порівняння версій файлів та визначено, що ті алгоритми та методи, що у них використано, не здатні забезпечити необхідну об'єктивність результатів порівняння версій у контексті розслідувань випадків порушення автентичності файлу вихідного коду.

Було проаналізовано критерії до ефективності алгоритмів порівняння версій та визначено що ключовими для оцінювання успішності розробки вдосконалення методу є похибка обчислень відносно оригінального алгоритму та часова ефективність алгоритму. Визначено, що відстань Дамероу-Левенштейна є кращим рішенням при виборі метрики оцінки відмінності між послідовностями кодів файлів вихідного коду оригіналу та копії, оскільки обчислення даної метрики передбачає операції, що дозволяють зробити оцінку відмінності більш точною та об'єктивною, а також враховувати особливості предметної області, як от транспозицію у непорядкових індексах.

Виявлено, що алгоритм визначення відстані Дамероу-Левенштейна через те, що початково був розроблений для порівняння текстових послідовностей, не враховує регістр буквених символів та визначає однакові літери у різних регістрах як різні символи. У кодовій послідовності регістр літер відіграє значно меншу роль у зміні змісту коду в порівняння із звичайним текстом. Таким чином, для досягнення більшої точності обрахунку відстані Дамероу-Левенштейна між двома послідовностями вихідного коду пропонується додати операцію виведення усіх символів обох послідовностей до одного регістру перед початком їх порівняння за алгоритмом Вагнера-Фішера.

Незалежно від того якої міри зміненою шляхом рефакторингу є послідовність копії відносно оригіналу, ці вирази завжди будуть однаковими. При цьому, саме такі вирази становлять більшу частину кодової послідовності, у той же час фактично несучи меншість впливових ресурсів на результати порівняння. Таким чином, обробка цих відрізків коду вимагає використання надлишкових ресурсів, як результат збільшуючи час обробки вхідних даних алгоритмом.

Також, з метою покращення показників швидкості проведення перевірки подібності запропонованим алгоритмом, пропонується проводити виключення усіх повторюваних та заздалегідь однакових елементів послідовностей, що обумовлено особливістю коду як текстової послідовності – незалежно від мови програмування якою написано код, у ньому завжди присутні деякі вирази, які не підлягають зміні навіть під час модифікації методами рефакторингу, оскільки це призведе до втрати кодом працездатності. Таким чином, дані

послідовності варто піддати виключенню перед початком проведення порівняння послідовностей коду, тим самим скоротивши обсяг вхідних даних, без втрати точності порівняння.

Показниками успішності реалізації запропонованого рішення є:

1. відношення результату обрахунку редакційної відстані за удосконаленою версією відстані Дамероу-Левенштейна до результатів отриманих при порівнянні тих же зразків коду за відстанню Левенштейна, як показник покращення точності визначення редакційної відстані;

2. покращення показників швидкості обробки удосконаленим алгоритмом порівняння версій коду за відстанню Дамероу-Левенштейна у порівнянні зі швидкістю обробки порівняння за відстанню Левенштейна.

Завдяки застосуванню удосконаленому алгоритму вдалося точніше визначити відстань Дамероу-Левенштейна між тестовими стрічками коду та довести, що вони є майже однаковими. Точність обчислення при цьому зросла в середньому на 33% для розглянутих випадків.

Розроблений удосконалений алгоритм передбачає скорочення довжини послідовностей коду, що перевіряються ще перед початком роботи алгоритму порівняння, отже автоматично навантаження на алгоритм у порівнянні із оригінальним рішенням буде зменшуватись. При проведенні тестування очікуваний характер поведінки зміни часової ефективності удосконаленого алгоритму відносно оригінального матиме вигляд параболи удосконаленого алгоритму буде більш пологою одночасно, ніж парабола O(n^2) та параболо-видний графік O(m*n) оригінального алгоритму.

