КПІ ім. Ігоря Сікорського

МАТЕРІАЛИ П`ЯТОЇ ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ МОЛОДИХ ВЧЕНИХ ТА СТУДЕНТІВ «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ УПРАВЛІННЯ»

ІСТУ - 2020

ОСІНЬ

1978

АСОІУ

26 та 27 листопада
Україна,Київ

Оргкомітет конференції

Голова організаційного комітету: О.А. Павлов – в.о. зав. кафедри АСОІУ, д.т.н., професор.

Члени організаційного комітету:

П.І.П.	Посада
О.А. Павлов	В.о. зав. кафедри АСОІУ, професор
І.В. Стеценко	Професор кафедри АСОІУ
І.П. Муха	Доцент кафедри АСОІУ
О.В. Гавриленко	Доцент кафедри АСОІУ
К.І. Ліщук	Доцент кафедри АСОІУ
О.Г. Жданова	Доцент кафедри АСОІУ
Т.О. Тєлишева	Доцент кафедри АСОІУ
Ю.О. Олійник	Старший викладач кафедри АСОІУ

Члени програмного комітету:

П.І.П.	Посада
В.І. Литвиненко	Професор ХНТУ
Г.В. Рудакова	Професор ХНТУ
І.В. Баклан	Доцент кафедри АСОІУ
М.О. Сперкач	Доцент кафедри АСОІУ
Е.В. Жаріков	Доцент кафедри АСОІУ
О.С. Жураковська	Доцент кафедри АСОІУ
В.Д. Попенко	Доцент кафедри АСОІУ
Л.В. Рибачук	Доцент кафедри АСОІУ
О.А. Халус	Старший викладач

V Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2020». Секція кафедри автоматизованих систем обробки інформації і управління. Матеріали конференції. – Київ. – 2020. 26–27 листопада 2020р. – 160 с.

У збірник включені тези доповідей, які були представлені на конференції “Інформаційні системи та технології управління – ІСТУ-2020” в секції кафедри автоматизованих систем обробки інформації і управління. В доповідях розглянуті наукові та методичні питання щодо сучасних аспектів інформатики та обчислювальної техніки.

Редакційна колегія:

Гавриленко О.В., к.ф.-м.н., кафедра АСОІУ НТУУ «КПІ ім. Ігоря Сікорського»,
Муравйова І. М., інженер ІІ категорії кафедри АСОІУ

Дизайн титульної сторінки: Майєр З.О.

ЗМІСТ

УДК 004.942

АБРАШИНА Н.О.
БАКЛАН І.В.

СЕМАНТИЧНЕ МОДЕЛЮВАННЯ ВІРТУАЛЬНОГО СТИМУЛЯТОРА ТИРУ

У даній статті розглянуто метод семантичного моделювання за допомогою практичної побудови та дослідження модельованої системи на прикладі віртуального симулятора тиру. Розглянуто алгоритми і методи, властиві для подібних систем, та їхнє відображення у семантичній моделі.

СЕМАНТИЧНЕ МОДЕЛЮВАННЯ, ВІРТУАЛЬНИЙ ТИР, МОДЕЛЬ, СТРІЛЬБА

This article considers the method of semantic modeling through the practical construction and study of the simulated system on the example of a virtual shooting simulator. Algorithms and methods inherent in such systems and their reflection in the semantic model are considered.

SEMANTIC SIMULATION, VIRTUAL SHOOTING, MODEL, SHOOTING

1. Вступ

Віртуальні симулятори тиру можуть мати значну практичну користь, тому корисним буде дослідити можливу семантичну модель таких систем. Серед можливих методів такого дослідження можна виділити аналіз вже існуючих систем та побудову власної системи для аналізу. Реверс-інженірінг багатьох програмних засобів є небажаним з точки зору їх розробників, до того-ж, створення власної системи допоможе нам дослідити деякі нові для сфери технології, тому зупинимося на цьому варіанті.

2. Огляд існуючих рішень

Хоча ми прийняли рішення не проводити реверс-інженірінг існуючих систем, усе ж буде доцільним розглянути відкриту інформацію про них, аби отримати загальне уявлення про методи, що використовуються при побудові подібних систем. За такою інформацією звернемося до деяких патентів, що зареєстровані у Європі та США.

Можна виділити дві основні групи подібних систем: системи доповненої реальності, що використовують фізичні мішені для віртуального ведення вогню (патент [1]) та повністю віртуальні системи (патенти [2-4]). Друга група більш цікава для нас, однак у загальних рисах розглянемо також першу.

Так [1] присвячений системі, що реєструє дані про постріл (положення зброї та цілі на момент віртуального пострілу, що визначають за GPS) та на основі даних про карту, стрілка та віртуальну зброю робить припущення про те, чи буде постріл успішним. Автори патенту стверджують, що система збирає дані про напрям зброї, рівень тремору рук стрілка, рельєф, рівень опадів, враховує особливості вибраної зброї та попередній досвід стрілка.

Наступні патенти [2-4] відзначаються іншим підходом до розв'язання розглянутої проблеми, зокрема, віртуальному веденню вогню за віртуальними цілями.

Патенти [2-3] присвячені найпростішим версіям систем віртуальної стрільби: винахід, що описаний патентом [2], заснований на принципі, згідно якого зброя для віртуального пострілу вловлює інфрачервону мітку, що випромінює ціль, а винахід, що описаний патентом [3] – на принципі, згідно якого сама зброя для віртуальної стрільби випромінює інфрачервоний промінь, який вловлює детектор, розміщений за екраном. Строк дії обох патентів вже сплив, і вони мають значення скоріше у історичному розрізі, як приклади одних з перших систем подібного роду.

І нарешті[4], присвячений системі віртуального полювання, що імітує постріли

у потоці відео, а також виконує відео зйомку при натисканні на спусковий гачок реальної або імітованої зброї. Як із пояснень авторів винаходу, так і з його опису помітно, що розглянута система не призначена для навчання веденню вогню, а її основною метою є створення відповідного досвіду без шкоди для довкілля.

Як можемо побачити, існує багато способів побудови таких систем, що, ймовірніше за все, працюють із різними даними і, відповідно, мають різну модель даних. У даній роботі ми розглянемо систему, що використовуватиме іще один принцип: розпізнавання положення голови та положення моделі зброї на зображенні з камери.

3. Задачі та процеси системи

Розглянута система має забезпечувати підтримку діяльності двох ролей користувачів:

1. Привілейований користувач, у перелік задач якого входить організація навчального процесу (далі інструктор);
2. Рядовий користувач, у перелік задач якого входить проходження навчання (далі курсант).

Таким чином, перед розглянутою системою постають такі основні задачі:

1. Ведення даних про курсантів;
2. Проведення тренувань;
3. Оцінювання результатів тренувань.

Для підтримки виконання даних задач потрібно забезпечити такі процеси:

1 Для ведення даних про курсанта:
- 1.1 перегляд даних;
- 1.2 додавання даних;
- 1.3 видалення даних~~;~~.

2 Для проведення тренувань:
- 2.1 авторизація курсанта;
- 2.2 вибір місії;
- 2.3 проходження завдання;
- 2.4 отримання звіту~~;~~.

3 Для оцінювання результатів тренувань:
- 3.1 перегляд історії звітів.

Взаємодію між процесами продемонструємо на прикладі діаграми варіантів використання:

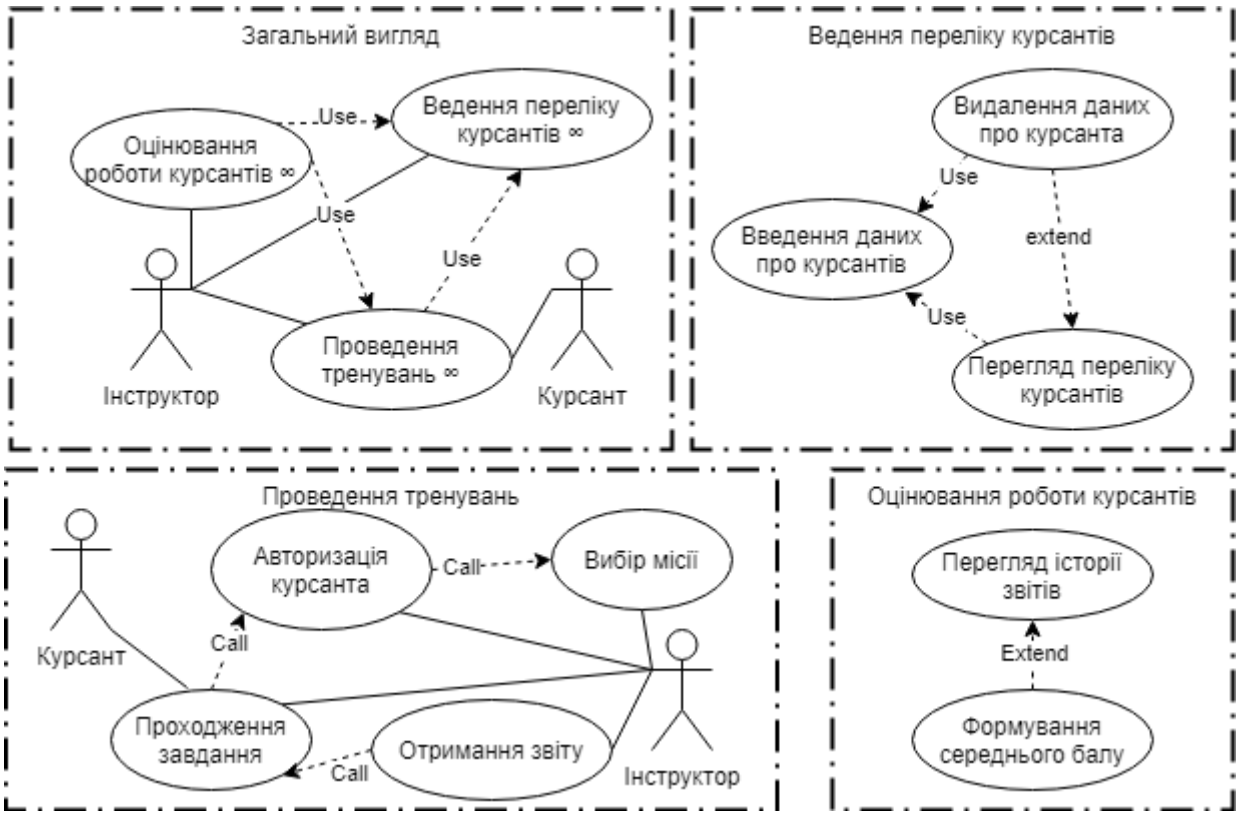


Рис. 1 Діаграма варіантів використання для системи

4. Реалізація проходження завдання

Зупинимося більш детально на процесі проходження завдання.

Процес проходження завдання полягає у тому, що курсант оглядає карту, прицілюється і виконує постріли, а система обраховує точки влучання куль і формує звіт про проходження завдання. Відповідно, потрібно реалізувати такі функції, як:

1. керування напрямом погляду;
2. керування напрямом ствола;
3. моделювання польоту кулі.

В якості способу керування напрямом погляду було обрано рух головою і його розпізнавання з використанням веб-камери. Алгоритм розв'язання даної задачі виходить за рамки роботи; для її реалізації використано open source бібліотеку Headpose Detection [5].

Для визначення положення ствола використовуємо куб із гранями різного кольору, закріплений на кінці моделі ствола. Визначення напряму ствола відбувається за допомогою визначення положення кубу на зображенні з камери.

Для визначення положення кубу на камері, скористаємося алгоритмом, що запропонований Т.М. Власовою та В.Г Калмиковим[6]: виділяємо крайні пікселі зображення кожної з граней, що потрапила у кадр, та за запропонованим у роботі методом розбиваємо сукупність пікселів на відрізки, таким чином, знаходимо координати кутів чотирикутників. Після цього, знаючи, що грані є квадратами, визначаємо кути між площиною екрану та гранями кубу, що потрапили до кадру, а знаючи, як спрямована кожна з граней, визначаємо напрям ствола.

Для моделювання польоту кулі, скористаємося методом Сіаччі, взявши за основу його формулювання у підручнику “Внешняя баллистика” Я.М. Шапіро [7]. Даний метод працює для будь-яких початкових швидкостей, і дає найточніший результат при прямому наведенні, що відповідає нашій задачі.

Приймемо:

- θ — кут траєкторії снаряду до горизонту;
- u – швидкість снаряду у горизонтальній площині;
- c’ = cβ – балістичний коефіцієнт;
- $U = \frac{u}{cos\theta_0}$ - псевдо-швидкість, що використовується у законі Сіаччі;
- J(U), T(U), D(U), A(U) – функції, значення яких задані таблично.

Загальна формула закону Сіаччі має вигляд:

$$tg\theta = tg\theta_0 - \frac{1}{2c'cos^2\theta_0}[J(U) - J(u_0)]$$

Виходячи з неї, елементи траєкторії мають вигляд:

$$t = \frac{1}{c'cos\theta_0}[T(U) - T(u_0)]$$

$$x = \frac{1}{c}'[D(U) - D(u_0)]$$

$$y = xtg\theta_0 - \frac{x}{2c'cos^2\theta_0}\left[\frac{A(U) - A(u_0)}{D(U) - D(u_0)} - J(u_0)\right]$$

5. Модель даних

Виходячи з попереднього, можемо встановити, що система має працювати із такими елементами даних:

1 Авторизаційні дані інструктора:
  1.1 логін;
  1.2 пароль.
2 Дані курсанта:
  2.1 ім’я;
  2.2 вік;
  2.3 стать.
3 Дані про карту:
  3.1 назва карти;
  3.2 посилання на карту.
4 Дані для симуляції пострілу:
  4.1 назва боєприпасу;
  4.2 початкова швидкість пострілу;
  4.3 коефіцієнт опору повітря для закону Сіаччі.
5 Дані про результати симуляції:
  5.1 тип запису:
    5.1.1 MissionStarted;

5.1.2 AmmoFired;
5.1.3 TargetShot;
5.1.4 ObjectShot;
5.1.5 ForbiddenObjectShot;
5.1.6 OutOfAmmo;
5.1.7 OutOfTime;
5.1.8 AllTargetsShot;
5.1.9 MissionFinish:

5.2 час;
5.3 номер враженого об'єкту (для записів типів TargetShot, ObjectShot, ForbiddenObjectShot).

Також, для роботи системи необхідні таблиці функцій Сіаччі.

Структуру даних системи зображено на ER діаграмі:

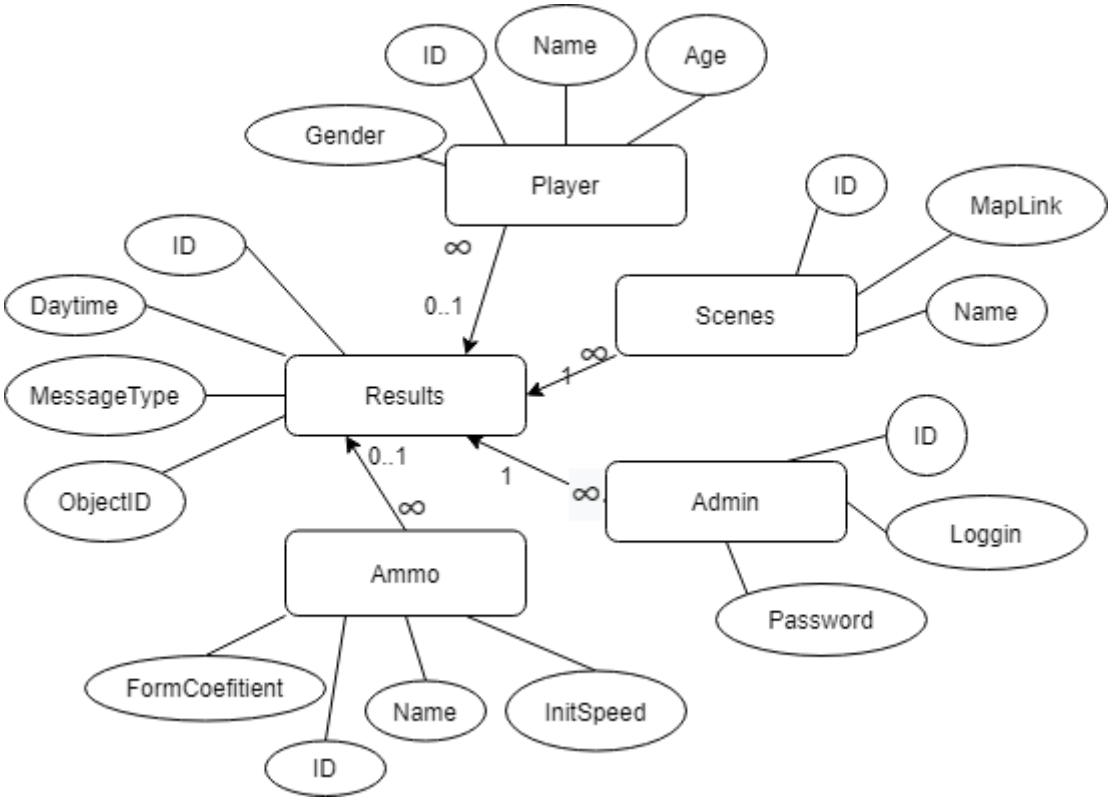


Рис.2 Entity-Relationship діаграма для системи

6. Опис програмного та технічного забезпечення

При розробці програмного забезпечення, використаємо такі програмні засоби:

- мова програмування C# [8];
- мова програмування Python 3 [9];
- Unity framework [10];
- середовище розробки Microsoft Visual Studio 2017 [11];
- середовище розробки Anaconda [12];
- open-sours бібліотека Headpose-Detection;
- бібліотека CSVHelper [13];

Програмне забезпечення складатиметься з двох компонентів: UnityFramework застосунку, що виконує основні функції програми (далі основна програма), та консольного застосунку на основі бібліотеки Headpose-Detection, що виконує розпізнавання обличчя (далі розпізнавач облич). Зв'язок між компонентами забезпечується за рахунок сокетів (sender на боці розпізнавача облич та listener на боці основної програми). Загальний вигляд структури програмного забезпечення наведено у діаграмі компонентів:

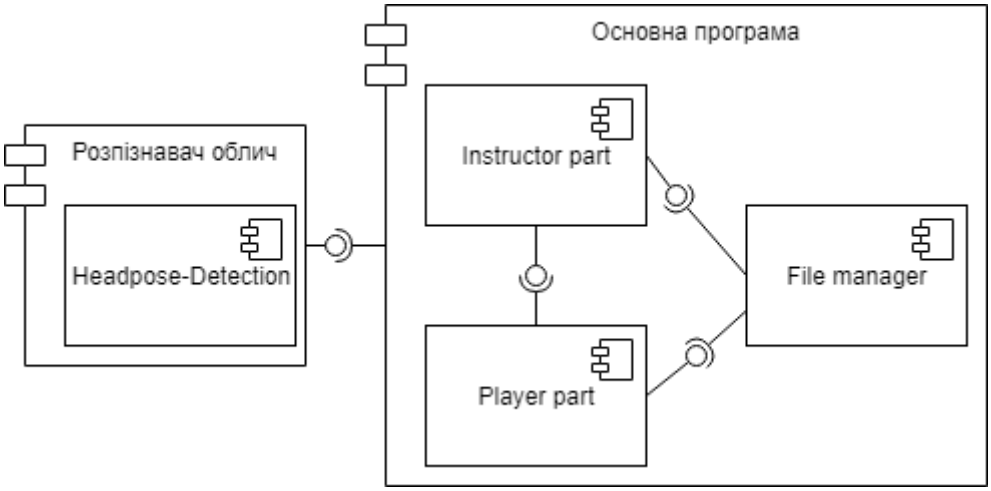


Рис. 3 Діаграма компонентів системи

Для розгортання програмної системи, необхідні такі технічні елементи:

- комп'ютер з встановленою операційною системою Windows 10 та інтерпритатором мови Python;
- два монітори (будь-який зручний для роботи для інструктора та широкоформатний монітор або проектор і екран для курсанта);
- клавіатура та миша для інструктора;
- модель зброї з описаними вище модифікаціями, до спускового гачка якої під'єднано виносну кнопку, для курсанта;
- веб-камера із великим полем зору та великою роздільною здатністю.

Також рекомендується використовувати ширму нейтрального кольору для уникнення випадкового спрацьовування розпізнавача облич, однак вона не є частиною апаратного забезпечення. Схему розгортання наведено у діаграмі розгортання:

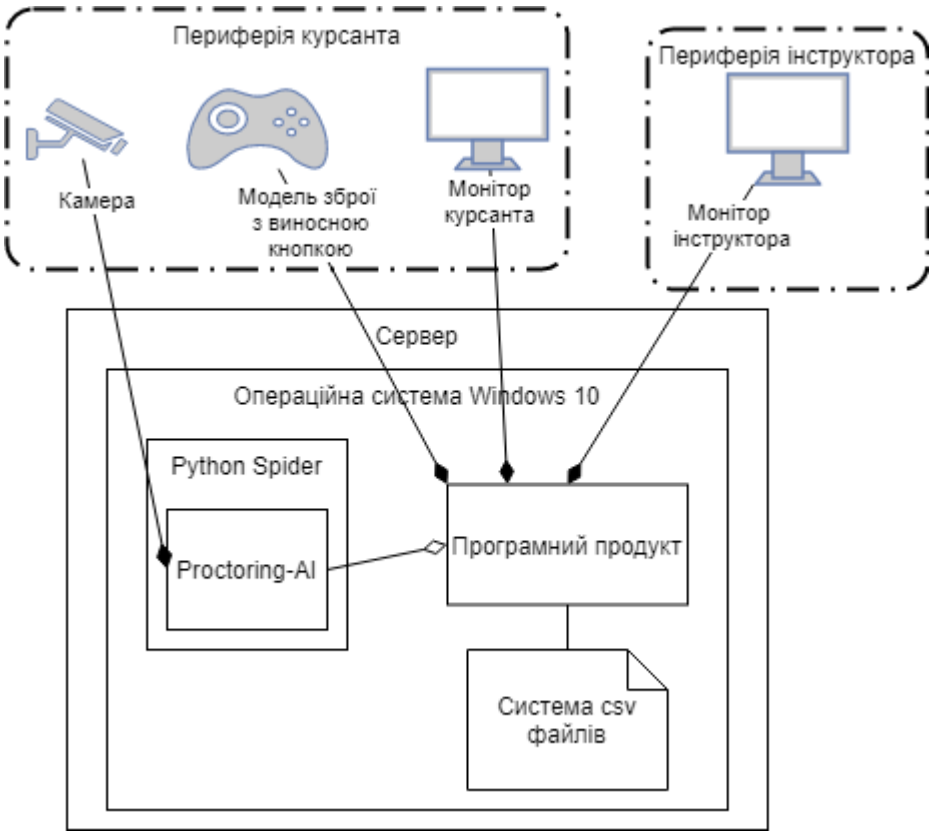


Рис. 4 Діаграма розгортання системи

Висновок

У даній роботі досліджено проблему побудови семантичної моделі віртуального симулятора тиру. Досліджено інформацію про існуючі системи такого типу, досліджено задачі таких систем та процеси, що необхідно забезпечити для виконання задач. Побудовано часткову математичну модель та модель даних у системі. Розглянуто структуру системи та логіку її розгортання. Система, побудована за наведеною у статті моделлю, має задовольняти усім вимогам, що ставляться до систем подібного роду, а також використовувати значно менш дорогу периферію, аніж більшість наявних аналогів.

Перелік використаної літератури

1. Joel Glauber, Monroe, NY(US) 2016, *SHOOTING SIMULATOR WITH GPS AND AUGMENTED REALITY*, US 20180100724A1
2. Gunpei Yokoi, Nintendo Co., Ltd., Kyoto, Japan 1975, *CLAY SHOOTING SIMULATION SYSTEM* 3,904,204
3. Allard, Jean-Claude, 13 rue Ravon, F-92340 Bourg la Reine (FR) Inventeur: Briard, Ren6, 4 rue du Chemin Creux Bures, F-78630Orgeval (FR) Inventeur: Saunier, Christian, 3 rue Derondel, F-95120 Ermont (FR) 1985, *DEMANDE DE BREVET EUROPEEN* 0146466
4. Ronnie Valdez, Denver, CO (US) 2017, *VIRTUAL HUNTING DEVICES AND USES THEREOF* US 20170 100675A1
5. Qhan Lin, 23 Jan 2019, *Headpose-Detection*, GitHub, переглянуто 09.11.2020, <https://github.com/qhan1028/Headpose-Detection>
6. Т. М. Власова, В. Г. Калмыков, 2005, ВЛАСОВА Т. М. АЛГОРИТМ И ПРОГРАММА РАСПОЗНАВАНИЯ КОНТУРОВ ИЗОБРАЖЕНИЙ КАК ПОСЛЕДОВАТЕЛЬНОСТИ ОТРЕЗКОВ ЦИФРОВЫХ ПРЯМЫХ, *Математичні машини і системи,* №4, С. 84–95.
7. Шапиро Я.М. 1946, *Внешняя баллистика*, ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО ОБОРОННОЙ ПРОМЫШЛЕННОСТИ, Москва
8. *C# documentation*, Microsoft, переглянуто 09.11.2020, <https://docs.microsoft.com/en-us/dotnet/csharp/>
9. *Python 3.9.0 documentation*, Python, переглянуто 09.11.2020, <https://docs.python.org/3/>
10. Unity official site, Unity, переглянуто 09.11.2020, <https://unity.com/>
11. *Visual Studio*, Microsoft, переглянуто 09.11.2020, <https://visualstudio.microsoft.com/ru/>
12. *Anaconda Documentation*, Anaconda, переглянуто 09.11.2020, <https://docs.anaconda.com/>
13. *CsvHelper*, GitHub, переглянуто 09.11.2020, https://joshclose.github.io/CsvHelper/

УДК004.8

СМІЛЯНЕЦЬ Ф.А.
МАЖАРА О.О.

КЛАСИФІКАЦІЯ НОВИН ЗА ДОСТОВІРНІСТЮ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

В роботі досліджено підходи до класифікації новин за достовірністю на основі двох методів машинного навчання: пасивно-агресивного класифікатору та нейронної мережі. Представлено сучасні підходи до формування навчальної вибірки та реалізації класифікаторів. Розроблено скрапер та на його основі сформовано навчальну вибірку для локальних новинних виданьдвомамовами. Реалізовано класифікатори за двома підходами. На основі отриманих результатів запропоновано напрямки подальших досліджень

КЛЮЧОВІСЛОВА

Дезінформативні новини, маніпулятивні новини, класифікація новин, обробка природньої мови, машинне навчання, нейронні мережі

The subject of the article is investigating approaches to classification of news by perceived truthness using two machine learning methods: a passive-aggressive classifier and a neural network. Modern approaches to formation of training data and classifier implementation were presented. A web scraper was implemented and used to create a training dataset of local news in two languages. Two classifiers were implemented. Further work directions were outlined based on current results.

KEYWORDS

Fake news, media manipulation, news classification, natural language processing, machine learning, neural networks

1. Вступ

Останнім часом все більше досліджень спрямовано на визначення впливу інформаційного простору на сучасне суспільство. Дезінформативна та недостовірна інформація може мати негативні наслідки на формування суспільних настроїв. Розповсюдженимиє публікації спрямовані на формування у читача конкретної позитивної чи негативної думки щодо певної події, історії, людини, компанії, тощо. Це робиться за допомогою поширеннявідверто хибних новин, надання емоційного забарвленнятексту, навмисного використання напівправди або виключно фактів які підтверджують точку зору, оминаючи ті, що спростовують.

Попри те, що абсолютно точне викриття дезінформації за допомогою комп'ютерів є AI-повною задачею, ця тема є поширеною в дослідженнях в області алгоритмів машинного навчання. Актуальною є задача створення засобів визначення недостовірних новин для попередження читача в реальному часі.

2. Підходи до класифікації новин

Класифікація новин на предмет правдоподібності є актуальною задачею, вирішенню якої присвячено ряд сучасних досліджень[1] [2] [3]. Прикладом комплексного рішення для попередження користувачів про неякісні новинні матеріали є trustedtimes.org. Однак здебільшого це стосується іноземних видань [2]. В той же час, в Україні дані дослідження знайшли обмежене застосування. На сьогодні, найбільш відомим представником на локальному ринку єєfgz.texty.org та Однак, даний сайт працюють з обмеженою кількістю видань. Також не поширюється інформація про методи, якими користуються для класифікації. Актуальною є задача дослідження підходів до класифікації новин за ознакою правдоподібності для двомовного українського середовища.

Наразі найпоширенішими засобами класифікації текстової інформації є методи машинного навчання. Умовно такі методи можна розділити за двома напрямками: класичні та на основі нейронних мереж. До перших належить, наприклад, пасивно-агресивний класифікатор. Серед нейронних мереж для задач такого типу найчастіше застосовують згорткові.

Аналіз існуючих методів пошуку маніпулятивних та дезінформативних новин, дозволив виокремлено наступні підходи до представлення даних та реалізації класифікатору (таблиця 1).

Істотна більшість існуючих робіт на дану тему фокусуються на новини представлені єдиною мовою, здебільшого англійською. Особливістю українського суспільного простору є двомовність засобів масової інформації. Тому в даному дослідженні, окремою завданням було забезпечити можливість класифікації новин представлених українською та російською мовами водночас.

Метою даної роботи було дослідити підходи до класифікації новин. Для цього була виокремлена задача порівняння двох алгоритмів машинного навчання у контексті двомовної вибіркидля визначення більш ефективного підходу з точки зору ресурсоємності та точності класифікації.

Для досягнення поставленої мети було виокремлено наступні завдання:

1. Обрати алгоритми для аналізу;

2. Визначити структуру даних, необхідних для навчальної вибірки та дослідити підходи до її формування;
3. На основі обраного підходу згенерувати навчальну вибірку з найпоширеніших сайтів новин;
4. Розробити класифікатори за двома обраними підходами та провести порівняльний аналіз їх ефективності;
5. Обрати підхід для подальшого застосування в технологіях розпізнавання неправдоподібних статей для використання на веб сторінках в режимі реального часу.

Для подальшого аналізу було обрано пасивно-агресивний класифікатор та нейронна мережа двосторонньої довгої короткочасної пам'яті (BidirectionalLongShort-TermMemory). Даний вибір зумовлено їх застосуванням в попередніх дослідженнях для класифікації англомовних видань, а такожшироким поширенням для вирішення інших класів задач в обробці текстової інформації.

Таблиця 1. Підходи до класифікації

Підхід до представлення даних	Класифікатор	Посилання
TF-IDF-метрика та її векторизація	Пасивно-агресивний класифікатор, наївний баєсів класифікатор, тощо.	Fake News Detection using Passive Aggressive and TF-IDF Vectorizer, 2020 [3]
Пріорітизуючатокенізація (топ-Nслів тренувального корпусу за частотою вживання)	Глибинні нейронні мережі, у тому числі згорткові та рекурентні	FakeNewsIdentificationonTwitterwithHybrid CNN and RNN Models, 2018 [4]
Глибокий лінгвістичний аналіз та підрахунок слів (LIWC)для оцінки емоційної забарвленості наявних у матеріалі слів та їх кількості.	Глибинні нейронні мережі, у тому числі згорткові та рекурентні	Linguisticinquiryandwordcount (LIWC), 1999 [5]. Truthofvaryingshades: Analyzinglanguageinfakenewsandpoliticalfact-checking. 2017 [6]
МодельDeepWalkдля побудови ембедінгу тексту у векторний простір	Класичні методи машинного навчання.	Deepwalk: Onlinelearningofsocialrepresentations, 2014 [7]

3. Постановка задачі

Дане дослідження присвячено аналізу підходів до класифікації новин за достовірністю. Під достовірністю в даному контексті розуміється приналежність новини до ресурсу, який вважається правдивим. Таким чином задача зводиться до задачі класифікації новин за ресурсами.

Для вирішення поставленоїзадачі необхідно вирішити наступні завдання:

1. Створити вибірку навчальних даних;
2. Розробити класифікатор;
3. Провести тестування.

Вхідними даними задачі є заголовок та текст статті, вихідними - ймовірність приналежності до одного з класів: достовірна/недостовірна.

Для формування навчальної вибірки було вирішено скористатися рейтингами та керівництвами такого авторитетного джерела аналізу розповсюдження дезінформативних та маніпулятивних новин як Інститут Масової Інформації (texty.org.ua).

Альтернативним підходом до створення навчальної вибірки розглядалась ручна розмітка на краудсорс платформі. Такий

підхід до розширення/модифікації набору даних може служити напрямком подальшого дослідження.

Навчальна вибірка має таку структуру:

- заголовок;
- текст статті;
- клас (достовірна/недостовірна).

На основі огляду алгоритмів, які знайшли широке застосування в аналізі англомовних видань для подальшого порівняння було виокремлено пасивно-агресивний класифікатор та нейронна мережа довгої короткострокової пам’яті. Обидва алгоритми дозволяють повернути ймовірність приналежності новини до вихідного класу.

Згідно з попередніми дослідженнями для навчання обраних алгоритмів доцільно виконувати попередню обробку набору даних. Для підготовки даних пасивно-агресивного класифікатору зазвичай використовується TFIDF-векторизація. Для використанняу рекурентній нейронній мережі доцільно застосовувати токенізацію тексту.

4. Формування навчальної вибірки

Для створення тренувальної вибірки було розроблено ресурс-агностичний веб-скрапер (

WebScraper). Він використовує інформацію розмітки гіпертексту для виокремлення значимих компонент веб-сторінки. За допомогою скраперу було проведено екстракцію 36781 новин з таких ресурсів:

- 40ka.info;
- bbc.com (тільки статті українською та російською);
- hyser.com.ua;
- liga.net;
- meduza.io (латвійське російськомовне ЗМІ, використане для балансування мов у наборі даних);
- novoross.info;
- ua3.info;
- radiosvoboda.org.

Відповідно до ступеню достовірності ресурсу згідно з рейтингом Інститут Масової Інформації новинам було додано ознаку приналежності до класу за такою схемою: 1 – для якісних видань, 0 – для видань з великою кількістю недостовірних новин та маніпуляцій.

В результаті навчальна вибірка містила по 26405 достовірних та 17068 недостовірних новин. В подальшому з даної множини випадковим чином було виокремлено тренувальну та тестові вибірки. До тестової вибірки належало 20% зібраних новин.

В якостіпопередньої обробки даних для пасивно-агресивного класифікатору було використано TF-IDF-векторизатор, який перетворює текст на матрицю чисельних TF-IDF-характеристик тексту.

Для підготовки даних для LSTMбуло використано базовий токенізатор (Tokenizer). Текстпредставлений набором слів без знаків розмітки відфільтрується за шумовими слова.В процесі токенізації він перетворюється на вектор значень у межах $[0, 2000)$. Слова, які не належать словнику токенізатора отримують індекс 0.

5. Реалізація класифікаторів

Для реалізації обранихкласифікаторів було проведено огляд сучасних засобів розробки алгоритмів машинного навчання.

Для реалізації пасивно-агресивного класифікатору було обрано бібліотекуsklearn.

Scikit-Learn (sklearn) -цеPython-бібліотека, яка містить реалізації великої кількості методів для підготовки та обробкиданиї, а також методів машинного навчання.

Нейронну мережу довгої-короткострокової пам’яті було реалізовано з використанням бібліотекиKerasна базі фреймворкуTensorflow.

Нейронна мережа є послідовністю шарів таких типів:

- Embedding-шар (перетворює словарний вектор індексів на "щільні" (dense) вектори);
- Шар двусторонньої довгої короткочасної пам'яті (BLSTM) на 64 елементи;
- Стандартний щільний шар на 32 елементи з випрямлячем (ReLU) у якості функції активації;

- Вихідний щільний шар у 1 елемент з сигмоїдною функцією у якості функції активації.

Процесс навчання виконувався у операційній системі Windows 10 з версією Python 3.7 на обчислювальній машині, обладнаній Ryzen 5 3600, Nvidia RTX 2070 Super, 16 GB 3200 MHz RAM.

Тренуванняпасивно-агресивного класифікатора з використанням sklearn виявилось неможливим на даній конфігурації на повному об'ємі даних внаслідок обмежень реалізації обраної бібліотеки. Для тренування на повній навчальній вибірці, що містить десятки тисяч статей системою було затребувано виділення 132 Гб оперативної пам'яті. Було вирішено спробувати натренувати модель на підмножині, яка включає по 5036 достовірних та 5036 недостовірних елементів. Тренування та тестування на обмеженому наборі даних тривали менше 30 секунд кожне. В результаті вдалося досягти точності класифікації 0.979.

Тренування нейронної мережі довгої-короткострокової, реалізованої на Keras та Tensorflow, тривало6 годин на навчальній вибірці того ж розміру.Точність класифікації становить0.966.

Незначна розбіжність в точності, дозволяє зробити висновок про перспективність застосування обох підходів.

В той же час, варто зауважити, що додатковим обмеженням деяких з бібліотек машинного навчання є складність їх інтеграції в різні екосистеми. Так Keras не надає реалізації пасивно-агресивного класифікатору, однак є більш зручним в використанні в веб-екосистемах за рахунок базової бібліотеки Tensorflow.js. Такий підхід дозволяє легко створювати вбудовані браузерні розширення.

Подальші дослідження в рамках даної теми спрямовані на інші способи представлення даних в навчальній вибірці, а також використання алгоритмів інших типів з метою створення браузерного розширення для класифікації новин в реальному часі.

Висновок

В роботі розглянуто підходи до вирішення задачі класифікації новин на предмет достовірності на основі методів машинного навчання. Представлено огляд підходів до формування навчальної вибірки та алгоритмів, які показали свою ефективність для англомовних видань. На його основі для подальших досліджень обрано пасивно-агресивний класифікатор та нейронні мережі з довгою короткостроковою пам'яттю. Розроблено скрапер, який дозволяє сформувати навчальну вибірку на базі двомовного контенту українських сайтів новин. Реалізовано класифікатори за двома обраними підходами. Вища точність класифікації досягнута з використанням пасивно-агресивного класифікатору. Однак для його адаптація для використання в якості основи для подальшого застосування в технологіях розпізнавання неправдоподібних статей для використання на веб сторінках в режимі реального часу ускладняється відсутністю реалізації в базових бібліотеках веб-розробки.Перспективною вважається реалізація алгоритму з подальшим використанням в Tensorflow.js.

Список літератури

1. J. C. S. Reis, A. Correia, F. Murai, A. Velosoand F. Benevenuto, "SupervisedLearningforFakeNewsDetection," in IEEE IntelligentSystems, vol. 34, no. 2, pp. 76-81, March-April 2019, doi: 10.1109/MIS.2019.2899143.
2. PedroHenriqueArrudaFaustini, ThiagoFerreiraCovões, Fakenewsdetectioninmultipleplatformsandlanguages, ExpertSystemswithApplications, Volume 158, 2020, 113503, ISSN 0957-4174
3. Jayashree M Kudari, Varsha V, Monica BG, Archana R, "Fake News Detection using Passive Aggressive and TF-IDF Vectorizer" International Research Journal of Engineering and Technology (IRJET)Volume 07, Issue, 09, Sep 2020e-ISSN: 2395-0056
4. Ajao, Oluwaseun&Bhowmik, Deepayan&Zargari, Shahrzad. (2018). FakeNewsIdentificationonTwitterwithHybrid CNN and RNN Models.

5. Pennebaker, James&Francis, Martha&Booth, Roger. (1999). Linguisticinquiryandwordcount (LIWC).
6. H. Rashkin, E. Choi, J. Jang, S. Volkova, and Y. Choi. Truthofvaryingshades: Analyzinglanguageinfakenewsandpoliticalfact-checking. In EMNLP, 2017.
7. B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Onlinelearningofsocialrepresentations. In KDD, 2014.
8. FakeNewsIdentificationonTwitterwithHybrid CNN and RNN Models (O. Ajao, D. Bhowmik, S. Zargari, C3Ri ResearchInstitute, 2018).

УДК 004.91

КУЩ А.В.
ГОЛОВЧЕНКО М.М.

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДОСЛІДЖЕННЯ ТА АНАЛІЗУ ТЕКСТІВ НА ПРЕДМЕТ НАЯВНОСТІ СПОЙЛЕРІВ У ОГЛЯДАХ МЕДІАКОНТЕНТУ

В даній статті описані алгоритми та основні принципи аналізу текстових даних для визначення наявностіспойлерів у тій чи іншійчастинітексту.Демонструється три моделі глибинного навчання, які вирішують поставлену задачу. На прикладі даних оглядів медіаконтенту та відгуків до них моделі порівнюються на основі метрик, які обчислюють їх ефективність.

КЛЮЧОВІ СЛОВА
АНАЛІЗ ТЕКСТОВИХ ДАНИХ, МЕДІАКОНТЕНТ, СПОЙЛЕР, РЕКУРСИВНІ НЕЙРОННІ МЕРЕЖІ

This article describes the algorithms and basic principles of text data analysis to determine the presence of spoilers in a particular part of text.Three models of deeplearning are demonstrated, which solve the problem. On the example of reviews of mediacontentan dresponses to them, the models are compared on the basis of metrics that calculate their effectiveness.

KEYWORDS
TEXT DATA ANALYSIS, MEDIA CONTENT, SPOILER, RECURSIVE NEURAL NETWORKS

1. Вступ

Аналіз текстових даних стає більш популярним з кожним роком. Даний комплексний процес складається з застосування методів перед-обробки текстових даних, застосування основних підходів до вирішення задачі на основі класичного машинного навчання та глибоких нейронних мереж. Текстові дані складають найбільшу частку інформації з поміж інших видів, саме тому аналіз цих даних є актуальною темою для дослідження. У основу машинного навчання взято концепцію навчання у процесі застосування рішень якоїсь множини задач. Тобто пряме вирішення задачі відходить на другий план.

Дана інтелектуальна система виконує аналіз текстових даних, за рахунок чого досягається класифікація текстових даних на ті, які містять або не містять спойлери.

Щодо технічної частини системи, то дана задача вирішується алгоритмами семантичного аналізу тексту (класифікації сентиментів), концепцій naturallanguageinference та імплементація рекурсивних нейронних мереж заснованих на цих алгоритмах, які виявляються досить ефективними.

Дана система написана на мовіпрограмуванняPython3 у середовищі JupyterNotebook та може бути інтегрована у будь-яку систему, яка містить опис медіаконтенту та необхідність визначення спойлерів. Прикладами можуть послужити веб-сервіси по типу IMDB, телеграм канали, де є можливість поділитися своїми думками щодо фільму тощо.

У наступних розділах буде коротко описано моделі та алгоритми, які було використано та їх порівняння.

2. Модель Manhattan LSTM

МодельManhattan LSTM[1] бере за основу принцип того, що, якщо умовно два речення (фрази) виражають одну і ту ж ідею – вони мають бути семантично схожі. А точніше семантична схожість має допомогти у визначенні схожих речень або фраз. Хороша модель не повинна залежати від кількості слів або синтаксису написання. Дана проблема фіксованої кількості слів та термінів до сих пір є актуальною через те, що такі моделі, як bag-of-words/tf-IDF (які є надзвичайно популярними у NLP), якраз дуже сильно спираються на специфічність термінів.

Як зрозуміло із назви цієї моделі, її основою є LSTM. LSTM – це різновид рекурентних нейронних мереж (RNN), які добре підходять через те, що приймають змінну довжину вхідних даних (речень або фраз у нашому випадку). Як відомо, LSTM використовує послідовні дані, де при кожній ітерації T виконуються оновлення вектора прихованого стану:

$$h_t = sigmoid(Wx_t + Uh_{t-1}) \#(1)$$

Manhattan LSTM – це покращення звичайного LSTM. У даній моделі береться дві мережі LSTM для кожного з порівнюваних текстів. Модель використовує LSTM для зчитування у векторів слів, що представляють кожне вхідне речення, і використовує його останній прихований стан як векторне представлення для кожного речення. Згодом подібність між цими поданнями використовується як предиктор семантичної подібності. Вектори порівнюються для того, щоб отримати оцінку схожості. Оцінка схожості рахується на основі Манхетенської відстані між векторами речень. Формула, яка використовується для підрахунку семантичної подібності зображена далі:

$$\exp\left(-\left|\left|h_{Ta}^{(a)} - h_{Tb}^{(b)}\right|\right|_1\right) \in [0,1], \#(2)$$

3. LSTM на основі attention механізму

LSTM на основі attention механізму [2] є надстройкою над попереднім через додатковий шар attention механізму. Тобто модель складається з двох частин: двонаправлена LSTM та attention механізм. По суті, attention механізм – це сума всіх попередніх векторів прихованого стану. Результуючі зважені вектори прихованого стану LSTM вважаються вбудованими реченнями.Для того, щоб закодувати речення змінної довжини у вбудоване речення фіксованого розміру, використовуєтьсяattention механізм, який виконує лінійну комбінацію прихованих векторів LSTM. Результатом цієї операції є вектор.

Так як великим реченням властиво мати декілька логічних частин, які розділені сполучниками, є необхідність у приміненніattention механізму на декількох частинах речення. Таким чином, attention вектор перетворюється у матрицю, яка містить усі вектори у кількості, яка дорівнює кількості логічних частин речення. Рівняння для attention матриці зображено далі.

$$A = softmax(W_{s2}\tanh(W_{s1}H^T)) \#(3)$$

Таким чином, для отримання вбудованого речення використовується матриця A та вектори прихованого стану H:

$$M = AH \ \#(4)$$

Дана модель також може перетворити будь-яку послідовність слів у сталу довжину. Як показали досліди, дана модель відповідно має перевагу при великому розмірі речення, що є великим плюсом зважаючи на розміри резюме до фільмів.

4. Enhanced LSTM (ESIM)

МодельEnhanced LSTM (ESIM)[3] заснована на одній із задач Naturallanguageinference.

Naturallanguageinference – це завдання визначити, чи є гіпотеза істинною, хибною чи невизначеною з урахуванням передумови. Наприклад, нехай передумовою буде «На вулиці дощить». Згідно з цією передумовою, гіпотеза типу «Дощ заливає вулиці міста» буде істинною, а гіпотеза типу «Вулиця знаходиться на пагорбі» буде хибною. Для задачі дисертаційної роботи невизначена гіпотеза не потрібна, так як потрібно у будь-якому випадку визначити чи частина тексту спойлер, чи ні.

Підзадача, яку повинна вирішити дана модель – це так називаємаentailment задача – співвідношення вірне, коли істинність одного фрагмента тексту випливає з іншого тексту.

З назви моделі стає очевидним, що для того, щоб знайти прихований вектор також використовується LSTM, але його двонаправлений різновид (BiLSTM). BiLSTM обчислює результат, запускаючи два

незалежних LSTM, один у прямому напрямку, а другий у зворотному напрямку. Потім отримані вектори об'єднуються.

Для отримання кінцевого результату текстові дані проходять три етапи.

Перший етап – кодування речень у вектори. Використовуючи BiLSTM, відображається зв'язок між словом у реченні та його контекстом.

Другий етап – процес підрахування attention. Використовуючи приховані вектори, які були отримані на попередньому етапі, підраховується матриця схожості.

Ваги матриці використовуються для обчислення контекстних векторів, що фіксують вигляд передумови для кожного слова в гіпотезі, і навпаки.

Потім модель використовує модуль генерації підкомпонентів.

Третій етап – агрегування результатів. Нарешті, для узагальнення результатів, другий цикл BiLSTM запускається для векторів підкомпонентів. Потім приховані стани поєднуються двома різними способами, обчислюючи максимальний та середній показники за часом (AvgPooling, MaxPooling).

5. Вхідні дані

Для порівняння алгоритмів було обрано набір даних, який містить опис кінострічок та оглядів на них. Найважливішими колонками датасету саме для даної задачі є опис сюжету кінострічки, текст відгуку користувача на цей фільм та прапорець, який повідомляє чи даний відгук є спойлером.Кількість вищеописаних даних наведена у табл. 1.

Табл. 1. Вхідні дані

Тип даних	**Кількість записів**
Описи кінострічок	1572
Відгуки користувачів	573913
Відгуки, які містять спойлери	150924

6. Порівняння роботи алгоритмів

Так як дана задача зводиться до класифікації тексту на той, що містить спойлери та той, що не містить спойлери, було обрано метрики, які засновуються на матриці помилок (confusionmatrix). Матриця помилок –це таблиця, яка показує ефективність алгоритму класифікації шляхом порівняння прогнозованого значення цільової змінної з її фактичним значенням. Основними термінами даної матриці є число вірно прогнозованих позитивних цілейTP, число фактично негативних цілей, які були спрогнозовані як позитивніFP, число фактично позитивних цілей, які були спрогнозовані як негативніFN та число вірно прогнозованих негативних цілейTN.

Отже, було обрано дані метрики для порівняння ефективності алгоритмів:

Точність (accuracy)–доля правильних передбаченихспойлерів до загальної кількості спойлерів.

Втрата (loss) – доля неправильно передбачених спойлерів до загальної кількості спойлерів.

Точність (precision) – доля спойлерів, які були визначені як позитивні.

Повнота (recall) – цевідношення правильно передбаченихспойлерів до всіхспойлерів.

F1 – цесередньозваженезначеннятočності (precision) та повноти (recall).

Precisiontarecall не залежить, на відміну від accuracy, від співвідношення класів і тому застосовні в умовах незбалансованих вибірок.

Для візуалізації значень метрик було побудовано графіки залежності значень accuracyталоss від номеру епохи. Також для порівняння значень метрики F1 булопобудовано таблиці, які відображають ці значення для обох типів класифікації – «спойлер» та «не спойлер».

Результати для методу Manhattan LSTM наведені на рис. 1, рис. 2 та табл. 2.

Результати для методу LSTM на основі attention механізму наведені на рис. 3, рис. 4 та табл. 3.

Результати для методуEnhanced LSTM (ESIM) наведені на рис. 5, рис. 6 та табл. 4.

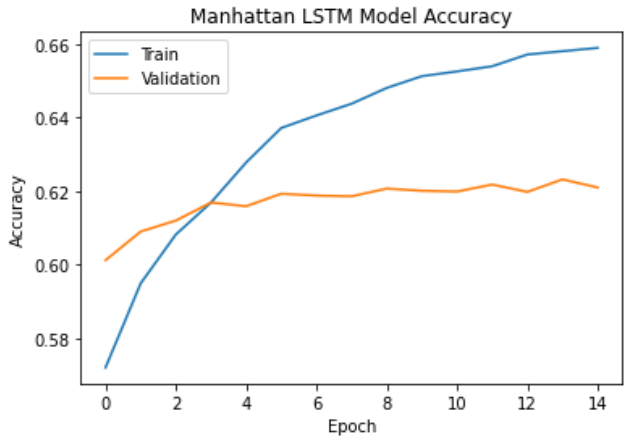


Рис. 1. Accuracyдля моделі Manhattan LSTM

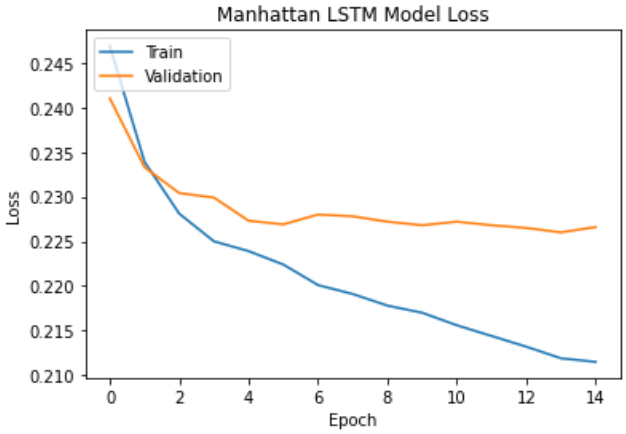


Рис. 2. Lossдля моделі Manhattan LSTM

Табл. 2. Результати метрик для моделі Manhattan LSTM

	Precision	**Recall**	**F1**
Спойлер	0.71	0.14	0.23
Не спойлер	0.74	0.96	0.84

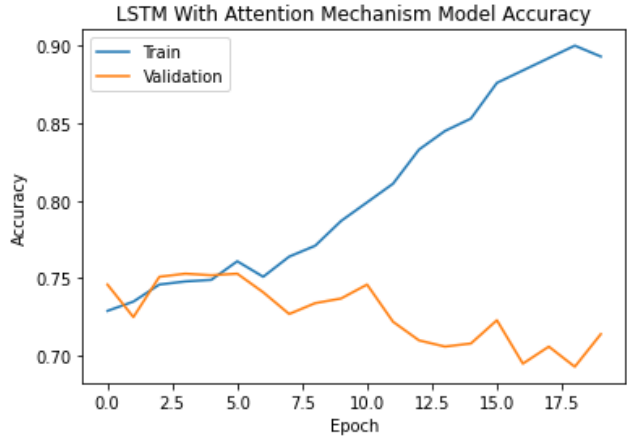


Рис. 3. Accuracyдля моделі LSTM на основі attention механізму

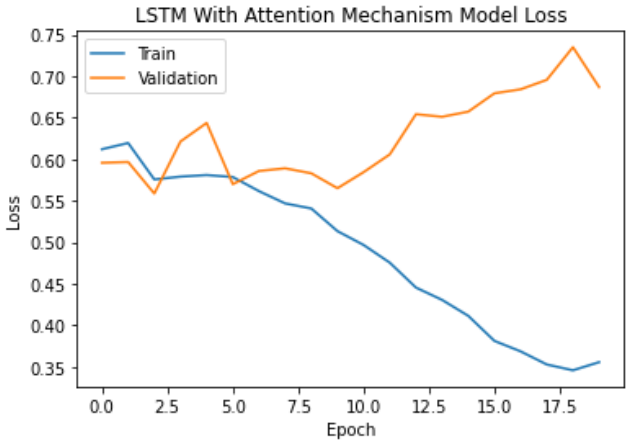


Рис. 4. Lossдля моделі LSTM на основі attention механізму

Табл. 3. Результати метрик для моделі LSTM на основі attention механізму

	Precision	**Recall**	**F1**
Спойлер	0.36	0.56	0.44
Не спойлер	0.38	0.55	0.45

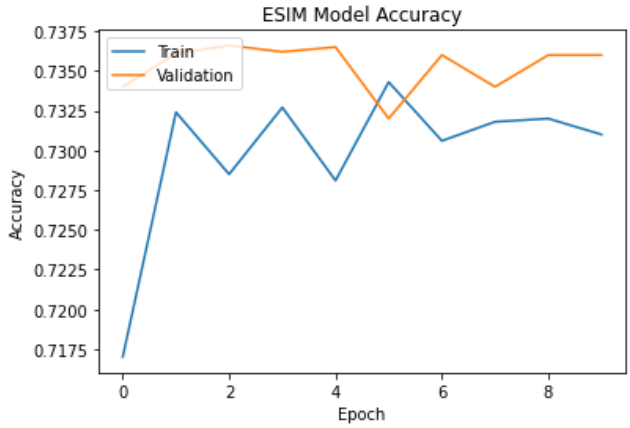


Рис. 5. Accuracyдля моделі Enhanced LSTM (ESIM)

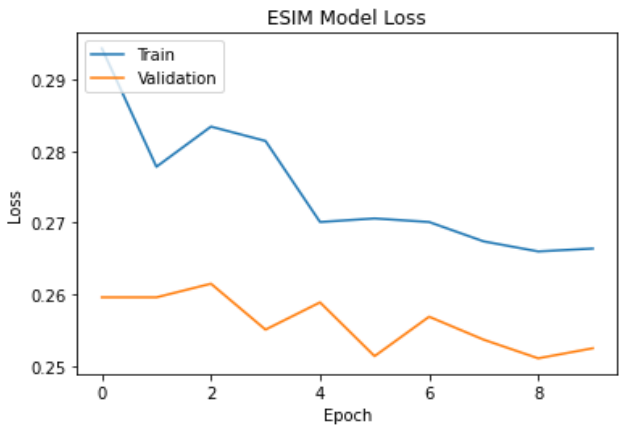


Рис. 6. Lossдля моделі Enhanced LSTM (ESIM)

Табл. 4. Результати метрик для моделі Enhanced LSTM (ESIM)

	Precision	**Recall**	**F1**
Спойлер	0.64	0.69	0.66
Не спойлер	0.66	0.62	0.64

Як видно із результатів для методу Manhattan LSTM, recall для «спойлерів» є 0.14, що може казати про те, що дана модель буде рідше позначати справжні спойлериспойлерами. Натомість F1 для «не спойлерів» є дуже високим. Це каже про те, що дана модель позначає більшість текстів такими, що не є спойлерами, що не є ідеальним.