Висновки. На сьогодні для проведення подібних перевірок найчастіше застосовуються алгоритми прямого порівняння, відстань Хеммінга та відстань Левенштейна. Основними недоліками цих методів є обмежена кількість операцій, які можуть виявляти ці алгоритми та врахування регістру літер як різних символів. Як альтернативу даним методам було розглянуто використання алгоритму для визначення відстані Дамероу-Левенштейна. Даний алгоритм дозволяє виявляти застосування транспозиції при виконанні рефакторингу для файлу копії, тим самим показуючи значно кращі показники точності порівняння послідовностей. Однак, і даний метод має недоліки у контексті порівняння саме файлів вихідного коду, зокрема те ж врахування регістру літер як різних символів. Рішення для усунення даного недоліку було знайдено у зведенні усіх символів обох порівнюваних файлів вихідного коду до одного регістру перед початком перевірки.

Також, було виявлено необхідність пошуку рішення для оптимізації часу виконання порівняння, адже на реальних задачах із пошуку порушення автентичності файлів вихідного коду об'єми порівнюваних даних є досить великими. Часова ефективність оригінального алгоритму для визначення відстані Дамероу-Левенштейна за Big O була визначена як O(m*n), що свідчить про те, що час обробки вхідних даних алгоритмом повністю залежить від обсягу цих даних. У той же час було виявлено характерну особливість коду як текстової послідовності – він завжди матиме повторювані та невід'ємні елементи, що заздалегідь не впливають на результати порівняння, оскільки їх зміна призводить до втрати кодом працездатності. Таким чином, було запропоновано рішення додати перед початком обчислення відстані обробку файлів перевірки алгоритмом, що видалятиме із послідовностей усі завчасно однакові елементи. Для перевірки працездатності запропонованих рішень було виконано обрахунки для однакової пари послідовностей за трьома алгоритмами: оригінальним алгоритмом для відстані Левенштейна, оригінальним алгоритмом для відстані Дамероу-Левенштейна та алгоритму для відстані Дамероу-Левенштейна із додаванням запропонованих рішень.

УДК 004.056.523.052(045)

О.В. Салієва[1], І.О. Бондаренко[1], М.О. Берестенко[1]
salieva8257@gmail.com
[1]*Вінницький національний технічний університет, м. Вінниця*

УДОСКОНАЛЕННЯ АЛГОРИТМУ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧА НА ОСНОВІ ЕЛЕКТРОННОГО КЛЮЧА ТА ДИНАМІЧНОЇ БІОМЕТРІЇ

Невпинний розвиток інформаційних систем обумовлює гостру потребу забезпечення захищеності даних, що циркулюють в них. Вирішити дану задачу можна різними способами, зокрема шляхом обмеження доступу до конфіденційної, службової, таємної та інших видів інформації за допомогою механізмів автентифікації.

Дослідженню питань щодо удосконалення методів автентифікації користувача присвячено безліч наукових праць, зокрема [1-5].

Особливий інтерес представляє двофакторна автентифікація, що забезпечує ідентифікацію користувача за допомогою двох різних типів автентифікаційних даних. Механізмами для проведення такого типу автентифікації є: знання (інформація, яку знає тільки користувач), володіння (предмет, який є лише у користувача) та невід'ємність (біометричні дані користувача).

Біометричні засоби захисту інформації гарантують високу надійність, підвищений рівень безпеки, неможливість відмови від авторства та зручність для користувачів, враховуючи невід'ємність біометричних характеристик від конкретної особи. Важливе місце серед біометричних факторів займають динамічні, що ґрунтуються на поведінковій характеристиці людини: клавіатурний почерк, рукописний підпис, динаміка роботи комп'ютерної мишки і т. п. При вдалій комбінації біометрії з іншим типом автентифікаційних даних, наприклад із захищеним електронним ключем, система здатна захистити інформацію як від внутрішніх загроз, так і від зовнішніх вторгнень.

Метою роботи є удосконалення алгоритму двофакторної автентифікації користувача на основі захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів мишки.

Об'єктом дослідження є удосконалений алгоритм багатофакторної автентифікації користувача.

Предметом є процес вдосконалення алгоритму для підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів мишки.