Як видно із результатів для методу LSTM на основі attention механізму середня accuracy є досить високою– на рівні 0.72, але показник loss більший, аніж у попередній моделі –~0.63. Також незважаючи на те, що показник F1 є однаковим для обох класифікацій, проте він не є більшим аніж у попередній моделі.

Як видно із результатів для методу Enhanced LSTM (ESIM) показник середньої втрати є маленьким та майже рівний тому для методу Manhattan LSTM. НатомістьF1 для обох класифікацій є майже однаковим та вищим аніж у Manhattan LSTM та LSTM на основі attention механізму. Це говорить про те, що дана модель має меншу залежність від відношення кількості«спойлерів» до «не спойлерів» у вхідних даних.Також показник точності даної моделі, що дорівнює ~0.73, є найвищим серед усіх моделей.

Отже, можна зробити висновок, що модель Enhanced LSTM (ESIM) загалом є покращенням, на відміну від Manhattan LSTM та LSTM на основі attention механізму, адже має найвищі показникиaccuracy, F1 для обох видів класифікаціїта показник loss, що дорівнює ~0.25.

Висновок

Аналіз текстових даних є комплексною задачою, яка може бути вирішена безліччю способами. У даній роботі було зроблено огляд методів заснованих на рекурентних нейронних мережах із застосуванням модифікацій, таких як attentionмеханізм, LSTM та двонаправленаLSTM.Моделі викладені у даній роботі були порівняні використовуючи метрики, такі як accuracy, loss, precision, recall та F1.Для показників accuracy та loss було побудовано графіки їх значень на кожній епосі тренування моделі. Для показників precision, recall та F1 було побудовано таблиці їх порівняння під час класифікації текстів на «спойлери» та «не спойлери». На основі побудованих графічних та числових значень метрик, було обрано модель, яка є найбільш ефективною при аналізі даних оглядів медіаконтенту та відгуків на них. Продовженням даної роботи є реалізація автоматизованої системи, яка з допомогою найефективнішої моделі зможе визначати спойлери з-поміж тексту відгуків користувачів. Звичайно існують і шляхи для покращення аналізу цих специфічних текстових даних, наприклад,врахування історії попередньо написаних відгуків для кожного окремого користувача або залучення рейтингу, який користувач поставив кінострічці для більш точного визначення спойлерів.

Список літератури

1. JonasMueller, Aditya ThyagarajanSiamese Recurrent Architectures for Learning Sentence Similarity – 2016.
2. Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, YoshuaBengioA Structured Self-attentive Sentence Embedding – 2017.
3. Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, Diana InkpenEnhanced LSTM for Natural Language Inference– 2017.

УДК 004.67

ВАЛЬЧУК Д.В.
ГОЛОВЧЕНКО М.М.

АРХІТЕКТУРНЕ РІШЕННЯ ДЛЯ ОБРОБКИ ВЕЛИКИХ ОБСЯГІВ СТАТИСТИЧНИХ ДАНИХ НА ПРИСТРОЯХ З НИЗЬКИМИ ТЕХНІЧНИМИ МОЖЛИВОСТЯМИ

В даній статті описані основні технології та алгоритми для вирішення типових задач для збору та обробки великих масивів даних на середніх за потужністю пристроях.

КЛЮЧОВІ СЛОВА:
Аналіз даних, машинне навчання, відображення файлів в пам’ять, набір даних,python,vaex, apachespark.

This article describes the basic technologies and algorithms for solving typical problems for collecting and processing large data sets on medium-power devices.

KEYWORDS:
Data analysis, machine learning, memory mapping, dataset, python,vaex, apachespark.

1. Введення

Розвиток інформаційних технологій з кожним роком займає все більше значення у багатьох галузях людської життєдіяльності,це стимулює технічний прогрес у цілому і супроводжується створенням великої кількості даних. Ці дані використовуються для різних цілей, таких як аналіз потреб користувачів, пошук закономірностей виникнення різних захворювань та багато інших. Але, дані стають недоцільними, коли з них неможливо вилучити корисну інформацію. Завдяки цьому, все з більшою швидкістю розвиваються такі напрями інформаційних технологій як машинне навчання, штучний інтелект та аналіз даних.

Для швидкого аналізу великих об’ємів даних розроблено безліч інфраструктурних рішень. Основним та найоптимальнішим, на мою думку, є використання хмарних технологій. Використання даної технології є дуже зручним, так як ви маєте можливість масштабувати технічні можливості обчислювальних пристроїв у відповідності з вашими потребами. Але з іншого боку, це коштує достатньо дорого і з економічної точки зору не є завжди доцільним. Також, у будь-якому випадку, вам доведеться шукати спеціаліста у даній галузі для розгортання обраної інфраструктури.

Розроблене архітектурне рішення дозволяє проводити аналіз даних на звичайному ПК чи ноутбуці, що дозволяє отримати статистику для будь-кого.

Також, дане архітектурне рішення дозволяє швидко завантажити набір даних до веб-інтерфейсу та в один клік отримати загальну статистику для завантаженого вами набору даних.

2. Внутрішня структура бібліотеки

Архітектурне рішення являє собою веб-додаток розроблений на мові програмування Python. Для розробки веб-додатку використовувавсяфреймворкDjango.

Даний фреймворк застосовує шаблон проектування MVC та використовується для написання клієнт-серверної архітектури.

Для завантаження великої кількості даних було розроблено модуль, який отримує на вході файли у вигляді архівів даних, після чого на стороні сервера розпаковує набір заархівованих даних у файли в форматі csv.

Для більш швидкого завантаження даних були використані асинхронні методи завантаження, а також методи розбиття файлу великого об’єму на дрібні частини, що значно пришвидшуєцей процес.

Далі, для обробки даних було використано бібліотеку Vaex, яка є бібліотекою з відкритим кодом, що може обробляти файли співрозмірні з розміром дискового простору, на якому розгорнуте архітектурне рішення[1]. В її основі лежить використання технології memorymapping, яка дозволяє

працювати з файлом не завантажуючи його до оперативної пам’яті комп’ютера цілком, а ставити його у відповідність з деякою ділянкою пам’яті та проводити операції зчитування та запису одразу у файл[2]. Дана бібліотека використовує «ліниві» механізми обробки даних, що дозоляє вираховувати значення полів лише тоді, коли вони необхідні.

Для використання обраної бібліотеки обробки даних, завантажені на сервер файли наборів даних у форматі csv необхідно привести до формату HDF5 або ApacheArrow. У розробленому архітектурному рішенні було обрано формат HDF5. Для цього було розроблено модуль для конвертування форматів та збору їх у єдиний файл[3].

Для швидкого отримання загальної інформації для завантаженого набору даних були реалізовані методи фільтрації даних, пошуку аномалій у даних та реалізовані агрегатні функції.Приклад структури зазначенона рис. 1.

3. Використання розробленої архітектури

Для використання даного архітектурного рішення достатньо розгорнути проект у локальному середовищі та перейти у веб-інтерфейс. Далі необхідно знайти набір даних та завантажити його до веб-інтерфейсу. Після чого необхідно дочекатися завантаження файлів на сервер та приведення їх до необхідного формату. У кінцевому результаті можна отримати загальну статистику за завантаженим набором даних.

Основні положення використання бібліотеки представлені на рис.2.

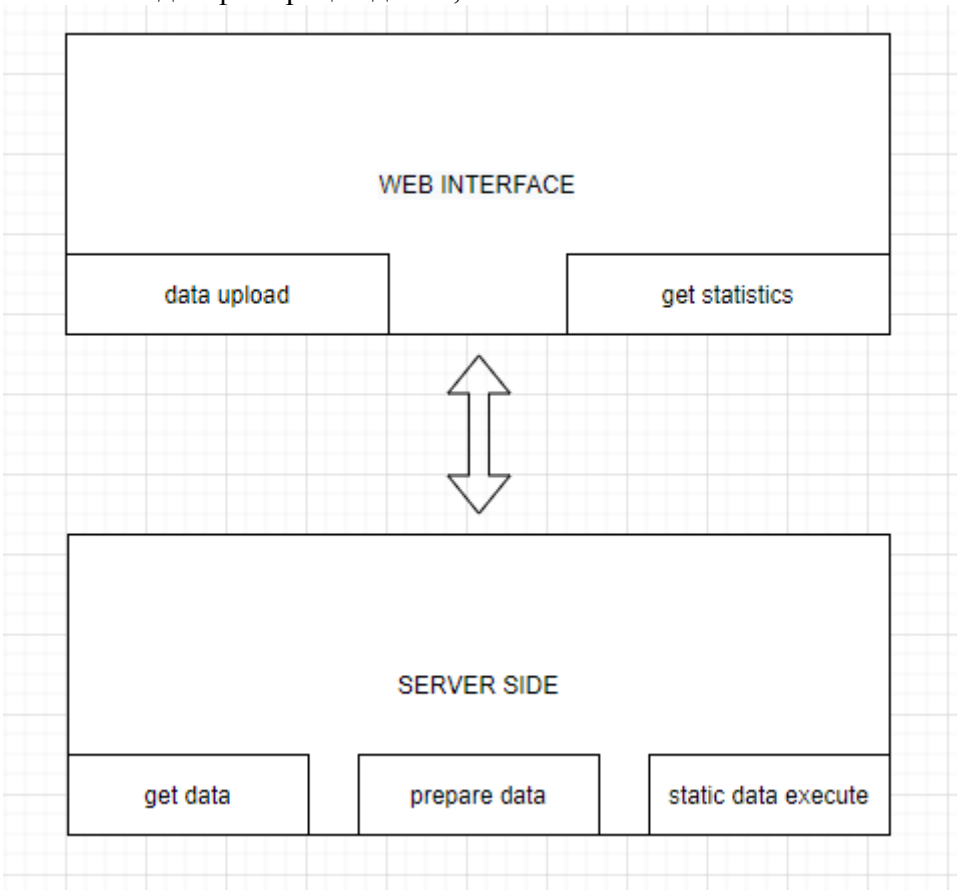


Рис. 1. Схема внутрішньої структури архітектури

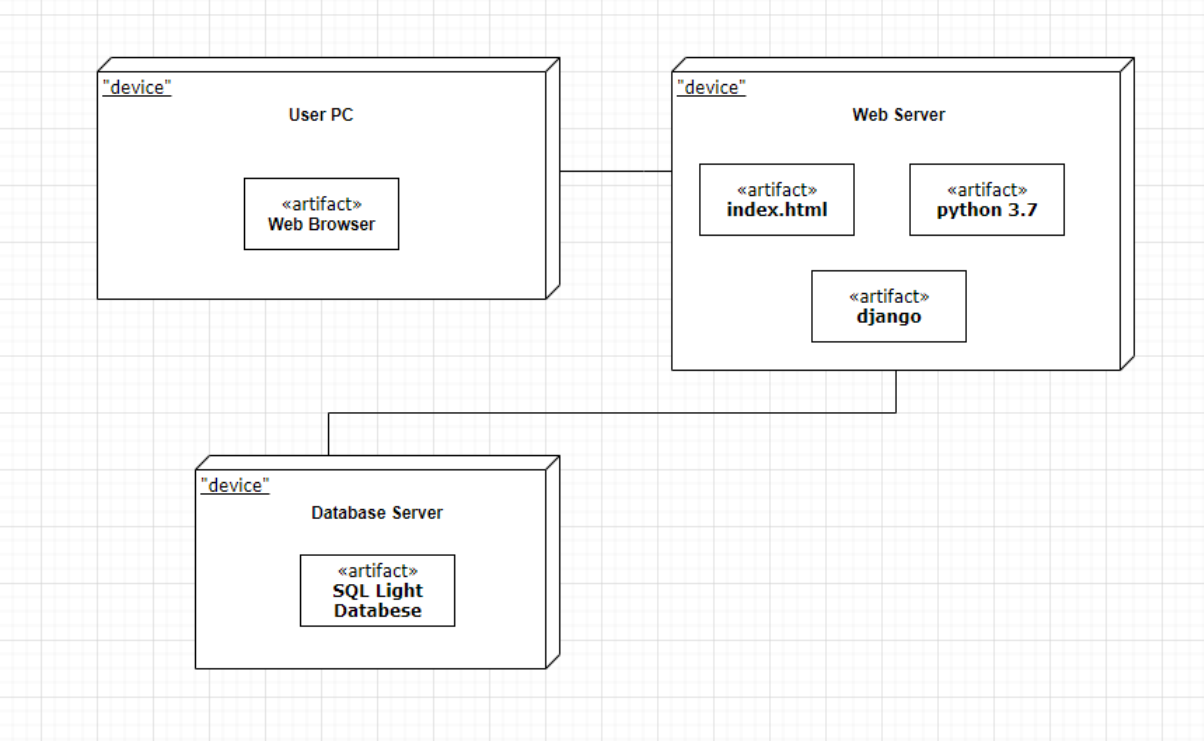


Рис.2. Схема розгортання веб-додатку з використанням обраної архітектури

4. Оцінка ефективності розробленої архітектури

Для проведення оцінки ефективності розробленої архітектури було обрано набір даних авіаперевезень США з 1995 по 2019 рік. Загальний обсяг обраного набору даних склав близько 20 гігабайтів. Для більш швидкої передачі даних на сервер набір даних було заархівовано, результатом проведення цієї операції став файл розміром 6,26 гігабайтів.

Після завантаження даних на сервер було розархівовано обраний набір даних і приведено до формату hdf5. Паралельно цьому процесу на персональному комп'ютері було розгорнуто інфраструктуру ApacheSpark та завантажено у неї обраний набір даних. У результаті було проведено порівняння швидкості виконання таких запитів як відкриття набору даних, фільтрація та виконання запитів. Результати тестів зазначені у табл. 1.

Табл. 1. Результати тестів

Параметр	Архітектурне рішення	ApacheSpark
Завантаження набору даних	0:00:00.813978	0:52:02.785845
Виконання фільтрації даних	0:00:00.300000	0:25:58.352359
Запит на знаходження кількості рейсів за певні роки	0:03:03.838997	0:26:04.552372

Висновок

Підсумовуючи вищезазначене у дослідженні, можна зробити висновки, що актуальність сфери аналізу даних з кожним роком лише збільшується. Для перевірки ефективності використання розробленого архітектурного рішення та швидкості дії виконання запитіву ньому, було проведено його порівняння з технологією ApacheSpark. Якщо розгортати цю технологію локально, то перевага у швидкодії залишається на стороні розробленого архітектурного рішення, але якщо використовувати технологію ApacheSpark разом з

хмарними обчисленнями, то можна сказати, що ця технологія практично не має обмежень у швидкодії за рахунок можливості збільшення обчислювальних пристроїв.

Крім того, було явно продемонстровано переваги використання бібліотеки Vaex порівняно з такими як pandas або Dask за рахунок використання підходу memorymapping та «лінивих»механізмів обробки даних.

Список літератури

1. Howtoprocess a DataFramewithbillionsofrowsinseconds. [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/how-to-process-a-dataframe-with-billions-of-rows-in-seconds-c8212580f447.
2. InternationalJournalofInformationTechnologyandKnowledgeManagementJuly-December. – 2009. – Vol.2. – No.2. – P.365-370.
3. Anoverviewofthe HDF5 technologysuiteanditsapplications – 2011.

УДК 004.93

ПРІСЕНКО Д.С.
ГОЛОВЧЕНКО М.М.

ВИЯВЛЕННЯ, АНАЛІЗ ТА ВИДАЛЕННЯ ПОШКОДЖЕНЬ НА СТАРИХ ФОТОКАРТКАХ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖІ

В даній статті описані основні етапи реставрації старих фотокарток, аналіз зображення на предмет механічних пошкоджень та їх подальше видалення з використанням нейромережі. Демонструється порівняння різних методів та підходів до загального питання ретуші фотокарток за допомогою нейромереж.

КЛЮЧОВІ СЛОВА
КОНВУЛЯЦІЙНАНЕЙРОМЕРЕЖА, РЕТУШ, ФОТОКАРТКИ, ЗГОРТКА, СЕГМЕНТАЦІЯ, МОДЕЛЬ UNET

This article describes the main stages of restoration of old photographs, image analysis for mechanical damage and their subsequent removal using a neural network. There is a demonstration of different methods to common task of photo-retouching with a neural network.

KEYWORDS
CONVOLUTIONALNEURAL NETWORK, RETOUCHING, PHOTOGRAPHS, SAMPLING, SEGMENTATION, UNET MODEL

1. Введення

У наш час людство не має важливішого пріоритету щодо власного розвитку суспільства, аніж розвиток науки та технологій. Ми живемо у еру великого технологічного прориву, коли з кожним днем нові цифрові рішення просочуються у кожну сферу суспільного життя. Але найцікавіші події, як на мене, з'являються там, де новітні технології зіткаються з тим, що людина створює на емоційному рівні дуже давно. Естетика як явище йде поруч з людством ще с часів перших петрогліфів, вона змінюється разом з особистістю.

Перша у світі фотографія була зроблена Жозефом Нєпсом у 1822-му році, але не дійшла до наших часів. Взагалі, фотографія як вид мистецтва навряд колись вже зникне з історії людства. Фотокартка – це фізичний носій інформації, якихось спогадів, важливих для людини. Але з плином часу фотокартки втрачають свій зовнішній вигляд, та, відповідно, і частину інформації.

Як можна вирішити цю проблему? Раніше цім займалися окремі люди – реставратори. Вони вручну дарували зображенню нове життя, повертали втрачені спогади. Зараз ситуація дещо інакша – реставрувати

фотокартки можна і за допомогою нейромереж.

2. Пошук пошкоджень

Все починається з питання до відповідної задачі – як за допомогою штучного інтелекту можна зімітувати дію живої людини, що ретушує старі фотокартки? Для того щоб відповісти на це питання, я дослідив ручний процес ретуші та проаналізував певні етапи.

Будь-яка ретуш починається з того, що «ретушер» шукає пошкодження, з якими у подальшому буде працювати. Коли розмова йдеться про штучний інтелект, то він не може зразу зрозуміти які саме пікселі растрового зображення є пошкодженнями. Тому спочатку необхідно обучити мережу.

Перший етап аналізу – це з'ясувати, які саме фотокартки будуть у більшості випадків завантажувати користувачі. Більшість старих фотографій – це портрети або групові фото, на яких є велика кількість механічних пошкоджень. Тому все починається зі збору первинної навчальної вибірки. Що собою являє ця вибірка? Навчальна вибірка в задачі сегментації – це загальне зображення та маска, яка представляє собою фізичне уявлення тих пошкоджених пікселів, з якими у подальшому мережа буде працювати. Простіше за все віддати фотокартки на обробку ассесорам. Звісно що люди вміють находити певні дефекти, але розмітка пошкоджень – дуже довгий процес. Розмітка однієї світлини може займати від декількох годин до цілого дня, тобто зробити вибірку більш ніж на 100 фотографій менше ніж за декілька тижнів дуже складно. Тому можна використовувати звичайні «чисті» світлини та наносити штучні дефекти у графічному редакторі, одразу отримуючи потрібну маску. Це дуже прискорює процес створення навчальної вибірки.

Найпоширеніший підхід у подібних задачах сегментації – це використання UNETз заздалегідь навченим енкодером та мінімізація суми BCE (binary-crossentropy) та DICE(Sørensen–Dicecoefficient)[1]. Бо навіть якщо нам здається що дефектів на фотокартці дуже багато, то площа зображення без пошкоджень все одно більше за пошкоджену. Тому можна запропонувати збільшити вагу позитивного класу BCE та за оптимальну вагу брати кількість усіх «чистих» пікселів до кількості пошкоджених.

Після того, як нейромережа стає здатною до самостійного аналізу світлини на предмет пошкоджень та здатна створювати необхідну маску, наступним кроком є задача видалення усіх дефектів.

На рис.1 наведено загальний вигляд первинної фотокартки та маски на прикладі старої фотографії Київського Політехнічного Інституту.

Рис. 1.Приклад створення маски з використанням нейромережі

3. Виправлення дефектів

Для рішення задачі виправлення дефектів знов найпоширенішим способом є використання мережі UNET. Вхідними даними до роботи є початкове зображення та маска, де одиницями представлені чисті пікселі, та нулями пошкоджені (ті, що хочемо виправити). Але як можна модифікувати готовий UNET у контексті загальної задачі та

оптимізувати його роботу? Після ряду експериментальних досліджень було з’ясовано, що використання часткової згортки значно оптимізує час обробки одного зображення, та візуально покращує результат виправлення дефектів. Ідея PartialConvolution полягає у тому, що при згортанні регіону зображення з якимось ядром ми не враховуємо значення пікселів, відносних до дефектів. Такий підхід було запропоновано компанією NVIDIAв одній зі своїх статей [2], результат див. на рис. 2.

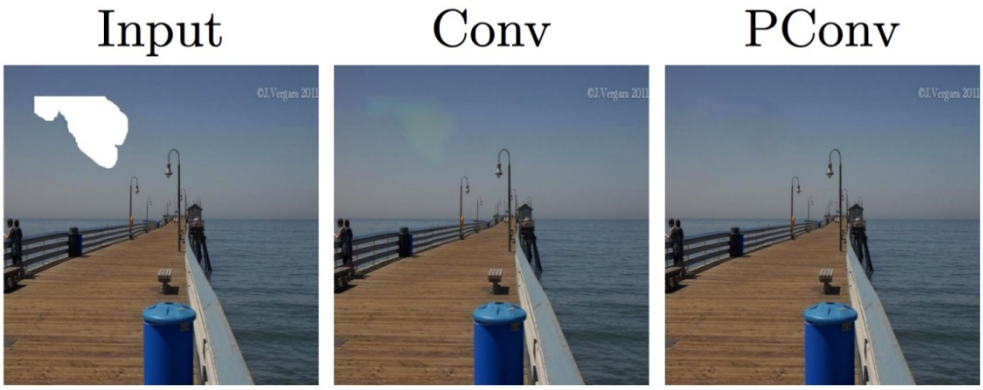


Рис. 2.Порівняння повної та часткової згортки

Картинку розміром 512x512 мережа обробляє за 50 мс. ВалідаційнийPSNRдорівнює 26,4. Але у подібних задачах, не зважаючи на чисельні метрики, ще дуже важливо постійно спостерігати за візуалізацією результату. Бо принцип роботи алгоритму запростий – нейромережа приймає на вхід маску з пошкодженими пікселями та намагається виправити їх з використанням чистих матричних пікселів, що знаходяться максимально близько до перших. Та все ж якщо мережа оперує недостатніми вибірками, то візуально результат ретуші може бути недосконалої якості. Як це контролювати? Робота мережі с дефектами різного розміру так чи інакше залежить від того, якого розміру дефекти були на світлинах з навчальної вибірки. Тобто, якщо ви думаєте що у подальшому вам потрібно буде працювати з фотокартками, де площа пошкоджень дуже велика, то і наповнювати навчальну вибірку треба згідними фотографіями. Чим більше варіацій пошкоджень (загальної площі, форми, розмірів, положення)[3] ви зможете надати нейромережі, тим краще буде візуальний результат.

4. Результати дослідження

Пропоную розглянути результати на прикладі старої фотокартки Київського Політехнічного Інституту. Об’єктом дослідження були три підходи до ретуші світлини – звичайна (повна) згортка, часткова згортка (PartialConvolution) та метод локалізованої реконструкції.

Для виявлення точності використовувався датасет старих фотокарток (200 тисяч зображень). Як можна побачити на порівняльному графіку (рис.3), різниця показника PSNRзі збільшенням кількості епох стає менш помітною. Також необхідно пам’ятати про візуальну складову – використовуючи метод повної згортки, візуальний результат все ще має помітні огріхи (рис.4а). Метод часткової згортки та метод локалізованої реконструкції показали кращі результати щодо ретуші (рис.4б та рис.4в).

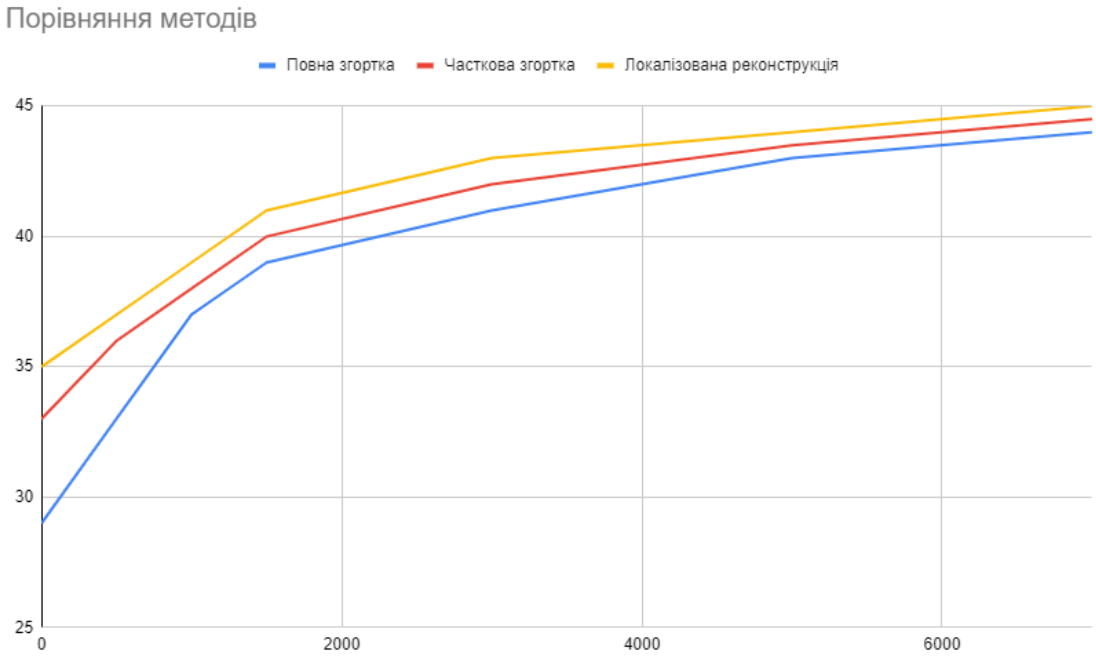


Рис. 3. Графік порівняння методів аналізу та ретуші фотокарток

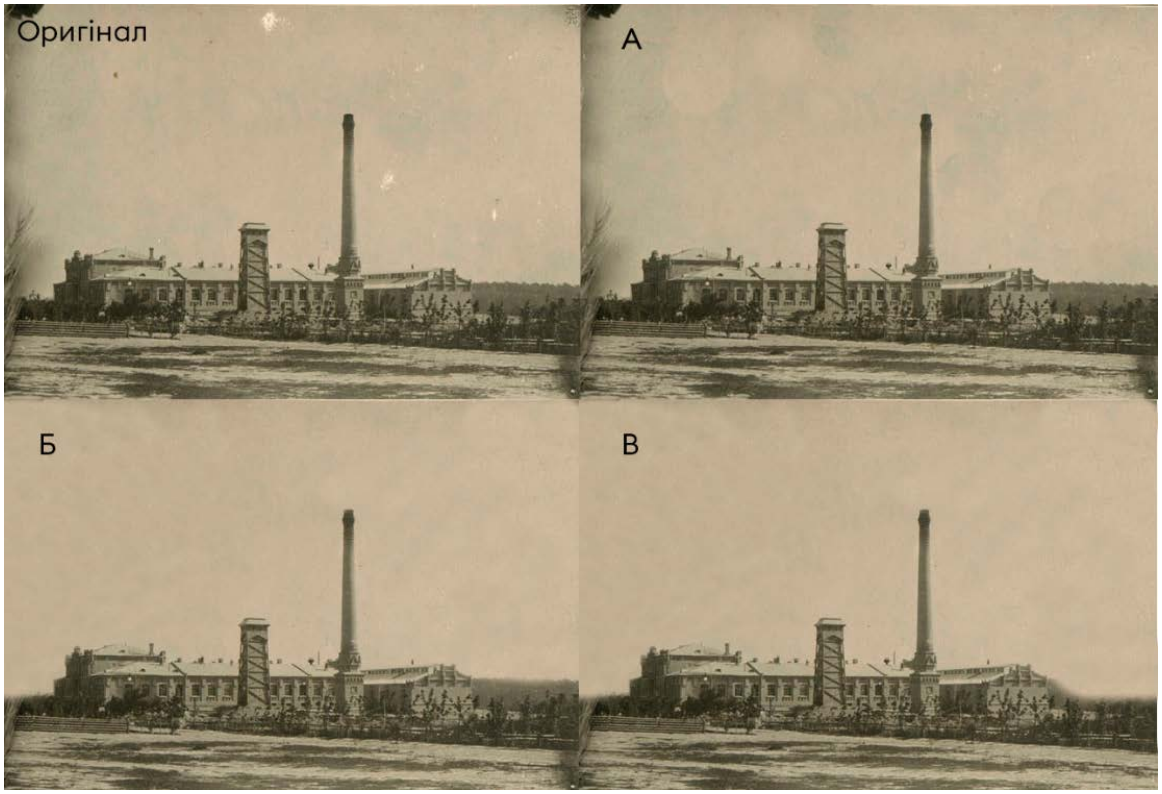


Рис. 4.Результати ретуші (А – метод повної згортки, Б – метод часткової згортки, В – локалізована реконструкція)

Висновок

У результаті дослідження я також з’ясовував, що модель UNETпоказує гарні результати у подібних задачах. В першій задачі сегментації я зіткнувся з проблемою навчання та роботи з зображеннями великої роздільної здатності. Тому було прийнято рішення використовувати In-PlaceBatchNorm. Це також допомогло мені оптимізувати необхідний фактичний час роботи.

Також у результаті дослідження була підтверджена важливість ассесорів. При чому, вони необхідні як на первинному етапі створення навчальної вибірки, так і на етапі валідації результату. Нажаль, у задачах ретуші та колоризації дуже важливо отримати валідацію

кінцевого рішення у людини. У подальшому, після того як нейромережа стає здатною до ефективного пошуку, аналізу, та виправлення певних дефектів, можна починати займатися задачею колоризації, або використовувати чисту фотографію для, можливо, інших задач.

Список літератури

1. Matthew Hutson. AI Can Edit Photos With Zero Experience
2. Guilin Liu;Fitsum A. Reda; Kevin J. Shih; Ting-Chun Wang; Andrew Tao; Bryan Catanzaro. Image Inpainting for Irregular Holes Using Partial Convolutions
3. Henderson, Thomas C. Analysis of Engineering Drawings and Raster Map Images

УДК 004.021

KOCHUBEY I.,
ZHURAKOVSKA O.

ALGORITHM OF FORMING RECOMMENDATIONS FOR USERS OFWEB DIRECTORIES

In this paper, the problem of forming recommendations to users of web directories was considered. An algorithm for the formation of recommendations is proposed, which is a modified algorithm of collaborative filtering. The efficiency of the developed algorithm has been researched. The results show its high efficiency and expediency of use it in recommendation systems.

KEYWORDS:RECOMMENDER SYSTEM, COLLABORATIVE FILTERING, WEB DIRECTORY

В цьому дослідженні розглянуто проблему формування рекомендацій користувачам веб каталогів. Запропоновано алгоритм формування рекомендацій, який є модифікованим алгоритмом колаборативної фільтрації. Проведено дослідження ефективності розробленого алгоритму. Результати показують його високу ефективність та доцільність використання в рекомендаційних системах.

КЛЮЧОВІСЛОВА:РЕКОМЕНДАЦІЙНА СИСТЕМА, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, ВЕБ КАТАЛОГ

1. Introduction

Web Catalog is a site that serves as an online showcase with a large number of product names. A web directory usually has recommender system [1]. There is a problem in work of the recommender systems. The number of objects in database continues to grow and there comes a time when recommender systems can give recommendations just over a longer period of time [2]. This problem is growing up.

In modern recommender systems, different methods are used to solve the problem of making personalized recommendations:

- neural collaborative filtering [3];
- latent factor models [4];
- the neighborhood approach [4];
- item-based algorithm of collaborative filtering [5];
- differential privacy protection [6].

Proposed algorithm of forming recommendations for users of web directory is based on algorithm of collaborative filtering.

2. Problem Statement

We find method to improve the algorithm of collaborative filtering.The problem is that there are too many products and it is usually not easy for the user to choose something. The user does not always trust web directory information but does trust the other users' recommendations.

There are many users that belong to the same cluster obtained as a result of the previous stage of the clustering task.Users judge objects. It is necessary to analyze the information that users of the web directory have put up and for a given user to identify many objects that he would like most - to form recommendations.

3. Substantive statement of the problem

Input data:

- B is a set of objects;
- b_i is an element i of set B;
- U is a set of users;
- u_j is an element j of the set U;
- i is an index of a recommended object.

The target function is

$$R = \{b_i \,|\, b \in B, i = \overline{1,c}\},$$

where R is a set of recommendations, and c is number of recommended objects.

4. Algorithm of forming recommendations

We proposed the algorithm of forming recommendations for users of web directories. This algorithm is based on algorithm of collaborative filtering [7]:

1. Select users from cluster.
2. Combining users into groups and searching for the average score for the object.
3. Search for similarity measures between groups and generalized scores relatively selected.
4. Multiplication of estimates by similarity measures.
5. Finding the overall assessment for the object in the cluster.
6. Dividing of the received number of estimates by the amount of measures.
7. Determining the rating for recommendations.

The main idea of improving the algorithm of collaborative filtering is to aggregate users in process of forming recommendation to improve runtime of algorithm [8].

5. Researching

Efficiency studies of the developed algorithm were conducted. The dependence of the result quality and algorithm performance on the following parameters was analyzed:

User count

The number of users is determined by the size of the cluster. A normal cluster contains from 30 to 100 people inside.

Number of objects

New systems have many objects, but to demonstrate the efficiency of the algorithm we can use a small number of objects from 200 to 1000 objects.

We used two criteria to compare the algorithm of collaborative filteringwithdeveloped algorithm:

- Runtime of the algorithm;
- the relative distance between the two algorithms.

Conclusion

The developed algorithm is faster than algorithmof collaborative filtering(by 1000 objects, 50 users the runtime of developed istentimes faster).The deviation of results of the developed and algorithmof collaborative filtering for 1000 objects, 50 users does not exceed 5%.The conducted research confirms high efficiency of the developed algorithm.It allows to use the developed algorithm in recommendation systems to form recommendations to users of web directories.

References

1. V. Srikar and R. Sudha,“Examining Lists on Twitter to Uncover Relationships Between Following, Membership and Subscription,” in *Proceedings of the 22nd international conference on World Wide Web*, Rio de Janeiro, Brazil, 2013 pp. 673-676.
2. P. Chatterjee, M. Yazdani, S. Chakraborty, D. PanchalSubro, *Advanced Multi-Criteria Decision Making for Addressing Complex Sustainability Issues*, Hershey, PA: IGI Global, 2019.
3. X. He et al.,“Neural Collaborative Filtering,” in *Proceedings of the 26th International Conference on World Wide*,Perth, Australia, pp. 173–182.
4. Y.Koren, R.Bell, “Advances in Collaborative Filtering,”in *Recommender SystemsHandbook*, Boston, MA: Springer, 2015pp. 77-117.
5. Sarwaret al.,“Item-Based Collaborative Filtering Recommendation Algorithms,” in *Proceedings of the 10th international conference on World Wide WebApril*,Hong Kong, China, 2001,pp. 285–295.

6. C. Yin, et al., “Improved collaborative filtering recommendation algorithm based on differential privacy protection,” in *The Journal of Supercomputing*,Hong Kong, China, 2020, pp. 5161–5174.
7. P.Melvilleet al.,“Content-Boosted Collaborative Filtering for Improved Recommendations,” in *National Conference on Artificial Intelligence*, Edmonton, Canada, 2016, pp. 187-192.
8. I. Kochubey, O. Zhurakovska, “Modified algorithm of collaborative filtering for forming user recommendations,” in*KPI science news*, 2020, pp. 43-49.

УДК 004.852

ЛОГВИНОВСЬКИЙ Є.О.,
ЖАРІКОВ Е.В.

ПРОГНОЗУВАННЯ ПОПИТУ НА АСОРТИМЕНТ ТОВАРІВ СУПЕРМАРКЕТУ

У роботі запропонована підсистема прогнозування споживчого попиту на асортимент товарів супермаркету на основі штучної нейронної мережі. Наведено фактори формування попиту та обґрунтовано застосування штучних нейронних мереж для вирішення задачі прогнозування попиту, що враховує комплекс взаємопов’язаних факторів, які визначаються як специфікою, так і особливостями споживання товарів підприємств роздрібної торгівлі, що дасть можливість підвищити ефективність управління підприємством. Проведено тестування програми прогнозування та проаналізовано результати.

КЛЮЧОВІ СЛОВА: ПОПИТ, ПРОГНОЗУВАННЯ, СУПЕРМАРКЕТ, МОДЕЛЮВАННЯ, ПЕРСЕПТРОН, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ.

This article examines the process of consumer demand forecasting for assortment of supermarket goods, describes the factors of demand generation and using the artificial neural networks to solve the problem of demand forecasting, which takes into account a set of interrelated factors that determine both the specifics and characteristics of retail goods consumption, to improve management. Testing and analysis of forecasting results were performed.

KEYWORDS: DEMAND, FORECASTING, SUPERMARKET, SIMULATION, PERSEPTRON, ARTIFICIAL NEURAL NETWORKS.

1. Вступ

У сучасному світі в умовах жорсткої конкуренції успішна діяльність підприємств роздрібної торгівлі неможлива без прогнозування попиту. В ринковому середовищі основою успішної діяльності підприємств роздрібної торгівлі є дослідження, аналіз та задоволення потреб споживачів, адже саме вони формують споживчий попит на асортимент товарів. Зміна доходів споживачів, а відповідно, їх купівельної спроможності, зміна цін на товари, вплив цілої низки інших факторів на попит, вимагає від підприємств мати чіткий план прогнозу попиту на товари, які продаються. Дослідження та прогнозування споживчого попиту є однією з найзатребуваніших і найскладніших задач як кон’юнктурного, так і стратегічного аналізу ринку. Розробка маркетингової стратегії ринку виробництва та збуту товарів потребує науково обґрунтованих прогнозів перспектив розвитку ринку. Прогнозування та дослідження споживчого попиту може дати відповіді на такі питання: який об’єм, структура, рівень попиту, яка тенденція зміни попиту та її швидкість, які фактори визначають попит в досліджуваному періоді, що передбачається в майбутньому. Крім того, прогнозування попиту є найважливішим критерієм доцільності інвестицій у виробництво товарів.

Прогнозування може виконуватися для різних подій та випадків, однак саме для економічних величин на сьогодні існує найбільше способів та методів передбачення, що активно застосовуються при плануванні виробничої зайнятості, торговельної активності та інших видів діяльності сучасних підприємств. Прогнозування сприяє збільшенню майбутніх доходів, забезпечує розвиток та розширення діяльності підприємств, мінімізує економічні ризики.

Прогнозування прибутків є важливим як для великих підприємств, так і для малих. Так до середнього бізнесу, що іноді реалізується силами фізичних осіб-підприємців, юридичних осіб відноситься робота супермаркетів – магазинів, що здійснюють торгівлю широким асортиментом товарів. Безперечно, такі об'єкти присутні на сьогоднішній день у кожному населеному пункті. Зважаючи на широке поширення відповідного бізнесу в Україні, задача прогнозування попиту на товари відіграє важливу роль в економічному просторі.

Отже, завдання моделювання і прогнозування споживчого попиту на асортимент товарів підприємств роздрібної торгівлі є актуальним, а впровадження таких моделей дозволить підвищити економічну ефективність і рентабельність роботи підприємств.

2. Суть задачі прогнозування попиту на асортимент товарів супермаркету

Супермаркети – це великі магазини самообслуговування, що забезпечують населення певним асортиментом товарів як продовольчої, так і непродовольчої груп (товарами широкого вжитку – ТШВ) [1]. Краще розуміння специфіки саме цього виду підприємницької діяльності надає дерево цілей супермаркету, наведене на рис. 1.

Рис. 1. Дерево цілей супермаркету

У результаті аналізу особливостей діяльності по забезпеченню населення товарами здійснена формалізація проблеми прогнозування в цілому та попиту на асортимент товарів супермаркету зокрема.

У загальному випадку поточний стан будь-якої системи, для якої визначаються економічні показники [2], характеризується набором з n параметрів (а не однією величиною, хоча у частинному випадку може бути і так). Поточному стану підприємства відповідає вектор $\vec{x}_i$ значень величин, що описують його економічний поточний i-тий стан (наприклад, стан рахунків, дебіторська та кредиторська заборгованість, величини поточних та ф'ючерсних контрактів тощо):

$$\vec{x}_i = \{x_{i1}, x_{i2}, ..., x_{in}\}. \qquad (1)$$

Інформація, яку надає (1) дозволяє професіоналу-економісту на основі знання типових значень відповідних величин x_{ij} оцінювати стан справ в цілому на підприємстві, а також у окремих його складових. Однак знання лише одного такого поточного значення вектора $\vec{x}_i$ сильно звужує можливості аналізу часових змін стану підприємства, а знання динаміки процесу у багатьох випадках навіть важливіше, ніж наявність інформації про поточний стан об'єкта.

Відповідно до теми роботи, аналізу та контролю підлягає набір величин – обсяги попиту на асортимент товарів супермаркету.

Існує багато способів [4] моделювання стану $\vec{x}_i$: статистичні, кваліметричні, на основі методів штучного інтелекту із застосуванням нейронних мереж.

3. Застосування нейронних мереж для вирішення задачі прогнозування

Нейронні мережі застосовуються для задач розпізнавання [3], класифікації [4], апроксимації і прогнозування.

Штучною нейронною мережею (ШНМ) називають систему, яка, приймаючи на вхід вектор вхідних величин, за складними (нелінійними) внутрішніми законами створює вектор вихідних величин, що є розв'язком поставленої задачі [5].

Елементарною складовою ШНМ є один нейрон (рис. 2). Це об'єкт, що має певну кількість входів (на рисунку – n входів), які називаються дендритами та один вихід, що називається аксоном [6].

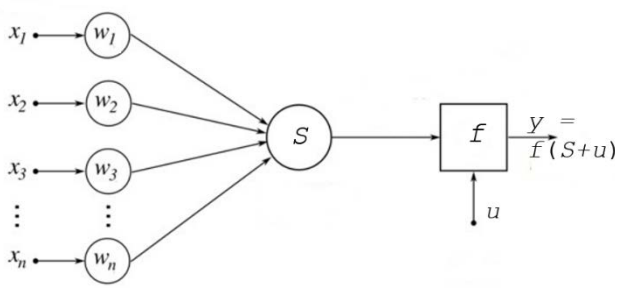


Рис. 2. Структура штучного нейрону

На входи нейрона подаються вхідні сигнали x_1, x_2, …, x_n, що є виходами попередніх нейронів, розташованих на попередніх шарах мережі. Кожен вхідний сигнал x_i множиться на певне число w_i, що називають синаптичною вагою відповідного входу. Усі зважені вхідні сигнали подаються на суматор, який виробляє зважену суму S виду:

$$S = \sum_{i=1}^{n} w_i x_i \,. \qquad (2)$$

До цієї зваженої суми вхідних сигналів може додаватися зсув або зміщення u (у поширеному частинному випадку зміщення відсутнє, тобто $u = 0$), і від цієї суми виробляється певна функція f, що називається активаційною. Так формується вихідний сигнал нейрона y, що передається далі по нейронній мережі:

$$y = f(S + u). \qquad (3)$$

Відповідно, комбінуючи (2) та (3) маємо функцію, що здійснює штучний нейрон [7]:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + u \right). \qquad (4)$$

При аналізі такого способу задання функції, що здійснює нейрон, як залежність (4), бачимо, що важливу роль для кожного нейрона відіграють три особливості:

– набір конкретних значень ваг w_i, що описують кожен вхід нейрона;

– вид активаційної функції f даного нейрона;

– наявність зсуву або порогу активації u (який може бути як додатним, так і від'ємним).

Зважаючи на те, що звичайна ШНМ складається з десятків або сотень нейронів та шарів, а в кожного з нейронів є багато входів, то задача вибору вагових коефіцієнтів w_i значно ускладняється [8]. Тривалий процес задання конкретних «правильних» значень цих вагових коефіцієнтів називають процесом навчання нейронної мережі. Існують різні алгоритми навчання нейронних мереж, які укрупнено можна поділити на дві групи: «з учителем» та «без учителя».

Вид активаційної функції f також впливає на функціонування нейронів, хоча й значно меншою мірою, ніж вибір синаптичних вагових коефіцієнтів [9]. На рис. 3 наведено найбільш поширені види функцій активації, що найчастіше використовуються на практиці при роботі з нейронними мережами. З точки зору обчислювальної складності кращим є використання порогових функцій, що фактично являють собою константну залежність (або хоча б лінійних, але не сигмоїдних).

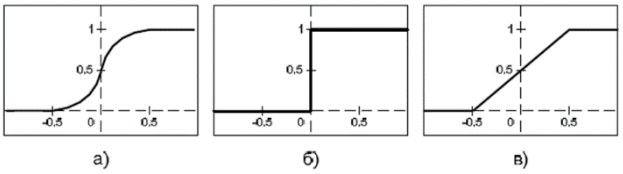


Рис. 3. Найбільш поширені типи функцій активації: а – сигмоїдна, б – порогова, в – лінійна.

При побудові нейронної мережі, крім характеристик окремих нейронів, важливим стає також спосіб їх з'єднання (так само, як і їх загальна кількість, організація нейронів у групи, шари тощо).

Найпростішим, і в той же час найпоширенішим способом з'єднання нейронів у конкретну ШНМ є персептрон, що обумовлено широким спектром задач, які можна вирішувати за допомогою ШНМ з такою внутрішньою структурою.

Враховуючи особливості ШНМ, нейронні мережі є гарним інструментом для вирішення задачі прогнозування [10]. Тому

використання нейронної мережі обрано за основу для програмної реалізації.

4. Постановка задачі прогнозування попиту на асортимент товарів супермаркету

Необхідно розробити підсистему прогнозування попиту на асортимент товарів супермаркету на основі використання штучних нейронних мереж. Для цього слід виконати наступні кроки:

- проаналізувати та запропонувати перелік показників роботи супермаркету, що можуть впливати на величину попиту і направити їх на входи нейронної мережі;
- обґрунтувати тип та структуру нейронної мережі для прогнозування обсягів попиту;
- виконати програмну реалізацію цієї підсистеми;
- знайти у відкритих джерелах відповідні відомості та сформувати пул навчальної інформації, на основі якої здійснити навчання нейронної мережі;
- протестувати роботу підсистеми та зробити висновки щодо ефективності впливу на кінцевий результат окремих економічних показників.

У якості вхідної інформації використовується:

- перелік назв та економічна ефективність показників роботи супермаркету, що можуть впливати на обсяги попиту на асортимент його товарів (загальна кількість видів товарів супермаркету, кількість кас, кількість висотних будинків, наявних поруч із супермаркетом; загальні заплановані витрати на закупку асортименту товарів, площа торгової зали, відстань до найближчого транспортного вузла, відстань до іншого найближчого супермаркету, номер місяця року, затрати на рекламу, освітленість у торговельній залі тощо);
- перелік значень обраних показників та відповідного значення попиту майбутнього періоду, що становить навчальну вибірку для навчання нейронної мережі;
- характеристики доступних засобів розробки програмного забезпечення, що дозволяють обрати оптимальний варіант для реалізації інтелектуальної системи на базі нейронної мережі.

Вихідною інформацією є величини попиту на окремі товари супермаркету у майбутній період**.**

Для розв’язання задачі прогнозування з використанням нейронних мереж, необхідно застосувати алгоритм навчання ШНМ.

5. Алгоритм навчання ШНМ

Для реалізації алгоритму навчання необхідно спочатку зчитати дані для навчання. У файлі розміщуються параметри, що впливають на попит, а також реально отримані значення попиту на обрані товари за заданий період. Кількість файлів, що використовуються для навчання, дорівнює числу магазинів, і кожен файл вміщує дані з одного магазину.

На рис. 4 схема процесу навчання показана узагальнено, хоча більшість її елементів мають порівняно просту деталізацію. Найскладнішим є реалізація елементу «Відкоригувати ваги синапсів мережі за допомогою методу зворотного поширення помилки», який здійснюється за допомогою однойменного методу.

Також, на рис. 4 не показані прості елементи на зразок вибору максимального значення із набору похибок. Самі ж похибки у випадку векторного виходу нейронної мережі можуть обчислюватися за різними процедурами, однак у запропонованій моделі використовується проста метрика *n*-вимірного евклідового простору, тобто корінь квадратний із суми квадратів різниць кожної прогнозованої величини попиту та реально отриманого значення (функція втрат).

Тренування нейронної мережі полягає у мінімізації функції втрат. Для вирішення задачі оптимізації використовується алгоритм стохастичного градієнтного спуску, за допомогою якого можна відкоригувати ваги синапсів мережі.

Алгоритм стохастичного градієнтного спуску:

1. Вибрати один параметр з набору даних.
2. Знайти всі частинні похідні функції втрат за вагою або за зміщенням.
3. Оновити усі ваги та зміщення.
4. Повторити з п. 1.

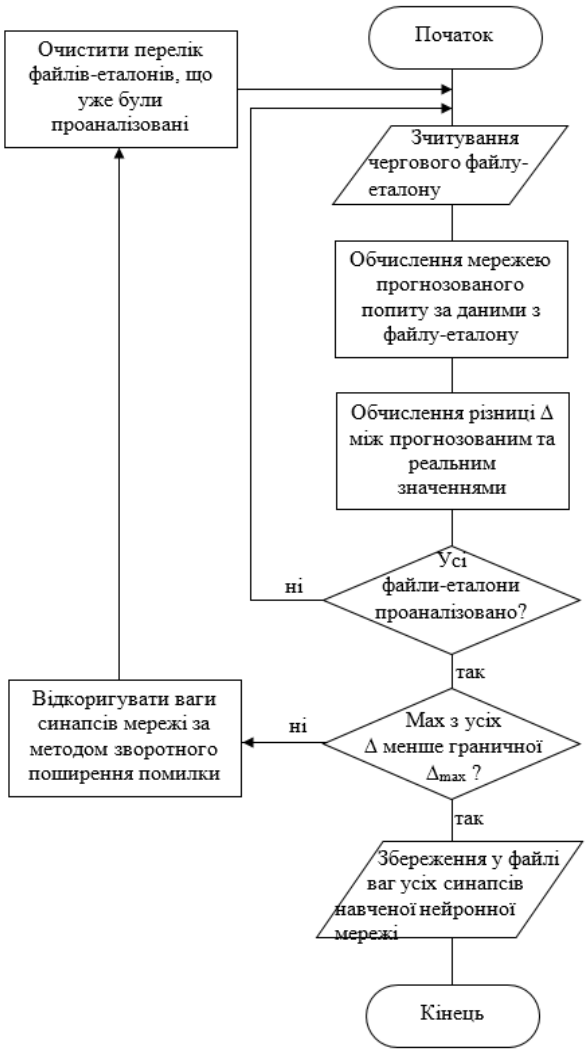


Рис. 4. Блок-схема алгоритму процесу навчання нейронної мережі

6. Тестування та аналіз результатів прогнозування

Проведено тривале у часі тестування стабільності роботи створеної програми, яке показало, що вона працює у штатному режимі, без виникнення якихось системних помилок, непередбачених аварійних завершень тощо.

Розроблена програма протестована на адекватність виконання поставленої задачі – прогнозування попиту на асортимент товарів супермаркету. Для цього випадковим чином із наявної сукупності усіх навчальних ситуацій (яких всього зібрано 72 – по два роки для трьох різних магазинів) виключено 3, на яких здійснювалося не навчання, а перевірка адекватності прогнозування навченої нейронної мережі. Ці 3 випадки, що виключені із навчальної вибірки, використані для тестування адекватності результатів роботи мережі.