У результаті дослідження удосконалено алгоритм двофакторної автентифікації на основі розробленого електронного ключа з використанням технології JSON Web Tokens та запропонованого алгоритму аналізу ентропії рухів мишки.

Першим етапом автентифікації користувача є застосування електронного ключа, використання якого можна описати таким чином:

Крок 1. Застосування даних користувача для запиту на формування токену.

Крок 2. Перевірка інформації користувача.

Крок 3. Формування електронного ключа та його шифрування.

Крок 4. Надання ключа користувачеві.

Крок 5 Зберігання користувачем токену та прикріплення його значення до кожного запиту.

Крок 6. Надання доступу користувачеві після перевірки ключа системою.

Крок 7. Можливість відновлення доступу на основі електронного ключа з використанням refresh-токену.

Запропонований алгоритм дозволяє не зберігати інформацію про всі видані токени, як при класичній схемі. У випадку реалізації окремого модуля сервісу аутентифікації за даним алгоритмом стає можливим створення єдиної точки входу в різні сервіси з однаковими обліковими даними.

Додаткове шифрування токену при передаванні користувачеві підвищує рівень захисту переданої послідовності та не викриває структуру даних, що розміщені у файлі. Для шифрування даних токену було використано симетричний ітеративний алгоритм AES, який дозволяє користувачам знайти компроміс між швидкістю та безпекою.

З метою визначення ентропії рухів комп'ютерної мишки було проаналізовано такі показники як:

траєкторія руху комп'ютерної мишки;

відхилення руху комп'ютерної мишки від зафіксованої траєкторії;

швидкість руху комп'ютерної мишки;

прискорення руху комп'ютерної мишки;

нетипові рухи комп'ютерної мишки;

кліки комп'ютерної мишки.

Розроблений алгоритм автентифікації користувача можна представити таким чином:

Крок 1. Запуск виконуваного файлу додатку.

Крок 2. Здійснення процесу автентифікації користувача.

Крок 2.1 Якщо користувач вже має обліковий запис у системі – перехід до кроку 6.

Крок 2.2 Якщо користувачеві необхідно отримати доступ до системи – перехід до кроку 3.

Крок 3. Реєстрація користувача.

Крок 3.1 Заповнення користувачем персональних даних.

Крок 3.2 Здійснення процесу аналізу рухів мишки.

Крок 3.3 Підтвердження процесу реєстрації.

Крок 4. Перевірка форм заповнених користувачем.

Крок 4.1 Якщо всі дані введені вірно та сформовано зразок рухів мишки, то запит на реєстрацію підтверджується.

Крок 4.2 Якщо форма реєстрації заповнена невірно, користувачеві виводиться на екран відповідне сповіщення, а форма реєстрації потребує повторного заповнення.

Крок 5. Підтвердження реєстрації користувача з боку адміністратора.

Крок 5.1 Адміністратор перевіряє нового користувача за його обліковими даними.

Крок 5.2 Якщо такому користувачеві необхідно надати доступ до системи – ставить відповідну відмітку для розмежування доступу.

Крок 5.3 Адміністратор формує електронний ключ, що надсилається користувачеві.

Крок 5.4 Збереження внесених змін.

Крок 6. Здійснення процесу автентифікації користувача.

Крок 6.1 Заповнення користувачем поля для введення логіну.

Крок 6.2 Завантаження електронного ключа.

Крок 6.3 Здійснення процесу аналізу рухів мишки.

Крок 6.4 Підтвердження процесу автентифікації.

Крок 7. Перевірка надання користувачеві доступу до системи.

Крок 7.1 Якщо логін та ключ користувача коректні, зразок ентропії рухів мишки відповідає зареєстрованому – користувач отримує доступ.

Крок 7.2 Якщо один із факторів не відповідає вимогам – користувачеві виводиться на екран сповіщення про невдалу спробу автентифікації.

У випадку, якщо користувач тричі здійснюватиме некоректний вхід, – доступ до облікового запису заблокується, електронний ключ буде недійсним. Для його поновлення потрібно звернутись до адміністратора та отримати новий електронний ключ на основі refresh-токену.

Крок 8. Після успішної автентифікації користувачеві надається доступ до системи.