Отже, після навчання системи, у програмі тричі запускався процес прогнозування, у якості вхідних даних використовувалися реальні параметри. В результаті було отримано прогнозовані величини попиту на п’ять груп товарів, що наведені в табл. 1-5. Також, можна побачити справжні обсяги споживання, що були отримані реальними супермаркетами при таких вхідних даних, і відповідні похибки прогнозування.

Табл. 1. Похибки прогнозування попиту на хліб для трьох реальних випадків (навчальних ситуацій)

Випадок:	Значення, отримане реальним магазином, грн.:	Прогнозоване значення, грн.:	Похибка прогнозу-вання:
Ситуація 1	6720	7718	13%
Ситуація 2	8117	8792	8%
Ситуація 3	5294	6213	15%
Середнє	-	-	**12%**

Табл. 2. Похибки прогнозування попиту на молоко для трьох реальних випадків (навчальних ситуацій)

Випадок:	Значення, отримане реальним магазином, грн.:	Прогнозоване значення, грн.:	Похибка прогнозу-вання:
Ситуація 1	10500	9871	6%
Ситуація 2	9873	12110	18%
Ситуація 3	14215	15974	11%
Середнє	-	-	**12%**

Табл. 3. Похибки прогнозування попиту на ковбасу для трьох реальних випадків (навчальних ситуацій)

Випадок:	Значення отримане реальним магазином, грн.:	Прогнозоване значення, грн.:	Похибка прогнозу-вання:
Ситуація 1	34860	39802	12%
Ситуація 2	45940	52114	12%
Ситуація 3	31780	26974	18%
Середнє	-	-	**14%**

Табл. 4. Похибки прогнозування попиту на горілку для трьох реальних випадків (навчальних ситуацій)

Випадок:	Значення отримане реальним магазином, грн.:	Прогнозоване значення, грн.:	Похибка прогнозу-вання:
Ситуація 1	43260	52105	17%
Ситуація 2	57211	52114	10%
Ситуація 3	40024	47974	17%
Середнє	-	-	**15%**

Табл. 5. Похибки прогнозування попиту на картоплю для трьох реальних випадків (навчальних ситуацій)

Випадок:	Значення отримане реальним магазином, грн.:	Прогнозоване значення, грн.:	Похибка прогнозу-вання:
Ситуація 1	6327	7412	15%
Ситуація 2	7282	8659	16%
Ситуація 3	6143	7586	19%
Середнє	-	-	**17%**

Середні значення похибок складають від 12 % до 17 %, що є цілком задовільним результатом проведених тестувань, на основі якого можна будувати стратегії розвитку та діяльності бізнесу на майбутні періоди. Розроблений програмний продукт має достатню практичну цінність та може застосовуватися у якості підсистеми прогнозування споживчого попиту на асортимент товарів супермаркетів України.

Висновки

У роботі обґрунтовано застосування штучних нейронних мереж для розв'язання задачі прогнозування, сформульовано постановку задачі, розроблено алгоритм навчання мережі та створено програмну реалізацію прогнозуючої мережі. В результаті тестування програми прогнозування встановлено, що середня похибка прогнозування становила 12-17 %, що є задовільним результатом, тому створену модель можна вважати адекватною. Розроблена програма може бути використана у якості підсистеми прогнозування споживчого попиту на асортимент товарів супермаркету.

Список літератури

1. Назгальдов Г.Г. Разработка теоретических основ квалиметрии: автореф. дис. … д-ра эконом. наук. – М., 1981.
2. Льюис К.Д. Методы прогнозирования экономических показателей. – М.: «Финансы и статистика», 1986. – 133 с.
3. Egmont-Petersen, M.; de Ridder, D.; Handels, H. (2002). Image processing with neural networks – a review. Pattern Recognition 35 (10): 2279–2301.
4. Duda, Richard O.; Hart, Peter Elliot; Stork, David G. (2001). Pattern classification (вид. 2). Wiley.
5. Bhadeshia H. K. D. H. (1999). Neural Networks in Materials Science. ISIJ International 39 (10): 966–979.
6. Haykin, Simon S. (1999). Neural networks : a comprehensive foundation. Prentice Hall.
7. Hertz, J.; Palmer, Richard G.; Krogh, Anders S. (1991). Introduction to the theory of neural computation. Addison-Wesley.
8. Lawrence, Jeanette (1994). Introduction to neural networks : design, theory and applications. California Scientific Software.
9. Cybenko, G.V. (2006). Approximation by Superpositions of a Sigmoidal function. J van Schuppen, Jan H. Mathematics of Control, Signals, and Systems[en]. Springer International. с. 303–314.
10. M., Bishop, Christopher (1995). Neural networks for pattern recognition. Clarendon Press.

УДК 004.891.3

ДМИТРУК О.Ю.

ВИКОРИСТАННЯ ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ З ПІДБОРУ ОДЯГУ

У даній статті розглянуті наявні застосунки доповненої реальності, що використовуються для надання можливості віртуальної примірки одягу та аксесуарів в інтернет-магазинах. Описано ідеї проектів віртуальних примірочних. Викладено принципи роботи рекомендаційних систем для підбору товарів.

ДОПОВНЕНА РЕАЛЬНІСТЬ, ВІРТУАЛЬНА ПРИМІРОЧНА, РЕКОМЕНДАЦІЙНА СИСТЕМА

The subject of the article is usage of the augmented reality applications for enabling virtual fitting of clothing and accessories in online stores, ideas of virtual fitting rooms projects and description of the main principles of work of recommendation systems.

AUGMENTED REALITY, VIRTUAL FITTING ROOM, RECOMMENDATION SYSTEM

1. Вступ

Упродовж останніх років технології доповненої реальності, засоби штучного інтелектута рекомендаційні системистали дедалі частіше використовуватися в індустрії продажу одягу та аксесуарів у мережі Інтернет. Найбільш широко нейронні мережі застосовуються для засобів онлайн-примірки предметів гардеробу.

2. Проекти з використанням доповненої реальності та штучного інтелекту для примірки одягу

Ідея використання комп’ютерних технологій для підбору одягу не є новою. У 2012 році українська компанія NICE Interactive представила віртуальну примірочну NiceFit, розроблену на основі технологій доповненої реальності з використанням сенсору Kinect. Сенсор Kinect дозволив користувачам за допомогою жестів переміщатись по меню, змінювати параметри зросту та об’єму та обирати для примірки товар з каталогів магазинів, підключених до системи. Застосунок пов’язувався з хмарним сховищем, з якого завантажувались попередньо підготовлені 3D-моделі одягу, створені за фотографіями. Програма враховувала реалістичну поведінку тканини при поворотах та нахилах людини[1].

Однією з найпопулярніших розробок створених для онлайн-примірки є FittingBox. Даний програмний продукт використовується інтернет-магазинами з продажу окулярів. Застосунок використовує сканер Studio Box для оцифрування окулярів з фото. Після цього моделюється та візуалізується зображення. Дане зображення накладається на завантажене фото або на відео з веб-камери в режимі реального часу. Для коректної примірки,нейронна мережа виконує розпізнавання обличчя та визначає, куди повинні накладатись окуляри. Детальніше про розробку можна прочитати на офіційному сайті [2].

Центр розробки програмного забезпечення EDISONy 2019 році поширив опис проекту віртуальної примірочної, головною перевагою якого є визначення більшості параметрів людини необхідних для підбору одягу за фотографією[3]. На вхід подається фото тіла людини в облягаючому однотонному одязі на контрастному фоні. Наявність інших людей чи предметів в кадрі є недопустимою.Контур тіла вираховується за допомогою бібліотеки OpenCV, після чого контур проходить валідацію: людина повинна стояти прямо з розташованим симетрично корпусом,руки мають бути розведені на 45° долонями до середини. За вигонами ліній отриманого контуру виділяються вузлові точки, між якими потім обраховується відстань у пікселях по горизонталі та вертикалі. Для коректування базових векторних моделей фігур чоловіка та жінки обраховується поправка вузлових точок за осями координат. На виході формується масив координат вузлових точок

векторної моделі. Двовимірна модель відображується перед користувачем у вигляді однотонного напівпрозорого заповненого контуру, накладеного на вхідну фотографію. На контурі моделі зображуються вузлові точки, які користувач може переміщувати в допустимих заданих межах. Переміщення вузлової точки призводить до дзеркального переміщення симетричної точки відносно вертикальної осі. Користувач може вказати один параметр або більше: зріст, обхват талії і т.д. Інші параметри розраховуються автоматично з урахуванням пропорцій частин тіла і анатомічних особливостей будови тіла людини.При примірці людина обирає річ з каталогу. Новий елемент одягу накладається на призначену для нього частину тіла. Автоматично розраховується розмір, враховується поєднання речей між собою.

3. Проект з використанням штучного інтелекту для пошуку речей та їхнього комбінування

Російський стартап Get Out fit спільно з білоруським проектом штучного інтелекту Oyperзапустили сервіс, який за допомогою нейронних мереж дозволяє підбирати образи та знаходить конкретні моделі предметів одягу. Щоб знайти певну річ, достатньо завантажити її фотографію і система надасть посилання на магазини, де можна придбати подібну. Розробники також навчили нейронну мережу підбирати поєднання верхніх та нижніх елементів одягу і формувати образи. В подальшому алгоритм зможе пропонувати більш складні образи, які включають в себе взуття та аксесуари[4][5].

4. Використання рекомендаційних систем для підбору одягу

Рекомендаційні системи на даний момент є невід'ємною складовою онлайн-маркетингу. За допомогою явних та неявних засобів вони збирають інформацію про користувачів для подальшого підбору товарів, що можуть їм сподобатися. Прикладом явного збору інформації про користувача є список вподобаних товарів, присутніх на більшості інтернет-платформ з продажу одягу. Неявним засобом збору інформації є відстеження поведінки користувача, наприклад, спостереження за тим, які товари він переглядає.

Виділяють два типи рекомендаційних систем: на основі фільтрації вмісту та колаборативної фільтрації.

Рекомендаційні системи на основі фільтрації вмісту пропонують користувачеві придбатитовари, подібні на переглянуті або придбані. Наприклад, якщо покупець переглядає білі кросівки певного розміру, йому пропонуються інші моделі білого кольору, для яких є в наявності відповідний розмір. Проте, корисність таких рекомендацій зазвичай обмежена для сфери продажу одягу, оскільки придбавши певний предмет гардеробу, покупець з малою ймовірністю захоче купити дуже подібний. Окрім цього недоліку, негативним фактором є сильна залежність від предметної області, тобто, рекомендація обмеженого кола товарів.

Підхід колоборативної фільтрації у рекомендаційних системах є більш універсальним, оскільки враховує історії покупок та переглядів як одного користувача, так і вибірки подібних до нього. Основою даного підходу є припущення про те, що схожим користувачам сподобаються схожі товари. Для формування вибірки схожих покупців, для кожної пари користувачів обчислюється міра їхньої подібності. Найчастіше за дану міру приймають коефіцієнт Жаккара[8], що визначається як міра спільної частини поділена на міру об'єднання множин:

$$J(u_i, u_j) = \frac{|u_i \bigcap u_j|}{|u_i \bigcup u_j|},$$

де u_i -множина товарів, що переглідав користувач i ; u_j - множина товарів, що переглядав користувач j.

Основною проблемою даного підходу є «холодний старт» - явище, коли новий товар нікому не рекомендуються. Ще одним недоліком є проблема створення рекомендацій для нових або нетипових користувачів.

Для підвищення точності передбачень рекомендаційних систем використовується машинне навчання. Основною метою є виділення параметрів об'єктів, спираючись на які модель передбачатиме вподобання користувачів з найменшим значенням квадрату помилки.

Стандартною метрикою передбачення оцінки є середньоквадратична похибка моделі (Root Mean Square Error, RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}},$$

де y_i - передбачення оцінки користувача інформаційною системою; $\hat{y}_i$ - реальна оцінка товару користувачем [9].

Недоліком такої оцінки є те, що похибка в передбачені високої оцінки товару користувачем має таку саму вагу, як похибка в передбачені низької оцінки. Для оцінки похибки використовуються також метрики ранжування, що опираються на множину рекомендованих товарів та множину товарів, що реально сподобалися користувачу. Вони теж не є достатньо точними, оскільки не враховують інформацію про товари, які користувач не придбав або не додав до списку обраних.

Основною проблемою будь-яких рекомендаційних систем є те, що через деякий проміжок часу вони починають пропонувати користувачеві однотипні товари.

Висновок

Зараз активно розвиваються проекти із застосуванням штучного інтелекту та технологій доповненої реальності для віртуальної примірки одягу та пошуку потрібних речей та поєднань у інтернет магазинах. Проте, дані програмні продукти на даний момент широко використовуються лише у сфері продажу окулярів. Більшість інтернет-магазинів використовує рекомендаційні системи для підбору товарів для користувачів. Недоліком таких систем є прихід до однотипності через деякий проміжок часу.

Задля покращення якості обслуговування покупців у інтернет-магазинах з продажу одягу, можна об’єднати технології для віртуальної примірки предметів гардеробу з рекомендаційними системами на основі колаборативної фільтрації. Більшість програмних продуктів для віртуальної примірки формують масиви вузлових точок обличчя та/або тіла користувача. Дані масиви точок та інформація про приміряні та вподобані товари може бути надана рекомендаційній системі. Таким чином, кластеризація користувачів буде проводитись не лише за множинами їхніх вподобань, але й з врахуванням типів їхньої фігури та обличчя, що допоможе підвищити ймовірність корисності рекомендацій. Кластеризація з урахуванням типу фігури, а не лише смаку, дозволить урізноманітнити набір рекомендованих товарів та запобігти швидкій появі однотипних рекомендацій, оскільки люди з однаковими параметрами можуть мати різні вподобання.

Список літератури

1. Nice Fit: первая в Украине «виртуальная примерочная» [Електронний ресурс] – Режим доступу до ресурсу: https://gagadget.com/other/8965-nice-fit-pervaya-v-ukraine-virtualnaya-primerochnaya/.
2. FittingBox Technology [Електронний ресурс] – Режим доступу до ресурсу: https://www.fittingbox.com/en/fittingbox-technology.
3. Виртуальная примерочная EDISON [Електронний ресурс] – Режим доступу до ресурсу: https://www.edsd.ru/virtualnaya-primerochnaya.
4. Get Outfit [Електронний ресурс] – Режим доступу до ресурсу: https://www.getoutfit.ru/oyper.
5. Искусственный интеллект для гардероба: когда виртуальный стилист начнет покупать одежду за вас [Електронний ресурс] – Режим доступу до ресурсу: https://vc.ru/future/54342-iskusstvennyy-intellekt-dlya-garderoba-kogda-virtualnyy-stilist-nachnet-pokupat-odezhdu-za-vas.
6. Рекомендаційна система - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0.
7. Пишем простую систему рекомендаций на примере Хабра [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/post/230155/#theory
8. Как работают рекомендательные системы. Лекция в Яндексе [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/company/yandex/blog/241455/.
9. Среднеквадратическая ошибка модели [Електронний ресурс] – Режим доступу до ресурсу: http://statistica.ru/glossary/general/srednekvadraticheskaya-oshibka/.

УДК 004.912

ЯСЕНОВА А.В.
ХАЛУС О. А.

ЗАСТОСУВАННЯ АЛГОРИТМІВ КЛАСТЕРИЗАЦІЇ НА РИНКУ ІНОЗЕМНИХ ВАЛЮТ

В даній статті проаналізовано застосування алгоритмів кластеризації для визначення груп активів з однаковою поведінкою на ринку іноземних валют. Описано процес вибору оптимальних гіперпараметрів, представлені результати роботи алгоритмів, описано переваги та недоліки обраних методів.
КЛЮЧОВІ СЛОВА: кластеризація, форекс, ринок іноземних валют, оптимальний торговий портфель
In this article, the application of clustering algorithms to determine groups of assets with the same behaviour in the foreign exchange market is analyzed. An overview of chosen methods for optimal hyperparameters selection and the results of algorithms work are presented, advantages and disadvantages of the chosen methods are described.
KEYWORDS: clustering, forex, foreign exchange market, optimal trading portfolio

1. Вступ

Сьогодні форекс відкриває можливості для торгівлі необмеженою кількістю валют і металів на біржах по всьому світу. Одним із головних правил трейдингу є диверсифікація ризиків, його суть полягає в тому, що трейдер не повинен торгувати лише однією валютною парою (індексом, акцією і тд), натомість він має обрати декілька активів тим самим сформувати своє портфоліо інструментів.

Торгівля на ринку іноземних валют є доволі складним процесом і навіть найбільш досвідчений трейдер може втратити частину капіталу, якщо помиляється в своїх припущеннях. Головною метою формування портфелю активів є отримання такого їх набору, що є менш ризикованим, ніж будь-який з цих активів поодинці. Диверсифікація ж розуміє під собою не відсутність збитків, а зменшення їх загального впливу на баланс трейдера, тобто ймовірність втрати при торгівлі одним інструментом існує, проте збиток буде не такий значний, якщо за допомогою інших позицій вдасться заробити.

Для досягнення такого результату всі елементи портфелю повинні мати різну природу і поведінку, наприклад, до складу набору варто включити валютні пари, цінні метали, індекси і акції. Зазвичай вибір активів виконується з огляду на певні події в світі, власний досвід чи вподобання.

2. Постановка задачі

Розглянемозадачу знаходження таких груп активів, що всередині групи поведінка активів схожа між собою, а за межами групи істотно відрізняється.

Нехай $\boldsymbol{X}$ – це множина всіх доступних для аналізу активів, також існує множина кластерів $\boldsymbol{Y}$ та функція відстані $\rho(x, x')$ (якавизначається алгоритмом). На вхід алгоритму подається вибірка об’єктів $\boldsymbol{X_m} = \{x_1, \ldots, x_m\} \subset \boldsymbol{X}$. Задача полягає в тому, щоб розбити задану вибірку на такі непересічні підмножини, щоб кожен кластер містив об’єкти, що близькі за вибраною метрикою, а об’єкти різних кластерів істотно відрізнялися. При чому кожен об’єкт має бути віднесений лише до одного кластера.

3. Опис використаних алгоритмівта отриманих результатів

Кластерний аналіз — задача розбиття заданої вибіркиоб'єктів (ситуацій) на підмножини, які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися [1]. Отже алгоритми кластеризації можна використовувати для того, щоб виділити ті самі групи активів з різною поведінкою в минулому, і враховуючи отримані результати сформувати портфель для торгівлі.

Для дослідження використані відкриті дані про зміни цін 140 валютних пар за 2

роки. Експерименти проводилися з алгоритмами KMeans та DBSCAN.

3.1.KMeans

Алгоритм KMeans – один з найпростіших алгоритмів машинного навчання без учителя, що забезпечує впорядкування множини об'єктів в порівняно однорідні групи [2]. В результаті його роботи кожна валюта буде віднесена до кластера, середнє значення якого є для неї найближчим.

Кількість кластерів - це вхідний параметр алгоритму, він визначений за допомогою ліктьового методу (Рисунок 1) [3].

Для кожної валюти обчислені два параметри: перфоманс і волатильність. Вони дають розуміння того, наскільки сильно в середньому змінювалася ціна активу й інтенсивність зміни ціни. Результат роботи алгоритму представлений на рис. 2.

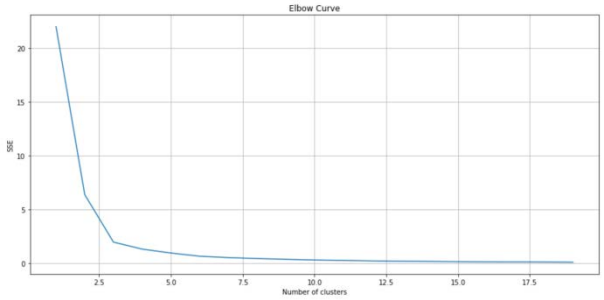


Рис. 1. Застосування ліктьового методу для вибору кількості кластерів методу к-середніх

```
Cluster  1 =  ['KHR', 'NAD', 'SZL', 'TND']
Cluster  2 =  ['AED', 'AMD', 'ANG', 'AUD', 'AWG', 'AZN', 'BAM', 'BBD', 'BDT', 'BGN', 'BHD', 'BMD', 'BOB', 'BSD', 'B
TN', 'BWP', 'BYR', 'BZD', 'CAD', 'CDF', 'CHF', 'CLF', 'CLP', 'CNY', 'COP', 'CRC', 'CUC', 'CZK', 'DJF', 'DKK', 'DZD'
, 'EGP', 'ERN', 'EUR', 'FJD', 'FKP', 'GBP', 'GGP', 'GIP', 'GTQ', 'GYD', 'HKD', 'HNL', 'HRK', 'HUF', 'IDR', 'ILS', '
IMP', 'INR', 'ISK', 'JEP', 'JMD', 'JOD', 'JPY', 'KES', 'KGS', 'KPW', 'KRW', 'KWD', 'KYD', 'KZT', 'LKR', 'LVL', 'LYD
', 'MAD', 'MDL', 'MKD', 'MMK', 'MNT', 'MOP', 'MUR', 'MVR', 'MXN', 'MYR', 'MZN', 'NIO', 'NOK', 'NPR', 'NZD', 'OMR',
'PAB', 'PEN', 'PHP', 'PLN', 'PYG', 'QAR', 'RON', 'RSD', 'RUB', 'SAR', 'SEK', 'SGD', 'SOS', 'SRD', 'STD', 'SVC', 'SY
P', 'THB', 'TJS', 'TTD', 'TWD', 'TZS', 'UAH', 'UGX', 'USD', 'UYU', 'VND', 'VUV', 'WST', 'XAF', 'XAG', 'XAU', 'XCD',
'XDR', 'XPF', 'YER', 'ZMK', 'ZWL']
Cluster  3 =  ['CVE', 'GNF', 'IRR', 'MGA', 'SDG']
Cluster  4 =  ['BTC', 'ETB', 'GMD', 'IQD', 'LAK', 'LBP', 'MRO', 'MWK', 'RWF', 'SHP', 'UZS']
Cluster  5 =  ['AFN', 'AOA', 'ARS', 'BND', 'BRL', 'BYN', 'HTG', 'NGN', 'PKR', 'SCR', 'TOP', 'TRY', 'ZAR', 'ZMW']
```

Рис 2. Результат виконання алгоритму к-середніх

До першого кластеру алгоритм відніс валюти нерозвинених країн, до четвертого дуже волатильні пари (наприклад, біткойн), а другий кластер - найбільший і це логічно, так як більшість валют мають нормальний розподіл дельти своїх цін (в теорії), тому в даному випадку алгоритм не здатен виділити “специфічні” групи.

3.2.DBSCAN

DBSCAN є алгоритмом кластеризації заснованим на щільності: для заданої множини точок у деякому просторі він відносить в одну групу точки, які розташовані найбільш щільно (точки з багатьма сусідами) та розмічає точки, які лежать в областях з невеликою щільністю (чиї сусіди розташовані занадто далеко) як викиди [4].

На вхід алгоритму подається матриця коваріації, побудована на дельтах цін досліджуваних валют. Для даного алгоритму також важливі значення вхідних параметрів. В нашому дослідженні вони підібрані за допомогою пошуку по сітці [5], якість кластеризації визначається величиною силуету (silhouette score) [6]. Результат роботи алгоритму наведений на рис. 3.

```
Cluster 1: AED, AMD, ANG, AWG, AZN, BBD, BDT, BGN, BHD, BMD, BOB, BSD, BTN, BWP, BYR, BZD, CAD, CDF, CHF, CNY, CRC,
CUC, DJF, DKK, DZD, EGP, ERN, FJD, GTQ, GYD, HKD, HNL, HRK, ILS, INR, JMD, JOD, JPY, KES, KPW, KRW, KWD, KYD, KZT,
LKR, LVL, LYD, MAD, MDL, MKD, MMK, MNT, MOP, MUR, MVR, MYR, MZN, NIO, NPR, OMR, PAB, PEN, PHP, PYG, QAR, RSD, SAR,
SGD, SOS, SRD, STD, SVC, SYP, THB, TJS, TTD, TWD, TZS, UAH, UGX, USD, UYU, VND, XAF, XCD, XDR, YER, ZMK, ZWL
Cluster 2: AUD, BRL, MXN, NZD, ZAR
Cluster 3: CLF, CLP, COP, RUB
Cluster 4: CZK, HUF, PLN
Cluster 5: FKP, GIP
Cluster 6: GBP, GGP, IMP, JEP
Cluster 7: NOK, SEK
```

Рис. 3. Результат виконання алгоритму DBSCAN

DBSCAN виділив доволі багато невеликих кластерів (по 2-5 валютних пар), а всі інші активи відніс до першого найбільшого кластеру. Використання даного методу є більш доцільним для вирішення поставленої задачі, так як важливо аналізувати наскільки схожими були зміни цін активів в минулому, що відображено в матриці коваріації.

Висновки

Отже, алгоритми кластеризації надають можливість розділити всю множину валютних пар на групи, елементи яких схожі між собою. Вони мають певні недоліки: KMeans дуже чутливий до викидів в даних, тому перед його використанням обов'язково необхідно перевірити їх цілісність (в нашому дослідженні така проблема виникла в зв'язку з тим, що для деяких валютних пар частина історичної інформації була відсутня, відповідно вони не використовувалися для дослідження, так як всі активи повинні порівнюватися за один і той же період часу), DBSCAN класифікує точки на межах кластерів як шум і не відносить їх до одного з кластерів, тому в теорії можлива ситуація, що алгоритм не виділить ніяких кластерів взагалі. Проте KMeans дозволяє виділити принципово різні групи валют, тобто вирішує поставлену задачу, хоч і без специфічних інсайтів, а завдяки можливості класифікувати частину множини точок як шум DBSCAN корисний для виявлення аномалій.

Список використаних джерел

1. Кластерний аналіз [Онлайн] Доступно з: https://uk.wikipedia.org/wiki/Кластерний_аналіз
2. Кластеризація методом к-середніх [Онлайн] Доступно з: https://uk.wikipedia.org/wiki/Кластеризація_методом_к–середніх
3. Elbow method [Онлайн] Доступно з: https://en.wikipedia.org/wiki/Elbow_method_(clustering)
4. DBSCAN[Онлайн] Доступно з: https://uk.wikipedia.org/wiki/DBSCAN
5. Grid search [Онлайн] Доступно з: https://en.wikipedia.org/wiki/Hyperparameter_optimization#Grid_search
6. Silhouette (clustering) [Онлайн] Доступно з: https://en.wikipedia.org/wiki/Silhouette_(clustering)

УДК 004.891.3

НОВАКІВСЬКА К. Д.

ДОСЛІДЖЕННЯ МЕТОДІВ ОБЧИСЛЕННЯ ПРОДУКТИВНОСТІ ПРАЦІВНИКІВ ІТ-КОМПАНІЙ

В даній статті проведено аналіз існуючих аналогів для визначення продуктивності праці та кластеризації працівників ІТ-компаній. Проведено порівняння багатокритеріальних методів кластеризації таких, як метод k-середніх, метод ієрархічної кластеризації та метод нечіткої кластеризації c-середніх. Для зазначених методів наведені переваги та недоліки.
ПРОДУКТИВНІСТЬ, КЛАСТЕРНИЙАНАЛІЗ

This article analyzes the existing analogues for determining the productivity and clustering of employees of IT companies. A comparison of multicriteria clustering methods such as the method of k-means, the method of hierarchical clustering and the method of fuzzy clustering of c-meansis presented. The advantages and disadvantages of the indicated methods are given.
PRODUCTIVITY, CLUSTERINGANALYSIS

1.Вступ

Тенденцією останнього десятиліття стало стрімке зростання кількості IT-спеціалістів, IT-компаній. За 5 років кількість IT-працівників в Україні подвоїлася і досягла 180 тисяч. Українських жінок в IT-сфері стає все більше – вони займають четверту частину працівників галузі.

Водночас, великі компанії часто займаються проектами, пов'язаними з різними галузями господарства, різного рівня складності, що вимагає створення робочих команд зі спеціалістів різних спеціалізацій.

Оцінка продуктивності працівника допомагає роботодавцям порівняти працівників та розділити їх на продуктивних, а саме тих, що вчасно виконують завдання та можуть виконувати завдання різних складностей, або непродуктивних, які не встигають виконувати поставлені задачі або виконують лише завдання найменшої складності.

Для того, щоб оцінити продуктивність роботи, необхідно проаналізувати такі фактори як середня складність виконаних завдань, кількість виконаних завдань за визначений проміжок часу, а також витрачені на роботу ресурси.

2.Аналіз існуючих аналогів для визначення продуктивності роботи

На сьогодні більшість працівників з метою збільшення продуктивності праці та організації робочого часу користуються спеціалізованими застосунками для організації своєї роботи або роботи своїх підлеглих. Прикладами таких застосунків є Yaware Time Tracker, Kickidler, Croco Time.

Yaware Time Tracker призначений для оцінки продуктивностіроботи за комп'ютером, визначити найкращого та найгіршого працівника, а також дозволяє оцінити реальне навантаження працівників. В основу роботи застосунку покладено ідею спостереження за відкритими вікнами на робочій машині (ПК, ноутбук, планшет та ін.). Недоліком даного методу є те, що вимірюванняпродуктивностіроботи з сайтами та застосунками є актуальним тільки для працівників, в обов'язки яких входить робота з сайтами та застосунками.[1][2]

Kickidler – це застосунок, який дозволяє проводити спостереження за відвіданими сайтами та застосунками працівника з метою забезпечення безпеки передачі даних у межах компанії. Застосунок здійснює блокування сайтів, відвідування яких не є необхідним для виконання роботи (наприклад Facebook, Viber чи інші соціальні мережі та месенджери). Аналіз ефективності здійснюється на основі таких фактів як час, проведений за роботою, та час відпочинку.[3]

CrocoTime використовує метод фотографії робочого часу, що ґрунтується на спостереженні за відкритими вікнами, а також надає можливість перегляду витрат часу і причин, що призвели до цих витрат. Прикладом у даному випадку може бути неефективно витрачений час.[4]

Метод фотографії робочого часу – метод, що дає можливість визначити розподіл часу конкретного працівника за допомогою спостережень, вимірювань та документування витрат часу на всі робочі операції протягом дня. Зазвичай цей метод застосовується для знаходження причин, через які не були досягнуті середні показники продуктивності роботи компанії в цілому. За допомогою цього методу можна визначити, скільки часу витрачається на виконання завдання працівником та за рахунок чого можна збільшити продуктивність, якщо це можна реалізувати.Метою методу є визначення найбільш затратних за часом робочих процесів, вивчення досвіду найкращих працівників, встановлення робочих норм, виявлення часових витрат, оцінка ефективності працівників, виявлення причин та наслідків невиконання норм.

Існує декілька видів фотографії робочого часу:

1. Індивідуальна – вимірюються характеристики для окремих осіб
2. Групова – виконується вимір для групи людей, поєднаних однією ціллю або завданням
3. Комплексна –визначає зв'язки між робочими процесами, в даному випадку проводять спостереження за робочою командою
4. Самовимірювання – проводиться людиною з власної ініціативи

Докладніше в [5].

Недоліком CrocoTime, YawareTimeTracker є те, що активність у застосунках не передбачає успішного виконання

поставлених керівником задач і тоді аналіз ефективності роботи за таким методом не є коректним. Оскільки у даних застосунках неможливо призначити працівнику задачу, то у застосунках не аналізується обсяг та складність виконаних завдань. Croco Timeta Yaware Time Tracker можуть використовуватися лише керівниками, а тому працівники не мають можливості побачити статистику. Наведені вище приклади застосунків є платними або мають безкоштовні версії, функціонал яких обмежений.

Слід зазначити, що загальнодоступного універсального програмного продукту для ефективного управління продуктивністю праці не існує. Можливо, причиною є комерційна таємниця та значна конкуренція на ринку.

3.Аналіз існуючих аналогів для кластеризації об'єктів

Для того, щоб визначити спеціалізацію працівника IT-компанії за оцінками знань технологій необхідно застосувати методи, які дозволяють поділити об'єкти на кластери. Кластерний аналіз – це сукупність статистичних операцій, що дозволяє збирати дані, що містять інформацію про вибірку об'єктів, а потім групує об'єкти у порівняно однорідні групи. До методів кластеризації відносять метод k-середніх, k-медіан, EM-алгоритм, метод нечіткої кластеризації c-середніх. До бібліотек, які кластеризують об'єкти, можна віднести ApacheMahout.

ApacheMahout – це бібліотека для машинного навчання, яка реалізує поширені методи машинного навчання, такі як рекомендації, кластеризація та класифікація. Даною бібліотекою користуються такі відомі компанії, як Adobe, Facebook, LinkedIn, Foursquare та Twitter. ApacheMahout має реалізацію таких методів навчання без вчителя, як метод k-середніх та метод ієрархічної кластеризації. Навчання без вчителя – це потужний інструмент для аналізу наборів даних та виявлення закономірностей.[6]

Метод k-середніх – це метод кластерного аналізу, метою якого є поділ nспостережуваних об'єктів на k кластерів.

Нехай маємо ряд спостережуваних об'єктів $x = (x^1, x^2, ..., x^n)$, де n – кількість об'єктів

Необхідно мінімізувати функцію $\sum_{i=1}^{k} \sum_{x \in S_i} (x - \mu_i)^2$, де k – кількість кластерів, S_i – отримані кластери, $i = \overline{1,k}$, μ_i – центроїд кластеру S_i

Ідея методу k-середніх полягає у тому, що на кожній ітерації відбувається перерахування центроїдів кожного кластеру, отриманого на попередній ітерації, потім вибірка розбивається на кластери знову відповідно до того, який з отриманих центрів знаходиться ближче за відповідними метриками.[7]

Переваги методу k-середніх:

- Зручний для кластеризації великої кількості об'єктів
- Простота реалізації та невеликі затрати часу на виконання

Недоліки методу k-середніх:

- Не гарантується досягнення глобального мінімуму для $\sum_{i=1}^{k} \sum_{x \in S_i} (x - \mu_i)^2$, а тільки одного з локальних
- Результат залежить від вибору початкових центрів кластерів, а їх найкращий вибір невідомий

Метод ієрархічної кластеризації – це алгоритми упорядкування даних, що полягають у побудові дерева вкладених кластерів. Алгоритми ієрархічної кластеризації поділяють на дві групи методів:

- Агломеративні: дерева створюються від листків до кореня, тобто від менших кластерів до більших
- Дивізійні: дерева створюються від кореня до листків, тобто від більших кластерів до менших (протилежні до агломеративних)

Переваги методу ієрархічної кластеризації:

- Легко будується
- Можливість виділення ознак подібностей, як на одному, так і на різних рівнях

Недоліки методу ієрархічної кластеризації:

- Важко розуміти структуру дерева, якщо велика глибина дерева
- Великі витрати часу

- Важкий у застосуванні через велику кількість взаємопов'язаних множин
- При невеликій глибині недостатньо інформації через недостатню кількість ознак

Докладніший опис методу можна знайти в [8].

Для того, щоб розв'язати таку проблему, як залежність від вибору початкових центроїдів кластеру, якщо їх найкращий вибір невідомий, можна застосувати метод нечіткої кластеризації c-середніх. Оскільки замість визначення класу, до якого належить даний об'єкт як у методі k-середніх, метод c-середніх визначає ймовірності належності до кожного кластеру. Також метод c-середніх є кращим для кластеризації великої кількості об'єктів в порівнянні з методом ієрархічної кластеризації.

Метод нечіткої кластеризації c-середніх – метод кластеризації, метою якого є поділ n спостережуваних об'єктів на k нечітких кластерів. Даний метод є покращеною версією методу k-середніх, при якому для кожного об'єкту множини, що розглядається, розраховується степінь належності об'єкту до кожного кластеру.[9]

Нехай маємо ряд спостережуваних об'єктів $x = (x^1, x^2, ..., x^n)$, де n – кількість об'єктів

Необхідно мінімізувати функцію $\sum_{j=1}^{k} \sum_{x \in C_j} u_{ij}^m (x_i - c_j)^2$, де k – кількість кластерів, C_j –центроїд кластеру (середнє значення всіх точок зважене по їх належності до класу) рахується за формулою(3), $j = \overline{1,k}$, u_{ij} – степінь належності поточного об'єкту x_j кластеру з центром C_j і розраховується за формулою (1), m – визначає наскільки нечітким може бути кластер (зазвичай приймають $m = 2$)

$$u_{ij} = \frac{\left[\frac{1}{d_{ij}}\right]^{\frac{1}{m-1}}}{\sum_{k=1}^{c}\left[\frac{1}{d_{ki}}\right]^{\frac{1}{m-1}}} \qquad (1)$$

, де d_{ij} – Евклідова відстаньі розраховується за формулою (2)

$$d_{ij} = \sqrt{(x_1^i - c_1^j)^2 + (x_2^i - c_2^j)^2 + ...} \qquad (2)$$

$$C_j = \frac{\sum_{x \in C_j} u_{ij}^m x_i}{\sum_{x \in C_j} u_{ij}^m} \qquad (3)$$

Переваги методу нечіткої кластеризації c-середніх:

- слабка залежність від початкових центроїдів
- невеликі витрати часу на виконання

Отже, для того, щоб поділити працівників за спеціалізаціями краще застосовувати метод нечіткої кластеризації c-середніх, що потребує менше ресурсу часу для отримання результату та є більш точним, оскільки не має сильної залежності від початкових центрів кластерів.

Висновок

Сьогодні є популярною практика використання спеціалізованих застосунків для покращення процесів управління працівниками. В ході дослідження було проведено аналіз найбільш використовуваних аналогів для визначення продуктивності праці та кластеризації працівників. Були порівняні методи багатокритеріальної кластеризації об'єктів такі, як метод k-середніх, метод ієрархічної кластеризації та метод нечіткої кластеризації c-середніхі визначено, що найкращим для кластеризації працівників є метод нечіткої кластеризації c-середніх.

Швидше за все різні компанії самостійно створюють комплекс, де кожна з частин виконує окремі функції управління. Тому раціональним рішенням буде створення однієї системи, яка буде виконувати функції управління продуктивністю, підбору учасників проекту та поділу працівників за спеціалізаціями.

Список літератури

1. YawareTimeTracker: можливості [Електронний ресурс]– Режим доступу до ресурсу: https://timetracker.yaware.com.ua/what-is-yaware/

2. Як працюєYawareTimeTracker[Електронний ресурс]– Режим доступу до ресурсу: https://help.yaware.com/ru/knowledge-base/kak-rabotaet-yaware-timetracker-klient/
3. Переваги Kickidler[Електронний ресурс]– Режим доступу до ресурсу: https://www.kickidler.com/ru/about/about-kickidler.html
4. Crocotime: Як працює програма по контролю роботи персоналу?[Електронний ресурс]– Режим доступу до ресурсу: https://crocotime.com/ru/kak-rabotaet-programma-po-kontrolyu-raboty-personala/
5. О. С. Полякова, эксперт, Статьяопубликована в журнале «Планово-экономическийотдел» № 8, 2016. [Електронний ресурс]– Режим доступу до ресурсу: https://www.profiz.ru/peo/8_2016/effektivnost_raboty/
6. ApacheMahout: Algorithms[Електронний ресурс]– Режим доступу до ресурсу: http://mahout.apache.org/
7. Gorban A.N., Zinovyev A.Y. (2009). PrincipalGraphsandManifolds, Ch. 2 in: HandbookofResearchonMachineLearningApplicationsandTrends: Algorithms, Methods, andTechniques, EmilioSoriaOlivasetal. (eds), IGI Global, Hershey, PA, USA, pp. 28-59
8. Жамбю М. Иерархический кластер-анализ и соответствия. — М.: Финансы и статистика, 1988. — 345 с.
9. Bezdek, James C. PatternRecognitionwithFuzzyObjectiveFunctionAlgorithms. — 1981. — ISBN 0-306-40671-3

УДК 004.9

КАЗМІРЧУК А.В.
СПЕРКАЧ М.О.

ІНФОРМАЦІЙНА СИСТЕМА НАРАХУВАННЯ БОНУСІВ СПІВРОБІТНИКАМ КОМПАНІЇ

У даній роботі розглянуто системунарахування бонусів співробітникам компанії для підвищення ефективності їх роботи. В якості алгоритмічного забезпечення запропоновано застосування моделей та методів теорії розкладів. Для розв’язання багатокритеріальної задачі оптимізації, мінімізації загального часу завершення виконання робіт паралельними виконавцями, що мають різну швидкість роботи та мінімізації загальної кількості бонусів, було обрано багатокритеріальний генетичний алгоритм. Сформульовано змістовну та математичну постановку задачі. Проведено дослідження властивостей задачі. Розроблено алгоритм розв’язання поставленої багатокритеріальної задачі. Проведено ряд досліджень ефективності алгоритму.

КЛЮЧОВІ СЛОВА: БОНУСНА СИСТЕМА, УПРАВЛІННЯ ПРОЕКТАМИ, БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ, ФРОНТ ПАРЕТО, ГЕНЕТИЧНИЙ АЛГОРИТМ, РАНГ, РОЗКЛАД.

This article describes the employee benefits system for increasing their efficiency of work. Models and methods of schedule problem are proposed for developing an algorithm’s engine of the system. Multi-objective genetic algorithm is chosen for solving multi-objective optimization problem, makespan minimization for parallel performersthat have different speeds and general bonus amount minimization.Informative and mathematical formulation of the problem are defined. A study of the properties of the problem is made. An algorithm for solving the multi-objective problem is developed. A number of studies of the efficiency of the algorithm are conducted.

KEYWORDS: BONUSSYSTEM, PROJECT MANAGEMENT,MULTI-OBJECTIVE OPTIMIZATION, FRONTPARETO, GENETIC ALGORITHM, RANG, SCHEDULE.

1. Вступ

Менеджментперсоналу компанії, є важливою складовою успішного ведення бізнесу.

Мотивація та стимулювання персоналу дозволяють керівнику успішно керувати співробітниками та проектами, які ті виконують. Керівник організації розробляє свою індивідуальну схему мотивації співробітників [1].

Дуже важливо підібрати ефективну систему мотивації, що спонукатиме до підвищення продуктивності співробітників і призведе до отримання намічених результатів. Однією ізіснуючих систем мотивування є матеріальна [2].

Відомо, що матеріальна винагорода виступає рушійною силою для мотивування працівників виконувати свої задачі якісно та прагнути до постійного покращення своїх навичок, підвищення професійної кваліфікації. Саме тому більшість компаній активно впроваджують бонусні системи для своїх працівників. Зокрема, найпоширенішою така практика є у відділах продажу та службах технічної підтримки. Однак, такі системи часто виявляються неефективними, оскільки вартують занадто високо, але при цьому очікувані показники продуктивності не досягаються.

Отже, існує необхідність у створенні системи, яка б підтримувала процес ефективного планування та призначення бонусів співробітникам компанії, з метою, підвищення продуктивності роботи співробітників та зменшення витрат компанії.

2. Система нарахування бонусів співробітників

Розроблена система призначена для підтримкипроцесу нарахування бонусів співробітникам компанії як складової мотиваційної програми. Система зберігає дані про працівників та задачі, які необхідно виконати. Ці дані можуть імпортуватися із відповіднихсистем управління проектами, таких як Jira, наприклад, або ж із будь-яких внутрішніх корпоративнихCRM-систем. На основі цих даних відбувається розподіл задач між працівниками та нарахування кожному працівнику відповідної кількість бонусів. Завдяки застосуванню методів теорії розкладів вдається отримати оптимальні розклади виконання завдань за декількома критеріями – мінімізація бонусних витрат та мінімізація загального часу завершення завдань. Керівник має можливість обрати найкращий варіант розподілу завдань в команді із множини отриманих оптимальних рішень.

3. Змістова постановка задачі

Необхідно знайти такий розподіл завдань між працівниками, щоб із мінімальними витратами на бонуси за якісне виконання завдань досягти високої ефективності роботи.

4. Математична постановка задачі

Нехай задано множину робіт $J = \{1, 2, ..., j, ..., n\}$ і множину виконавців $M = \{1, 2, ..., i, ..., m\}$. Кожна робота може виконуватися будь-яким виконавцем. Передбачається, що виконавці мають різну швидкість виконання робіт v_i, $v_i > 0$, $i = \overline{1,m}$.

Для j-ї роботи задано час виконання p_j та кількість бонусів $b_j > 0$, $j = \overline{1,n}$. Якщо $v_i > 0 -$ швидкість i-го виконавця, то тривалість виконання j-ї роботи i-м виконавцем становить:

$$p_{ij} = \frac{p_j}{v_i},\ i = \overline{1,m},\ j = \overline{1,n}.$$

Кількість бонусів нарахованих за j-ту роботу залежить від часу її виконання i-м виконавцем і обчислюється за формулою:

$$b_{ij} = \frac{b_j}{v_i},\ i = \overline{1,m}\ ,\ j = \overline{1,n}.$$

Загальну суму нарахованих бонусів отримуємо з:

$$B = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij}\ ,$$

де $a_{ij} \in \{0,1\}$, $a_{ij} = 1$, якщо j-та робота призначена i-му виконавцю.

Нехай C_{ij} – момент завершення j-ї роботи i-м виконавцем. Тоді $C_{ij} = C_{ij-1} + p_{ij}$, $C_{i0} = p_{i0}$, $i = \overline{1,m}$, $j = \overline{1,n}$.

Загальний час виконання робіт визначається наступним чином:

$$C_{\max} = \max_{ij}\{C_{ij}\} = \max_{i}\{C_{in_i}\}.$$

Необхідно скласти розклад виконання робіт, при якому загальний час завершення робіт з множини J буде мінімальним $C_{\max} \to \min$ та загальна кількість нарахованих бонусів B буде мінімальною $B \to \min$.

5. Багатокритеріальна задача оптимізації

Для розв'язання багатокритеріальної задачі оптимізації мінімізації загального часу завершення виконання робіт паралельними виконавцями [3,4], що мають різну швидкість роботи та мінімізації загальної кількості бонусів було обрано багатокритеріальний генетичний алгоритм.

Важливою перевагою генетичних алгоритмів є те, що вони з самого початку працюють з великою популяцією кандидатів на рішення, і, відповідно, оптимальні по Парето рішення генеруються вже в 1-му поколінні. Таким чином, не зважаючи на складність поставленої задачі, можна отримати достатньо повну множину оптимальних по Парето рішень уже після першої ітерації процедури генетичного алгоритму при відносно невеликих затратах часу на обчислення [5].

Модифікація генетичного алгоритму для вирішення багатокритеріальної задачі оптимізації відрізняється від звичайної схеми оператором селекції, визначення кращих особин популяції, обчисленням значення пристосованості особин та застосуванням підходу Парето [6].

Отже, для кожної особини популяції обчислюються окремо значення кожного з критеріїв: загальний час завершення виконання робіт в системі та загальна сума бонусів.

Після обчислення значень критеріїв застосовується підхід Парето. Для кожної особини популяції визначається ранг. Наприклад, розглянемо як визначаються рангидля двухкритеріальної задачі мінімізації (Рис.1). Точки A, B, Сіз області I є недомінуючими по обом критеріям, тобто кращими за критеріями загального часу завершення робіт та загальної суми бонусних витрат, ніж значення в інших точках із I та II областей. Аналогічно, точки D, F, Кіз області II є недомінуючими точкою Міз області III. Таким чином, точки A, B, C належать до рангу 1 і складають множину оптимальних рішень по Парето. Точки D, F, K належать до рангу 2. Точка M належить до рангу 3.

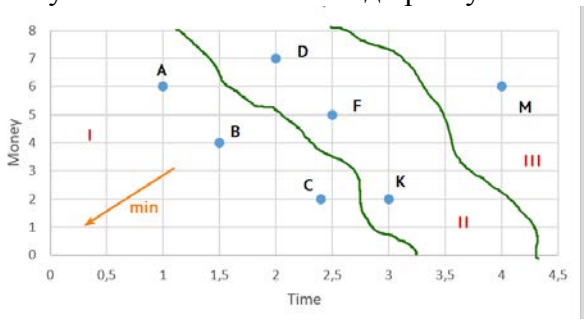


Рис.1. Ілюстрація визначення рангів

Також для всіх особин кожного рангу обчислюється відстань до сусідніх особин.

На Рис.2 наведено приклад визначення відстаней між особинами.

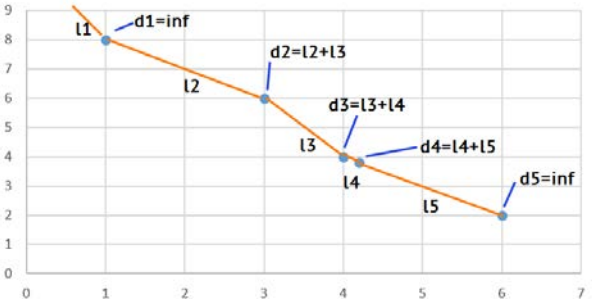


Рис.2. Відстані між особинами

Тоді для селекції особин поточної популяції та їх нащадків для створення нового покоління необхідно впорядкувати всіх особин по зростанню рангу і, після цього, по спаданню відстані до сусідніх особин. Таким чином, до нового покоління потрапляють перші N особин. Відстань до сусідніх особин одного рангу необхідно обчислювати для того, щоб уникнути збіжності до локального мінімуму та зберегти різноманітність осіб популяції. Так, наприклад, позначені на Рис.3 особини мають в 2 рази більше шансів бути обраними в якості батьків для створення нащадків, що може призвести до хибних результатів.

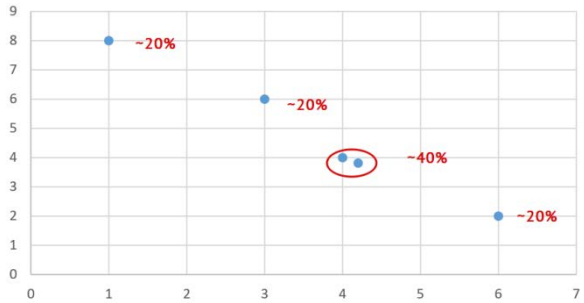


Рис.3. Ймовірність вибору особин для створення нащадків

Тому для наступного покоління обираються лише особини з великою відстанню до сусідніх особин.

6. Дослідження

Розроблений багатокритеріальний алгоритм було оптимізовано за допомогою введення паралельних обчислень в програмній реалізації генетичного алгоритму. Оскільки обчислення відстані до сусідніх особин в межах рангудля кожної особини популяції є незалежними операціями, то було застосовано розпаралелювання обчислень для даного кроку алгоритму.

Застосування паралельних обчислень в генетичному алгоритмі дозволило отримати підвищення швидкодії на 16% в порівнянні із його стандартною версією.

Висновок

У рамках даної роботи було розроблено систему розподілу бонусів для підвищення ефективності роботи працівників компанії. Наведено модель та метод для створення алгоритмічного забезпечення системи для оптимізації робочого процесу відповідно до сформульованих критеріїв. Описано алгоритм розв'язання поставленої багатокритеріальної задачі оптимізації та розроблено його оптимізовану версію. Проведені дослідження показали, що внаслідок оптимізації час роботи алгоритму було зменшено на 16%.

Список літератури

1. Устіловська А. С. «Мотивація персоналу як один з основних інструментів успішного управління персоналом» / А. С. Устіловська. // Молодийвчений. – 2017. – №4.
2. Чебан А. А. Ефективна система мотиваціїпраці як елемент підвищення конкурентноспроможності підприємства / А. А. Чебан., 2015. – (Молодийвчений).
3. German Y. Scheduling to Minimize Makespanon Identical Parallel Machines / YousefGerman., 2016. – (International Journal of Scientific&Engineering Research).
4. ZhenminCheng. Anapproximate algorithm for parallel machines cheduling problem to minimize total completion time. Computing&InformationSystems, 2010.
5. Багатокритеріальна оптимізація [Електронний ресурс] – Режим доступу до ресурсу: http://eos.ibi.spb.ru/umk/4_4/5/print/5_R1_T8.pdf.
6. Ashis K. M. Multi-ObjectiveGeneticAlgorithm / K. M. Ashis, K. M. Anil. // ResearchGate. – 2012.

УДК514.7

НОВОСЬОЛ К. І.,
ЛІЩУК К.І.

THE TECHNIQUE FOR STUDYING THE CHARACTERISTICS OF THE STOCHASTIC RELATIONSHIP CAN BE USED TO PREDICT THE FACTORS OF THE INFLUENCE OF METEOROLOGICAL CONDITIONS ON FLIGHT SAFETY AND AUTOMATE THE CONSTRUCTION OF CLIMATIC CHARACTERISTICS OF AERODROMES

У цій статті досліджено змішані центральні моменти висоти нижньої межі хмар і горизонтальної дальності видимості в районі аеродрому, що знаходиться в зоні помірно-континентального клімату. В якості вихідного статистичного матеріалу використані дані багаторічнихспостережень і побудована на їх основі кліматична характеристика аеродрому. Побудовано рівняння регресії горизонтальної дальності видимості на висоту нижньої межі хмар.

МЕТЕОРОЛОГІЧНІ ФАКТОРИ, НИЖНЯ МЕЖА ХМАР, ВИМІРЮВАННЯ, ГОРИЗОНТАЛЬНА ДАЛЬНІСТЬ ВИДИМОСТІ, БЕЗПЕКА ПОЛЬОТІВ

In this article, the mixed central moments of the height of the cloud base and the horizontal visibility range in the area of the airfield located in the zone of moderate continental climate are investigated. The data of long-term observations and the climatic characteristic of the aerodrome built on their basis were

used as the initial statistical material. The equations of regression of the HRV to the height of the oil-and-gas bearing are constructed.