Тобто, забезпечивши два фактори автентифікації, які залежать виключно від конкретного користувача (оскільки і ключ, і ентропія рухів мишки індивідуальні для кожного), маємо змогу підвищити достовірність автентифікації користувачів при здійсненні функції реєстрації та автентифікації у системі.

Висновки. З активним розвиток інформаційних технологій та систем невпинно зростає можливість зламу алгоритмів автентифікації, які ще донедавна вважалися надійними. Тому дане дослідження було спрямоване на вдосконалення алгоритму двофакторної автентифікації користувача на основі захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів мишки, що є індивідуальною характеристикою для кожної людини.

Для уникнення помилок автентифікації було розроблено електронний ключ з використанням ефективної, гнучкої та безпечної технології JSON Web Tokens. Застосування аналізу ентропії рухів мишки дозволило здійснити автентифікацію користувача на основі біометричних поведінкових характеристик, які є універсальними, унікальними та постійними.

Список літератури

1. В. В. Фесьоха, та Н. О. Фесьоха, «Модель нечіткої автентифікації користувачів інформаційних систем органів військового управління на основі поведінкової біометрії,» Захист інформації, т. 23, № 2, с. 116–123, 2021.

2. О. В. Горбенко, Ю. Л. Горбенко, А. Ю. Горбенко, та О. М. Сівоха, «Захист інформаційних систем за допомогою використання методів автентифікації,» Збірник наукових праць Центру воєнно-стратегічних досліджень НУОУ імені Івана Черняховського, с. 79-85, 2020.

3. О. Г. Корченко, А. М. Давиденко, та О. О. Висоцька, «Метод автентифікації користувачів інформаційних систем за їх рукописним почерком з багатокроковою корекцією первинних даних,» Захист інформації, т. 21, №1, с. 40-51, 2019.

4. P. Jayapriya, R.R. Manimegalai, and R. Kumar Lakshmana, "A Survey on Different Techniques for Biometric Template Protection," Journal of Internet Technology, vol. 21, no. 5, 2020.

5. A. Sarkar and Binod K. Singh, "A Review on Different Biometric Template Protection Methods," Recent Advances in Computer Science and Communications, vol.14, issue 5, pp. 1551–1572, 2021.

УДК 004.056, 004.75

А.А. Шиян[1], І.В. Абрамчук[1], В.В. Гуменюк[1]
anatoliy.a.shiyan@gmail.com
[1]*Вінницький національний технічний університет, м. Вінниця*

ОСОБЛИВОСТІ УПРАВЛІННЯ МОРАЛЬНО-ЕТИЧНИМ СТАНОМ НАСЕЛЕННЯ ПІД ЧАС ВІЙНИ В ІНФОРМАЦІЙНОМУ ПРОСТОРІ

Морально-етичний стан населення є важливим фактором. Саме ним часто визначається стійкість суспільства, яка дозволяє йому захищатися від інформаційно-психологічних операцій ворога. Поділ наративів на «свої» та «чужі», ідентифікація «своїх» є важливим елементом для забезпечення оптимальних умов функціонування суспільства. Це підвищує адаптивні ресурси людей, надаючи їм можливість вірити у наступну перемогу в майбутньому та жити і працювати у важких стресових умовах сьогодення.

На даному етапі розвитку демократичного суспільства такої європейської країни як Україна, життя людини є абсолютною цінністю. Але війна щодня приносить звістки про смерті людей, і тут важливим моментом є поділ загиблих на «своїх» та «ворогів». Смерть ворогів сприймається як даність, як природній розвиток подій. Дуже важливо утримувати саме таке сприйняття ситуації. Воно допомагає населенню прийняти ситуацію війни, допомагає навчатися жити у період війни.

І саме на руйнування такого морально-етичного стану нашого суспільства спрямовано ряд інформаційно-психологічних атак ворога. На перед початком війни це були яскраво виражені наративи «ми ж братні народи», «Україна молодший брат» (тобто повинен підкорятися волі «старшого»), «українці «завинили» перед «старшим» братом», «пора повернутися у сім'ю» тощо.