METEOROLOGICAL FACTORS, CLOUDBASE, MEASUREMENT, HORIZONTAL RANGE OF VISIBILITY, SAFETY OF FLIGHTS

1. Introduction

Measurements of meteorological parameters at aerodromes are one of the most important elements of the meteorological support system for aircraft takeoff and landing and, ultimately, flight safety. In accordance with this, increased requirements are imposed on the volume, efficiency and reliability of meteorological information and on the technical means used[1]. The general trend in the development and implementation of aerodrome measuring and information systems at aerodromes is to expand the capabilities of the systems, the range of tasks to be solved, i.e. increasing the level of automation of meteorological support at the aerodrome. Automation makes it possible to simplify the solution of the problems of the climatic description of the aerodrome and increase the reliability and objectivity of the data, as well as the possibility of their visual presentation in the form of graphs, diagrams, etc. First of all, it is necessary to note the expansion of the possibilities of meteorological measurements.It is necessary to conduct regular studies of the variability of the height of the cloudbase (CB)[2], horizontal visibility range (HDV), weather conditions of varying degrees of complexity, dangerous weather phenomena, wind, etc. Taking into account the relatively slow change in meteorological elements, it was proposed to recalculate the climatic characteristics of aerodromes in five years.

To solve the problems of automating measurements, predicting changes in some meteorological elements based on the results of the analysis of others, it is necessary to develop software systems for processing statistical data, to analyze the characteristics of the stochastic relationship of the most significant meteorological elements. This work is devoted to the study of these issues.

2. Statistical characteristics of the most important meteorological elements of the aerodrome

One of the most important meteorological elements of the airfield are the height of the cloud base and the horizontal visibility range. Based on the results of the analysis and forecast of their averaged values, recurrence over the time of year and day, it is possible to more accurately plan its capacity, the degree of adherence to the schedule and, ultimately, ensure the required level of flight safety.

The height of the cloudbaseis constantly changing, and this boundary itself is a wavy surface with an amplitude of oscillations up to 50-100 m / h. The course of changes in the value of HRV is very close to the course of changes in the cloudbase(see Fig. 1 and 2).

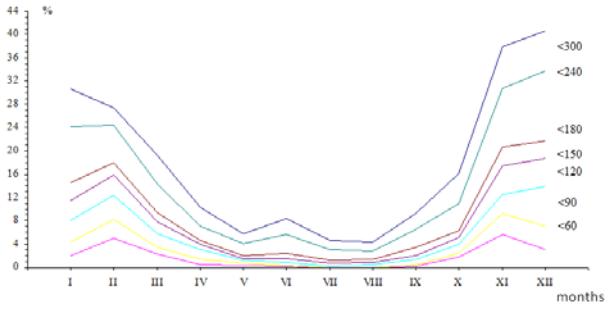


Fig 1 Changes in the height of the cloud border by months within the specified limits(as a percentage of the total number of measurements).

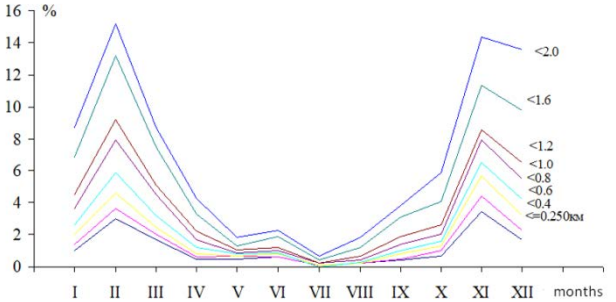


Fig 2 Changes in HRV by months within the specified limits(as a percentage of the total number of measurements).

Judging by the appearance of the graphs, the processes of changes in the CB and HRV, on the one hand, are essentially non-stationary, and, on the other, the tendencies

of their changes are very similar. Therefore, it is of interest to study the characteristics of their stochastic relationship[3]. This interest is not only theoretical but also practical. The HRV measurement procedure, in contrast to the measurements of the cloudbaseheight, is very laborious, and the measurement results have a very low accuracy. It is difficult to establish the theoretical dependence of HRV on weather conditions, since various atmospheric phenomena affect HRV in different ways, their contribution to the overall result is contradictory and difficult to formalize.

The coefficient of multiple correlation and multiple regression are used as the main characteristics of the stochastic relationship. Based on the physical essence of the problem, it is logical to choose the HRV regression equation as the dependent variable, and the height of the cloudbaseas the independent variable (argument). First of all, it is necessary to calculate the sample correlation coefficients. If they are much less than one, then the forecast of the HRV value based on the results of measuring the height of the cloudbaseusing the regression equation will be unreliable. In addition, to automate measurements and calculations, it is necessary to choose a method for approximating the curves of the repeatability of the height of the cloudbaseand HRV. The most flexible and accurate method is the polynomial approximation in terms of the minimum mean square of the error.

The most flexible and accurate method is the polynomial approximation in terms of the minimum mean square of the error . In fig. Figures 3, 4 show graphs of experimentally obtained dependences and their approximation by polynomials of the 6th and 10th order.

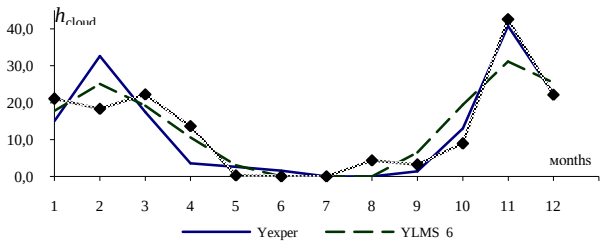


Fig 3 Approximation of the experimental curve for the recurrence of the height of the cloudbase<60 m by polynomials of the 6th and 10th degrees.

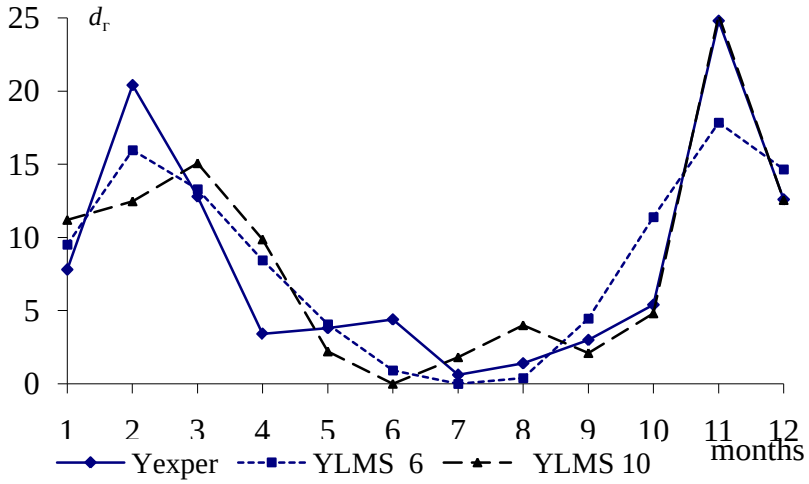


Fig 4 Approximation of the experimental repeatability curve of HRV<250 m by 6th and 10th degree polynomials

In the course of the calculations, it was found that the approximation by even order polynomials gives a higher accuracy than the approximation by the odd order polynomials close to the corresponding even order. In general, the order of the polynomial, at which an acceptable accuracy is achieved, strongly depends on the smoothness of the curve being approximated (compare Figs. 3 and 4). Therefore, in addition to calculating the approximating functions, it is necessary to perform statistical processing of the measurement results.

The sample correlation coefficient for pairs of values r_{hd} and N of the

cloudbaseand HRV heights can be estimated as follows:

$$r_{hd} = \frac{\sum_{i=1}^{N} h_{li} d_{vi} - N \bar{h}_l \bar{d}_v}{\left[\left(\sum_{i=1}^{N} (h_{li})^2 - N(\bar{h}_l)^2\right)\left(\sum_{i=1}^{N} (d_{vi})^2 - N(\bar{d}_v)^2\right)\right]^{1/2}} \quad (1)$$

.

Where h_{li} are the sample values of the cloudbase; d_{vi} - sampled valuesof HRV; $\bar{h}_l, \bar{d}_v$ - Selected average heights of the cloudbaseand HRV, respectively.

Using formula (1), the correlation coefficients of pairs of samples of heights of the cloudbaseand HRV were calculated from the minimum to the maximum values. Table 1 shows data on the correspondence of values, and in Table 2 - some results of calculations.

Table 1 Data selection on the correspondence of values

Cloudbase height, m	< 60	< 90	< 120	<150	< 180	< 240	< 300
HRV, m	<250	<400	<600	<800	<1000	<1200	<2000

Table 2 Correlation coefficient calculation results

Selected data	CB< 60 - HRV<250	CB<150 - HRV<800	CB< 300 - HRV<2000
Correlation coefficient r_{hv}	≈0.86	≈0.93	≈0.93

When calculating by the direct method and with preliminary data smoothing, as a result, a strong correlation of the studied samples is found, therefore, one should expect a high forecast accuracy using the regression analysis method.

3. Building regression models and calculating regression coefficients

The work investigated the linear regression of a random variable HRV (function) on a random value of the height of the cloudbase(argument):

$$d_v = A + Bh_l. \quad (0.2)$$

where A is the free term, B is the coefficient (the tangent of the slope of the regression line).

If the value of the correlation coefficient r_{hd} is equal to one, in essence, there is a deterministic functional relationship between h_l and d_v. In this case, the forecast error d_v of the measured value h_l will be zero.

In practice, there will always be $r_{hd} < 1$ so some scatter of points on the graph $d_v = f(h_l)$. Fig. 5 shows a set of pairs of values (*X*,*Y*) and a regression line Y on X. Deviations of the measured values from the predicted ones

$$d_{vi} - \hat{d}_{vi} = d_{vi} - (A + bh_l). \quad (3)$$

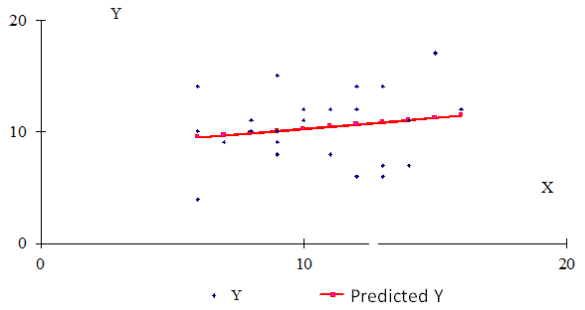


Fig 5 Regression Y on X. (X, Y) - a set of random numbers, distributed according to Poisson's law, with $\lambda = 10$ zero cross-correlation.

To calculate the coefficients A and B with a minimum error, the least squares method is also usually used - the sum of the squares of the deviations is minimized

$$\delta_\Sigma^2 = \sum_{i=1}^{N} (d_{vi} - A - Bh_l)^2. \quad (4)$$

The minimum is attained for such values A and B of the coefficients and that satisfy the condition

$$\frac{\partial \delta_\Sigma^2}{\partial A} = \frac{\partial \delta_\Sigma^2}{\partial B} = 0. \quad (5)$$

For samples of limited size, condition (5) allows one to obtain only estimates of the coefficients and, which we denote by and, respectively. Substituting expression (4) in (5) and solving equation (5) by the method of undefined coefficients, equating the coefficients at the same degrees, we obtain the corresponding equations for the estimates:

$$\left.\begin{aligned} a &= \bar{d}_v - b\bar{h}_l, \\ b &= \frac{\sum_{i=1}^{N} h_{li} d_{vi} - N\bar{h}_l\,\bar{d}_v}{\sum_{i=1}^{N} h_l^2 - N\left(\bar{h}_l\right)^2} \end{aligned}\right\}. \qquad (6)$$

Using the equations for estimates (6), it is possible to finally obtain an expression for the predicted values d_v from the measured h_l ones:

$$\hat{d}_v = a + bh_l = \left(\bar{d}_v - b\bar{h}_l\right) + bh_l = \bar{d}_v + b\left(h_l - \bar{h}_l\right) \qquad (7)$$

which is essentially a regression equation for d_v on h_l.

In fig. 6,7 show the graphs of the regression lines d_v on h_l for two pairs of characteristic values of the heights of the cloudbaseand HRV, and in table. 3 shows the calculated values A and B of the estimates of the coefficients and. Table 4 shows the values of the root-mean-square errors (RMS) for the estimation of the coefficients.

The standard deviation of the coefficient B estimate is relatively small, so it can be expected that the accuracy of predicting the slope of the regression line is very high. As for the standard deviation of the coefficient A estimate, it is much higher than for B. However, the coefficient of variation of the value d_v is much less than one, therefore, estimation errors A practically do not affect the resulting forecast accuracy.

The hypothesis of the normality of deviations was also tested and confirmed. The -statistics were evaluated and the confidence intervals calculated for the 5% significance level. The values of these intervals are much less than the absolute values of the coefficient estimates. Consequently, the probability of a gross error such as jumping from one curve to another in forecasting is a second-order value, and such an event itself can be classified as rare.

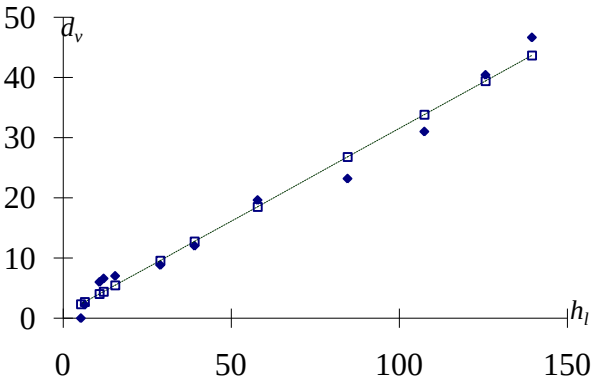


Fig 6 Regression line on. cloudbase<150 - HRV<800, non-decreasing series.

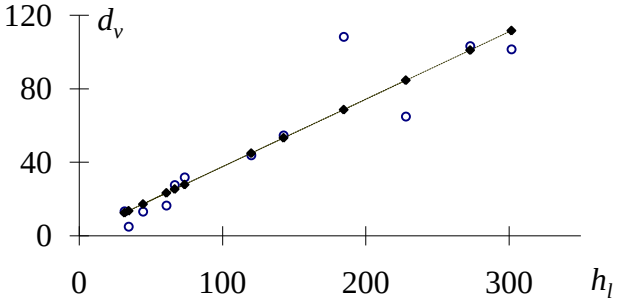


Fig 7 Regression line on. cloudbase<300 - HRV<2000.

Table 3 The calculated values A and B of the estimates of the coefficients

Coefficients of the equation	CB<150 - HRV<800	CB<150 - HRV<800 non-decreasing	CB< 300 - HRV<2000	CB< 300 - HRV<2000 non-decreasing
A	1.67	0.72	0.93	-2.17
B	0.29	0.31	0.37	0.39

Table 4 the values of the root-mean-square errors (RMS) for the estimation of the coefficients

RMSD	CB<150 - HRV<800	CB<150 - HRV<800 non-decreasing	CB< 300 - HRV<2000	CB< 300 - HRV<2000 non-decreasing
a	2.54	0.96	7.46	3.14
b	0.035	0.014	0.047	0.02

Conclusion

The climatic characteristics of the aerodrome is one of the most important components of flight safety, rhythm of work and adherence to the schedule (or planned flight table). The variability of the main meteorological elements is small. Therefore, the recalculation of the climatic characteristics can be carried out at sufficiently long intervals and, what is even more important, the results of long-term measurements can be used to obtain highly accurate estimates of climatic characteristics over a long period.

Statistical analysis of aerodrome climate data has shown that there is a strong correlation between cloud base height and horizontal visibility. Therefore, it is possible to predict the change in one parameter from the measured data for another parameter. This technique can also be used "in the opposite direction" - to predict the height of an oil-gas bearing from the results of measuring the HRV, if, for example, the latter have high accuracy and can be obtained with less labor costs.

The developed methodology for estimation and forecasting can be used to create hardware-software measuring and computing systems for meteorological support of flight safety.

References

1. Persin SM Modern trends in the development of aerodrome meteorological information and measurement systems. // Meteospectrum. – 2008. – C. 140 – 143.
2. Shakina N. P. Requirements for compiling a climatic description of the aerodrome / N. P. Shakina, L. A. Nudelman. – Moscow, 2007. – 35 c.
3. Mirsky G. Y. Characteristics of stochastic relationships and their measurements / Y. Mirsky. // Energoizdat. – 1982. – C. 320.

УДК004.02

ШИШМАН Ю.М.
ЖДАНОВА О.Г.

СИСТЕМА З ПІДТРИМКИ ПРОЦЕСУ ВЕРИФІКАЦІЇ ПРОГРАМНОГО ПРОДУКТУ

Робота присвячена розгляду системи з пітримки процесу верифікації програмного продукту, метою якої єавтоматизація тестування - генерація автотестів. Детально розглянута підсистема розробки планів тестування, що виконуєпобудову планів тестування програмного забезпечення на основі діаграми станів продукту, що тестується. Функція генерації планів була зведена до розв'язання задачі знаходження покриття множини згідно обранихкритеріїв якості множини тестів. Для цієї функції були розроблені та реалізовані метаевристичні алгоритми, наведені приклади результатів роботи даної підсистеми.

КЛЮЧОВІ СЛОВА: АВТОМАТИЗАЦІЯ, ТЕСТУВАННЯ ПРОГРАМНИХ ПРОДУКТІВ, ПОБУДОВА ПЛАНІВ ТЕСТУВАННЯ, ЗАДАЧА ПОКРИТТЯ МНОЖИНИ, МЕТАЕВРИСТИЧНІ АЛГОРИТМИ, АЛГОРИТМ МУРАШИНИХ КОЛОНІЙ, ГЕНЕТИЧНИЙ АЛГОРИТМ

The work is devoted to the consideration of the software product testing support system, the purpose of which is automation testing - the generation of autotests. The subsystem of development of testing plans which carries out construction of testing plans of the software on the basis of the diagram of states of the tested product is considered in detail. The function of generating plans was reduced to the task of finding the coverage of the set according to the selected quality criterion of the set of tests. Accurate and metaheuristic algorithms were developed and implemented for this function, the results of this subsystem are given.

KEYWORDS: AUTOMATION, TESTING OF SOFTWARE PRODUCTS, CONSTRUCTION OF TESTING PLANS, COVER SET, METAHEURISTICALGORITHMS, ANT COLONY OPTIMIZATION, GENETIC ALGORITHM

1. Вступ

Розробка якісного програмного продукту - це складний процес, що вимагає високого рівня підготовки команди розробників. Потрібно велику увагу приділяти тестуванню задля підвищення якості та роботоспроможності програмних продуктів. Ручне тестування в деяких випадках займає більше часу для перевірки системи, в той час як автоматизоване тестування дозволяє значно знизити витрати компаній замовників, зекономити ресурси та час. Отже, виникає проблема визначення плану тестування для універсальної автоматизованої системи тестування Web-порталів.

Зазвичайна етапі проектування функціонування програмного продукту представляється у вигляді діаграми станів та переходів.Базуючись на цій діаграмі розробляється план тестування. В даній роботі розглядається підхід до тестування чорним ящиком програмної системи за допомогою відповідної діаграми станів та переходів.

2. Постановка задачі

На сьогоднішній день основними видами тестуванняє наступні [1].

Системне тестування (SystemTesting) – оцінка програмного забезпечення щодо проектування архітектури. Тестування ПЗ, що виконується на повній, інтегрованій системі, з метою перевірки відповідності системи початковим вимогам.

Інтеграційне тестування (Integration Testing) – оцінка програмного забезпечення щодо проектування підсистем. Тестування програмного забезпечення, при якому окремі програмні модулі об'єднуються і тестуються у групі.

Модульне тестування (Module Testing) – оцінка програмного забезпечення щодо деталізації проектування, дозволяє перевірити коректність окремих модулів початкового коду програми.

Юніт тестування (Unit Testing) – оцінка програмного забезпечення щодо виконання. Це тестування є тестуванням “найнижчого” рівня.

Приймальне тестування (acceptance testing) – формалізоване тестування, спрямоване на перевірку застосунку з точки зору кінцевого користувача / замовника і винесення рішення про те, чи приймає замовник роботу у виконавця чи ні.

В системі з підтримки процесу верифікаціх програмного продукту реалізованафункція приймального тестування.

Подальший опис підсистеми розробки планів тестування продемонстровано на прикладідіаграми станів та переходів для основної частини функціоналу інтернет-магазину - додавання товару у кошик та успішне його замовлення.

Поставимо у відповідність кожному стану діаграми вершину орієнтованого графу, а в якості ребер буде перехід від одного стану до іншого. Отримали орієнтований граф, де кожному автотесту у відповідність ставиться шлях, який починається у конкретній початковій вершині та завершується в одній з термінальних (рис. 1).

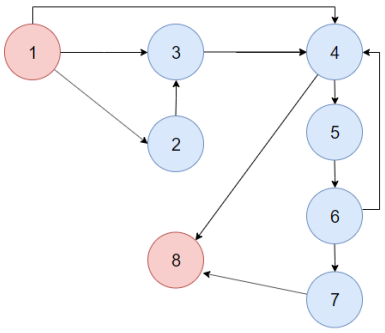


Рис. 1. Граф станів та переходів

Під планом тестування будемо розуміти сукупність автотестів, яка задовольняє певним умовам.В залежності від цих умов можливі декілька підходів до побудови плану тестування. Поширеними є покриття вершин (EdgeCoverage) та ребер (BranchCoverage), покриття шляхами (PathCoverage), яке має багато різних видів, одним з яких є Edge-PairCoverage (покриття типу вхід-вихід) [2].

Таблиця 1 відображає покриття типу Edge-PairCoverageдля графа зображеного на рис. 1, знаходження такої мінімальної множини шляхів, яка покриває усі комбінації пар дуг вхід-вихід для кожної вершини

графу. Наприклад, для вершини 4 множина пар дуг вхід-вихід така:{[(1;4)(4;5)],[(1;4)(4;8)], [(3;4)(4;8)],[(3;4)(4;5)],[(6;4)(4;5)], [(6;4)(4;8)]}.Надалі ці пари будемо називати тріадами [3].Після цього кожній тріаді ставиться у відповідність підмножина шляхів, що її покривають. У таблиці 1 представлені: множина усіх тріад, множина усіх шляхів від початкової до термінальних, а також ознака входу/не входу тріада у шлях.

Наступне важливе питання - визначення критеріїв якості множини тестів.Для тестера одним з важливих критеріїв є час виконання тесту тому були виділені такі стратегії: «довга», під якою розуміється знаходження плану тестування, який включає в себе найбільш довгі тести та «коротка», яка вимагає знаходження плану тестування, який б складався з більш коротких тестів.

Далі формулюється задача знаходження покриття, в основу якої покладені ці стратегії.

Табл.1. Входження тріад у шляхи

Шляхи	(1;2) (2;3)	(1;3) (3;4)	(1;4) (4;5)	(2;3) (3;4)	(1;4) (4;8)	(3;4) (4;5)	(3;4) (4;8)	(4;5) (5;6)	(5;6) (6;4)	(5;6) (6;7)	(6;4) (4;5)	(6;4) (4;8)	(6;7) (7;8)
[1,4,8]	0	0	0	0	1	0	0	0	0	0	0	0	0
[1,3,4,8]	0	1	0	0	0	0	1	0	0	0	0	0	0
[1,2,3,4,8]	1	0	0	1	0	0	1	0	0	0	0	0	0
[1,4,5,6,4,8]	0	0	1	0	0	0	0	1	1	0	0	1	0
[1,3,4,5,6,4,8]	0	1	0	0	0	1	0	1	1	0	0	1	0
[1,2,3,4,5,6,4,8]	1	0	0	1	0	1	0	1	1	0	0	1	0
[1,4,5,6,7,8]	0	0	1	0	0	0	0	1	0	1	0	0	1
[1,3,4,5,6,7,8]	0	1	0	0	0	1	0	1	0	1	0	0	1
[1,2,3,4,5,6,7,8]	1	0	0	1	0	0	0	1	0	1	0	0	1
[1,4,5,6,4,5,6,7, 8]	0	0	1	0	0	0	0	1	0	1	1	0	1
[1,3,4,5,6,4,5,6, 7,8]	0	1	0	0	0	1	0	1	0	1	1	0	1
[1,2,3,4,5,6,4,5, 6,7,8]	1	0	0	1	0	1	0	1	0	1	1	0	1

3. Математична постановка задачі

Задача знаходження покрить зводиться до класичної задачі мінімального покриття множини [3]. Початковими даними цієї задами є скінчена множинаT тріад і множина W шляхів (кожен з яких визначається множиною покритих тріад, тобто шлях можна розглядати як підмножину T). Покриттям$\overline{W} \subseteq W$ називають множину найменшої потужності (найменшої вартості) підмножин, об'єднанням яких є T.

Ця задача відноситься до класу NP-складних і може бути сформульована як задача булевого програмування. Нехай граф маєm тріад (множинаT) і в ньому можливі n шляхів від початкової до термінальних вершин (множина W). Для кожного шляху відомі тріади, що входять до нього, тобто для $i = \overline{1,m}$ та $i = \overline{1,n}$ задані

$$a_{ij} = \begin{cases} 1, \text{ якщо тріада } i \text{ входить у шлях } j, \\ 0, \text{ якщо тріада } i \text{ не входить у шлях } j. \end{cases}$$

Вимагається знайти таку підмножину шляхів$\overline{W} \subseteq W$, щоб кожна тріада входила хоча б у один шлях і ця множина була б найкращою згідно обраної стратегії. За умови, що

$$x_j = \begin{cases} 1, \text{ якщо } j-\text{й шлях входить до множ. } \overline{W}, \\ 0, \text{ якщо } j-\text{й шлях не входить до множ. } \overline{W}, \end{cases}$$

вказане обмеження формулюється наступним чином:

$$\sum_j a_{ij} x_j \geq 1 \, , i = \overline{1,m}$$

Щодо цільової функції. «Довга» стратегія призводить до мінімізації кількості шляхів, які будуть включені в план тестування, тобто

$$\sum_{j=1}^{n} x_j \to min$$

«Коротка» стратегія ж призводить до максимізації кількості шляхів. Для цього випадку цільова функція є такою:

$$\sum_{j=1}^{n} x_j \to max$$

4. Розв'язання задачі

В залежності від розмірності задачі та наявних обчислювальних ресурсів пропонується один з наступних методів:

– розв'язання задачі із застосуванням жадібного алгоритму;

– розв'язання задачі із застосуванням генетичного алгоритму;

– розв'язання задачі із застосуванням мурашинного алгоритму

4.1Розв'язання задачі із застосування жадібного алгоритму

Досить часто для реальних практичних NP-складних задач знайти точний розв'язок за прийнятний час не можливо. В такому випадку можна взяти ліпший розв'язок, який вдалося знайти за певний встановлений ліміт часу роботи точного алгоритму, або використовувати наближені або евристичні алгоритми. Жадібний алгоритм є одним з найпоширеніших зразків евристичних алгоритмів. Перевага цього алгоритму полягає в тому, що час знаходження рішення поліноміально зростає з ростом розмірності. Для задачі знаходження покриття графу шляхами розроблено та реалізовано наступний жадібний алгоритм. Спочатку маємо початкове покриття, яке не включає в себе жодного шляху. Доки це покриття не задовольняє умові усі комбінації пар вхід вихід додаємо до нього новий шлях, який покриває найбільш можливу кількість ще не покритих тріад. Цілком можливо, що отриманий розв'язок буде надлишковим. Вихідна система рівнянь будується на основі шляхів, що увійшли у знайдений розв'язок. Після оптимізації можливе отримання декількох розв'язків, що дуже схожі між собою.