Російські спеціалісти із інформаційної війни, несподівано для себе, попали у пастку. Пастка ця полягала у їх власній картині світу, яка була сформована за часів СРСР та практично без змін продовжувала існувати в Росії. Тоді як у суспільстві Україні дуже швидко, буквально протягом лічених років, почала формуватися нова картина світу. Українці почали згадувати свою історію, аналізувати та переосмислювати історичні події. Відкрився доступ до нових фактів, до нової інформації, тієї, яка раніше була недоступна широким верствам народу. І картина світу українця почала потроху змінюватися. Люди стали менше звертати увагу на зовнішньополітичні події, все більше увага населення ставала прикута до внутрішнього життя в Україні. А російське суспільство все ще жило міжнародними подіями, переживаючи за «обличчя Росії» та отримання нею «достойного місця» у світі. Недаремно перемога СРСР над Німеччиною у Другій Світовій війні перетворилася у культ в Росії, - і стала практично непомітною в Україні.

Початок військового вторгнення, повномасштабної війни українське суспільство зустріло адекватно: як ворожу навалу, яка прийшла руйнувати наше життя. Суспільство мобілізувалося досить швидко. Декілька перших місяців кожен українець прожив у стані шоку. Після цього наступив етап адаптації нашого суспільства до війни. Почалася формуватися нова картина світу, яка відповідала життю в умовах війни, яка продовжується.

Кожна суспільство на етапі реформування прийнятних правил життя, внесення змін у картину світу, стає вразливим до сторонніх впливів. Тому сьогодні зростає необхідність у захисті населення від інформаційних атак, особливо морально-етичного плану. Такі атаки здійснюють російські спецслужби переважно через ЗМІ та Інтернет, залучаючи до цього потужні інформаційні канали впливу. Це і література, і кіно, і мистецтво, і брехня, і «аналітика» - і все це інформаційними каналами розповсюджується не тільки в Україну, але й по практично всім країнам світу. Це робиться для того, щоб частина російських наративів прийшла до нас із-за кордону, особливо із розвинених країн, передовсім з Європи. Тепер напрям атак на України – це показати, що Україна залишається одна у світі, що суспільство розвинених країн перестає підтримувати Україну.

Найбільш ефективна протидія цьому сьогодні полягає у тому, щоб залучати все більшу частину суспільства до активної праці на ниві протидії ворогу. Народ, який своєю працею працює на перемогу, отримує свого роду імунітет проти інформаційно-психологічних атак на морально-етичний стан.

Деякі напрямки залучення широкого загалу населення, наведено в [1]. Використання запропонованих там технологій дозволить отримати активний підтримуючий зворотний зв'язок від суспільства розвинених країн, що стане важливим фактором підтримки Україні.

Список літератури

1. Shyian Anatolii, International Financing Directions for the Transformation of the Country's Economy in the Modern War (Ukraine as the Example) (July 26, 2022). Available at SSRN: https://ssrn.com/abstract=4173426 or https://dx.doi.org/10.2139/ssrn.4173426. 10 p.

ЗМІСТ

СЕКЦІЯ 1. ІНФОРМАЦІЙНА БЕЗПЕКА ДЕРЖАВИ, СУСПІЛЬСТВА ТА ОСОБИСТОСТІ

СЕКЦІЯ 2. ПРОГРАМУВАННЯ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

СЕКЦІЯ 3. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ, МЕДИЦИНІ ТА ОСВІТІ

НАУКОВЕ ВИДАННЯ

ТЕЗИ ДОПОВІДЕЙ

VI Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології"

20-21 квітня 2023 року

Матеріали публікуються в авторській редакції.
За достовірність викладених фактів, цитат та інших відомостей
відповідальність несуть автори.

Відповідальний за випуск: *О.А. Смірнов*

Комп`ютерна верстка: *Р.М. Минайленко*

Електронне видання

Центральноукраїнський національний технічний університет
пр-кт Університетський, 8, м. Кропивницький, 25006.
тел. (0522) 559-245, www.kntu.kr.ua