З урахуванням сутності дій алгоритму побудовані розв'язки будуть в більшості випадків задовольняти вимогам довгої стратегії.

4.2Розв'язання задачі із застосування мурашинного алгоритму

Даний алгоритм відноситься до метаевристичних алгоритмів[4]. Для побудови покриття множини визначимо основні елементи мурашиного алгоритму.

Розміщення мурашок по шляхах

На початку першої ітерації створюється стільки мурашок, скільки шляхів має множина$\boldsymbol{W}$ ($\boldsymbol{k}$-та мурашка – на $\boldsymbol{k}$-ий шлях). Це робиться для того, щоб на початку роботи алгоритму кожен з шляхів попав хоча б в одне з покрить (що дозволить максимально розширити область пошуку). На наступних ітераціях створюється певна кількість мурашок (ця величина є параметром алгоритму), які випадково обирають один з шляхів множини $\boldsymbol{W}$.Це зменшення кількості мурашок (у порівнянні з першою ітерацією) обґрунтовується тим, що при великому значенні кількості мурах значно зростає час роботи алгоритму.

Локальні правила поведінки мурашок

Тепер з урахуванням особливостей задачі про мінімальне покриття множини, ми можемо
описати локальні правила поведінки мурашок при виборі шляху.

На відміну від класичного мурашиного алгоритму будемо вважати, що мурашки, при своєму руху «збирають» шляхи з множини W у свій кошик (до тих пір, поки збирана множина шляхів не стане повним покриттям).

Мурашки мають власну "пам'ять", оскільки кожен шлях можна пройти (взяти в кошик) тільки один раз, то у кожної мурашки є список вже пройдених шляхів - список заборон (W_k^+) – множина шляхів, яку вже набрала мурашка). Позначимо через W_k^- - список шляхів, які ще не відвідала мурашка k (в цю множину входять тільки ті шляхи, які містять не покриті тріади). Мурашки

володіють "зором". Видимість - евристичне бажання пройти шлях j.

Мурашки володіють "нюхом", вони можуть уловлювати слід феромона, підтверджуючий бажання пройти шлях $\boldsymbol{j}$ із множини W_k^- на підставі досвіду інших мурашок. Кількість феромона, що відповідає шляху $\boldsymbol{j}$ у момент часу $\boldsymbol{t}$ позначимо через τ_j .

Правило, що визначає вірогідність вибору наступного шляху (переходу)

Правило визначення величини, від якої прямо пропорційно буде залежати вірогідність вибору наступного шляху $\boldsymbol{k}$-ої мурашки залежить від обраної стратегії.

Нехай $\boldsymbol{k}$-та мурашка набрала шляхи:

$$\boldsymbol{W_k^+ = \{i_1, i_2, \dots\}}$$

(на початку першої ітерації $\boldsymbol{i_1 = k}$).

Ця множина однозначно визначає:

a) множину$\boldsymbol{T^+}$- множину покритих тріад.
b) $\boldsymbol{T^-}$- множина непокритих тріад.
c) $\boldsymbol{W_k^-}$- множина всіх шляхів, що відповідають непокритим тріадам.

Визначимо величину видимості шляху

$$\boldsymbol{j \in W_k^-}.$$

Коротка стратегія

Логічно, що видимість шляху $j \in W_k^-$ для короткої стратегії обернено пропорційна довжині l_j цього шляху:

$$\eta_j = \frac{1}{l_j}.$$

Довга стратегія

Очевидно, що видимість шляху$j \in W_k^-$для довгої стратегії прямо пропорційна довжині l_j цього шляху ($\eta_j \sim l_j$):

$$\eta_j = \frac{l_j}{l_{max}}$$

де l_{max} - довжина найдовшого шляху.

4.3.Розв'язання задачі із застосування генетичного алгоритму

Структура множини допустимих розв'язків задачі знаходження покриття графу шляхами (кожний розв'язок є булевим вектором розмірностіn) дозволяє розробити для цієї задачі генетичний алгоритм, що відноситься до метаевристичних алгоритмів [4].

Схема роботи генетичного алгоритму

Крок 1) Генеруємо початкову популяцію з n хромосом.

Крок 2) Обчислюємо для кожної хромосоми її придатність.

Крок 3) Вибираємо пару хромосом-батьків за допомогою одного із способів відбору.

Крок 4) Проводимо кросинговер двох батьків з імовірністю pc, виробляючи двох нащадків.

Крок 5) Проводимо мутацію нащадків з імовірністю pm.

Крок 6) Повторюємо кроки 3-5, поки не буде згенеровано нове покоління популяції, що містить n хромосом.

Крок 7) Повторюємо кроки 2-6, поки не буде досягнутий критерій закінчення процесу.

З урахуванням специфіки задачі були розроблені основні оператори алгоритму.

Формування початкової популяції

Параметрами алгоритму формування початкової популяції є:

- максимальний розмір популяції;
- максимальна кількість ітерацій.

Оператор вибору батьків

Реалізованооператор вибору батьків, в основу якого покладений «турнірний» спосіб відбору. В результаті якого в якості першого батька обирається найкраща на поточний момент особина, а в якості другого – випадкова.

Оператор схрещування

Реалізовано багаточковий кросинговер. При цьому при схрещуванні забезпечується допустимість та не надлишковість розв'язку. Перше досягається за рахунок розташування точок схрещування таким чином, що відповідні ділянки генетичного коду кожного з батьків, відповідають одній і тій самій множині тріад.

Оператор мутації

Випадковим чином виключається певний шлях, а замість нього випадковим чином додаються шляхи (доти, поки не буде отриманий допустимий розв'язок.

Оператор додавання особини в популяцію

Оператор реалізована за рахунок видалення найгіршої особини наприкінці кожної ітерації.

5. Програмна реалізація

В результаті даної роботи була розроблена підсистема для побудови планів тестування. Реалізована функція для генерації планів тестування, яка включає в себе:

- редактор для створення графу з можливістю додавання, видалення дуг та вершин графу;
- пошук та відображення усіх шляхів з початкової вершини у термінальні;
- пошук покриття за допомогою жадібного, генетичного та мурашинного алгоритмів;
- збереження результатів.

На рисунках 2 та 3 зображені результати роботи даної підсистеми, а саме генерація планів за допомогою жадібного та генетичного алгоритмів відповідно.

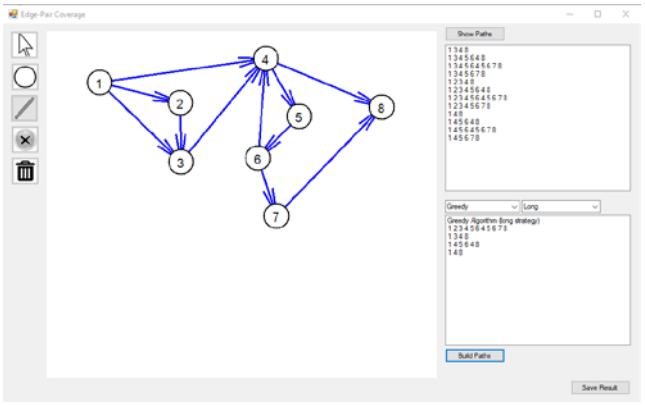


Рис. 2. Результат роботи жадібного алгоритму

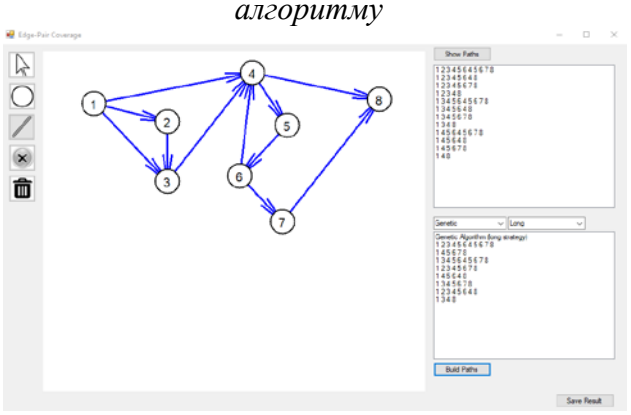


Рис. 3. Результат роботи генетичного алгоритму

Висновки

В рамках системи представлена функція побудови планів тестування програмного забезпечення на основі діаграми станів продукту, що тестується. В процесі роботи над системою було розглянутоосновні види тестування, задача покриття множини згідно обраної стратегії. Розроблені метаевристичні алгоритми розв’язання задачі. Створена програмна реалізація функції генерації планів, яка дозволить групі тестувальників значно зменшити час для планування тестування.

Список літератури

1. Куликов С. Тестирование программного обеспечения. Базовый курс, 2019. Версия 2.1.3. С. 84.
2. Ammann P., Offutt J. Introduction to software testing. Cambridge University press - New York. 2008, –322 p.
3. О. Г. Жданова, О. Б. Падалко. Про проблему генерації планів тестування для автоматизації процесу верифікації програмного продукту // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка : збірник наукових праць. – 2009. – № 50. – 34–41 с.
4. Сергиенко И.В., Шило В.П. Задачи дискретной оптимизации: проблемы, методы решения, исследования. – К. Наук. думка, 2003. – 264 с.

УДК 004.93

УМАНСЬКИЙ В.А.
ГАВРИЛЕНКО О.В.

ІНФОРМАЦІЙНА СИСТЕМА З ПІДТРИМКИ РОБОТИ КЕРІВНИКА МОБІЛЬНОЇ ГРУПИ З ПИТАНЬ ОХОРОНТ ТА БЕЗПЕКИ ТОРГІВЕЛЬНОЇ МЕРЕЖІ

У даній статті наведено математичну постановку задачі підтримки працівника мобільної групи з урахуванням часових вікон. Використано метаевристичні алгоритми для розв`язку поставленої задачі: повторюваний локальний пошук, алгоритм імітації відпалу, задача комівояжера. Наведено експеримент з результатів роботи пропонованих алгоритмів.
Ключові слова: ЗАДАЧА ОРІЄНТУВАННЯ, ЗАДАЧА КОМАНДНОГО СПОРТИВНОГО ОРІЄНТУВАННЯ З ЧАСОВИМИ ВІКНАМИ, МЕТАЕВРИСТИКА, ЛОКАЛЬНИЙ ПОШУК, ПОВТОРЮВАНИЙ ЛОКАЛЬНИЙ ПОШУК, АЛГОРИТМ ІМІТАЦІЙНОГО ВІДПАЛУ, ЗАДАЧА МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ, ЗАДАЧА КОМІВОЯЖЕРА

The mathematical model for team orienteering problem with time windows is given. Metaheuristics are used to solve this problem: repetitive local search, simulation annealing and the salesman problem The computational experiment was conducted to evaluate the work of the used algorithms.

Keywords: ORIENTEERING PROBLEM, TEAM ORIENTEERING PROBLEM WITH TIME WINDOWS, METAHEURISTIC, LOCAL SEARCH, ITERATED LOCAL SEARCH, SIMULATED ANNEALING ALGORITHM, VEHICLE ROUTING PROBLEM, SALESMAN PROBLEM

1. Вступ

Сьогодні існує безліч задач які можна віднести до класу Vehicle routing problem (VRP), велика кількість задач з різних галузей може бути змодельовано термінами даного класу, як наслідок усі ці задачі добре досліджені та адаптовані до цих областей, наприклад туризм, торгівля, будівництво, будь яка подібна задача з теорії розкладів. Але, існують задачі,які можуть бути сформульовані термінами задач VRP,а також з використанням алгоритмів Orienteering problem (OP), зазначимо що формулювання задачі у термінах OP дозволяє більш точно змоделювати специфіку предметної області кожної предметної області задачі. Саме тому, проводиться аналіз та вивчення даного класу задач, з’являються нові постановки задач і методи рішення.

Необхідність дослідження класу задач OP зростає з появою нових галузей, проблематика яких зводиться до постановки задачі OP. Як приклад можна навести зростання попиту на написання ПЗ для прогнозування розподілу персоналу (галузь HR), складання карти маршрутів і поїздок для туристів, людей, яким необхідно оптимально планувати свій час, працівникам мобільних груп, працівникам клінінг компаній, директорам, у яких щодня заплоновано чимало зустрічей у різних куточках міста а тд. Такі проблеми можна змоделювати термінами задач класу OP – Team orienteering problem with time windows (TOPTW). Для рішення побних задач також можна обрати формулювання задачі VRP, а також інші, більш склідні модифікації TOPTW. Але обрано TOPTW, тому що мат. поставнока задачі VRP не враховує специфіку проблем (наприклад, той факт, що не всі візитиобов`язкові),а модифіковані потановки TOPTW ускладнюють математичну постановку і методи розв’язку математичної задачі.

2. Математична постановка задачі

SLPTW – задача, що вирішеє потання максимізації сумарної корисності множини маршрутів з урахуванням часових рамок.

Точки, які потрібно відвідати, представимо вершинами орієнтовного повного зваженого графу. Граф містить n вершин, за умови що , $n \varepsilon N$. Необхідно

побудувати m маршрутів у графі за умови що , m ε N.

Маршрути обмежені у часі. Tmax – максимальна тривалість кожного з маршрутів. Кожен маршрут має свій початок у вершині і = 1, а кінець у вершині і = n. При відвідуванні кожної вершини корисність маршруту збільшується на певну величину Scorei.
Таким чином, постановка задачі побудови маршрутів для керіввників мобільних груп зводиться до розв'язання задачі комівояжера.

Збільшення корисності маршруту вібдувається тільки при першому відвідуванні вершини і належить (1,n). Потрапити у кожну вершину можна тільки у рамках її часового проміжку (вікна). Даний проміжок обмежується величинами: Oi – початок часового вікна, коли можна відвідати пункт, Ci – кінець часового вікна, коли можна відвідувати пункт і ε 1,n.

Потрібно зазначити, що відвідування певної вершини може починатись з часу Ci. Ti – відведений час для відвідування пункту і ε 1,n. cij – необхідний на переміщення між вершинами час і ε 1, n та j ε 1,n. Для формулювання математичної постановки введемо додаткові змінні та константи. Sid – час початку відвідування пункту і ε 1,n на маршруті d ε 1,m. xijd – змінна, що приймає значення 1, тільки коли на маршруті d ε 1, m пункт j ε 1, n відвідується після пункту і ε 1, n. M – константа, значення якої достатньо велике відносно потрібних даних. Вихідна змінна yid, ідентифікує відвідування певної вершини на маршруті визначається так: якщо на маршруті d ε 1, m yid = { відвідується пункт і ε 1, n 0, в іншому випадку.

$$y_{id}=\begin{cases}\mathbf{1}, \text{якщо на маршруті } \mathbf{d}\in\overline{\mathbf{1,m}}\\ \text{відвідується пункт } \mathbf{i}\in\overline{\mathbf{1,n}}\\ \mathbf{0}, \text{у іншому випадку}\end{cases}$$

Цільова функція, позначає сумарну корисність відвідвідування всіх вершин, що включені в маршрути, записується наступним чином:

$$\sum_{d=1}^{m}\sum_{i=2}^{n-1}\text{Score}_i y_{id}\rightarrow\max$$

Початкова та кінцева вершина повинні бути відвідані рівно m разів:

$$C_n=T_{max}$$

Обов'язково необхідно врахувати, що пункт однаково відвідується і покидається однакову кількість разів (правило збереження потоку).

$$\sum_{i=1}^{n-1}x_{ikd}=\sum_{j=2}^{n}x_{kjd}=y_{kd},k=\overline{2,n-1},d=\overline{1,m}$$

Проміжок часу між двома послідовними пунктами, достатньо довгий для виділення часу, щоб відвідати перший пункт і переміститись між пунктами, визначається за формулою:

$$\square s_{id}+T_i+c_{ij}-s_{jd}\le M\left(1-x_{ijd}\right),\ i=\overline{1,n},\ j=\overline{1,n},d=\overline{1,m}$$

Таким чином формується обмеження на відвідування кожного пункту не більше 1го разу на кожному маршруті:

$$\sum_{d=1}^{m}y_{kd}\le 1,\ k=\overline{2,n-1}$$

Загальна тривалість кожного маршруту не більшеTmax, тому математична постановка міститьобмеження такого вигляду:

$$\sum_{i=1}^{n-1}\left(T_i y_{id}+\sum_{j=2}^{n}c_{ij}x_{ijd}\right)\le T_{max},\quad d=\overline{1,m}$$

Початок відвідування певної вершини повинен відбуватись урамках певного часового вікна, тому введені наступні обмеження:

$$O_i\le S_{id}\ ,\ i=\overline{1,n},\ d=\overline{1,m}$$

Cn = Tmax, бокінцевий пункт можна відвідати не пізніше Tmax. T1 = Tn = 0, бо кінцевий і початковий пункти не потребують часу для відвідування. розв'язком задачі є:

- розбиття множини вернинграфу на підмножини (маршрути);
- задання порядку обходу на кожній підмножині.

Розв'язок є прийнятним (допустимим), якщо всі маршрути задовольняють обмеженням задачі.

3. Метод розв'язаннязадачі

За складністю задача SCPTW не менш складною аніж TOP.B своючергу,TOP-модифікація задачі OP (OP- частковий випадок TOP) тому, як і OP, TOP і SCPTW є NP-складною задачею. Доведення, що задача OP є NP-складною задачею наведено у[6]. Задача SCPTW оптимально розв'язується точними

методами, але ці методи використовуютья для розв'язання задач із невеликою кількістю вершин. До вищезгаданих методів відносять динамічне програмування і метод гілок та меж. Наприклад, уроботі [7] SCPTW розвязана точними методами для графу з 30 вершинами. Зваживши на складність данної проблеми більшість літературних джерел присвячено розв'язку задачі не точними методами. Для розв'язку поставленої задачі використано два ефективних і добре відомих алгоритми: алгоритм повторюваного локального пошуку та алгоритм імітації відпалу. В їх основу покладено однакову процедуру локального пошуку [8]. Побудова розв'язку засновується на тому, містить щокоженз d ∈ 1, m маршрутів початкову і кінечну вершини. Для кожної вершини, що не належить жодному маршруту, визначається величина shifti i ∈ 1, n. Дана величина позначає те, зміщення в часі прибуття до вершини перед якою буде розміщена нова вершина. А коли для кожної вершини визначена найкраща позиція, проводиться оцінка Scorei2/shifti. Метою оцінки є побудова коротких маршрутів між вершинами максимізація сумарної корисності всіх маршрутів (Routes).

Обирається вершина з максимальною оцінкою і пошук повторюється спочатку. Робота локального пошуку завершується, при умові, що не можливо додати нову вершину через обмеження маршрутів у часі. Після цього проводиться збурення, що являє собою вилучення вершин змаршруту. Потім процерура локального пошуку повторюється. Отримане значення цільової функції може бути меншим ніж попереднє,тоді попередній розв'язок замінює поточний. У протилежному випадку поточний розв'язок стаєновим і знову відбувається збурення. Алгоритм завершує свою роботу, коли виконана кількість ітерацій описана умовою задачі. Для скорочення часу роботи обчислення околу при виконанні процесу локального пошуку відбувається паралельно. Отже для кожної вершини, яка не входить до маршрутів, незалежно обраховується найвигідніша позиція для розміщення вершини. Алгоритм імітації відпалу діє як і алгоритм повторюваного локального пошуку, тобто використовуємо модифікований локальний пошук, який описано вище. Імовірність прийняття рішення, яке погіршує значення цільової функції вираховується згідно розподілом Больцмана, а розклад вираховується на основі геометричного закону з константою, щодорівнює 0,95.

4. Обчислювальнийекпериментт

Для всіх алгоритмів взяті однакові вхідні дані, а саме:

- моделюється 2 маршрути;
- програмується 20 ітерацій для кожного алгоритму.

В таблиці №1 наведено результати роботи алгоритмів повторюваного локального пошуку із застосуванням модифікованого та немодифікаваного алгоритмів.

Проведено 10 прогонів та зафіксовано середнє часу їх роботи. На вхід подаємо очікувану імовірність прийняття розв'язку задачі, що мають найгірше значення ЦФ. У цьому випадку обираємо значення 0,79. На рисунку 1 викладені результати роботи алгоритмів імітації відпалу і повторюваного пошуку. На графіку наведено залежність розмірності задачі від значення ЦФ.

Табл. 1 – усереднений час роботи алгоритму повторюваного локального пошуку

Кількість вершин	50	74	98	146	164	194	218	242	290
Час роботи ЛП, с	4,98	21,44	27,9	47,73	57.78	78.74	89.2	156.2	167.4
Час роботи модифікованого ЛП, с	3,23	12,3	21,03	39,56	44.43	69.33	70.8	108.8	127.7
Покращення, %	35	42,6	24.34	23,09	23,09	11,92	20.6	30.61	23.75

5. Інтерпритація результатів

Зважаючи на те, що обрахунок околу в локальному пошуку є вельми трудомістким, то із застосуванням паралельних обчислень отримаємо пришвидшення роботи алгоритмів,

тому таку модифікацію вважаємо доцільною. Розглянуті алгоритми дають подібні результати роботи,але алгоритм повторюваного локального пошуку знаходить більш кращі рішення. Ця ситуація може змінитись при збільшенні кількості ітерацій. Але необхідно зазначити, що задача TOPTW розв`язується щоб в подальшому лягти в основу мобільних застосувань та веб-додатків. Тому надається перевага алгоритмам, час роботи яких менший та менш ресурсомісткий. А оскільки, оцінка початкової температури досить трудомістка процедура, яку треба проводити перед початком роботи алгоритму на новому наборі даних. Тому у більшості випадків надається перевага алгоритму повторюваного локального пошуку.

Висновок

Наведено і описано математичну постановку задачі роботи керівника мобільної групи з урахуванням часових вікон. Задачу вирішено з використанням алгоритмів імітації та повторюваного локального пошуку. Побудова маршрутів у виконаній задачі роботи керівника мобільної групи базується на задачі комівояжера. Проведене обчислення для розахунку часу роботи зазначених алгоритмів і проведена порівняльна характеристика отриманих результатів.

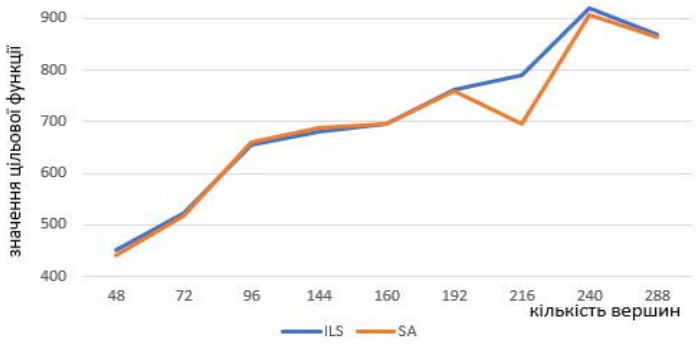


Рис.1. Результати роботи алгоритмів повторюваного локального пошуку і імітації відпалу

Списоклітератури

1. Vansteenwegen P. The orienteering problem: A survey / P. Vansteenwegen, D. Oudheusdena, W. Souffriau. // 209. – 2011. – №1. – С. 1–10.
2. Kara I., Derya T., Bicakci P. New Formulations for the Orienteering Problem // Procedia Economics and Finance. – 2016. – №39. – С. 849–854.
3. Gavalas D., Mastakas K., Charalampos K., Pantziou G. A survey on algorithmic approaches for solving tourist trip design problems // Journal of Heuristics. – 2014. – №20. – С. 291–328.
4. Tackling large-scale home health care delivery problem with uncertainty. / C.Chen, Z. Rubinstein, S. Smith, H. Lau. // Twenty-Seventh International Conference on Automated Planning and Scheduling. – 2017. – №27. – С. 18–23.
5. TRACCS: A Framework for Trajectory-Aware Coordinated Urban Crowd-Sourcing / [C. Chen, S. Cheng, A. Gunawantaін.]. // HCOMP. – 2014.
6. Golden L. The orienteering problem / L. Golden, R. Vohra, L. Levy. // Naval Research Logistics. – 1987. – №34. – С. 307–318.
7. Kim B. A Branch-and-Price Approach for the Team Orienteering Problem with Time Windows / B. Kim, H. Tae. // The International Journal of Industrial Engineering: Theory, Applications and Practice. – 2015. – №22. – С. 243–251.
8. Iterated local search heuristic for the team orienteering problem with time windows / P.Vansteenwegen, W. Souffriau, G. Berghe, D. Oudheusden. // Computers & Operations Research. – 2009. – №36.– С. 3281–3290.
9. Walid B. Computing the Initial Temperature of Simulated Annealing / Ben-AmeurWalid. // Computational Optimization and Applications. – 2004. – №29. – С. 369–385.

УДК 519.7; 681.513.7; 612.8.001.57; 007.51/.52

НАБОКОВ Е.М .,
ГАВРИЛЕНКО О.В.

МЕТОДИ ВИРІШЕННЯ ПРОБЛЕМ З ПЕРЕВАНТАЖЕНИМ ТРАФІКУ У МЕРЕЖАХ.

У сучасному світі все більше й більше даних передається між користувачами інтернету. Дані можуть бути різного формату: фільми, музика, документи, тощо. Кожні дані мають свою вагу, та передаються завдяки маршрутизаторам. Проблема виникає у кількості даних, які передаються у даний момент часу. Тому у даному документі наводяться приклади як зменшити навантаження в Інтернеті. Ці оптимізації можуть бути засвоєні як у глобальних мережах так і у локальних. Ідея полягає в пришвидшенні обміну трафіку та його полегшенням.

METHODS OF SOLVING ISSUES WITH OVERLOADED TRAFFIC IN NETWORKS.

Nowadays, more and more data is transmitted among different consumers of the Internet. Data can be kept in different formats such as: movies, music, documents, etc. Each data has its own weight, and it’s transmitted with routers. The problem is raised when a huge amount of data is transmitted at the certain point of time. Therefore, there is an explanation with examples how to diminish load of the Internet traffic. These optimizations can be applied both in global networks and in local networks. The main idea is about accelerating exchange of traffic and making it easier.

KEYWORDS
The subject of the article is a review of methods that solve issues with overloaded traffic in networks. There are multiple approaches that are related to different facets of optimizations such as cryptography, compression, network routing.

1. Introduction

Nowadays, theworldutilizestheInternetforconsumingande xchanginginformation. As for rule of thumb, any information is kept on servers. These servers are provided either by incorporated companiesor public organization.

To retrieve any information from those servers, there is an existing solution that is used by all of the Internet users – BGP and DNS.

BGP is the compulsory component of the Internet. It allows to union different networks between each other. BGP solves the problem of routing between networks itself. Meanwhile, DNS is just a convenient wrapper, which allows people to avoid using IP address directly, but IP address instead. These 2 systems are fragile and monstrous. Nevertheless, it’s supported by both many public organizations and private companies. The end of the Internet era would come true only if BGP was broken. Until then we are good to use the Internet resources that it provides for us.

Any user requests information from one end being in another end. To support routing between any 2 ends, there are existing hardware computers: routers, hubs, switches. All of them are responsible for routing requests from end user to the target server and vice versa. In case of any computers knows nothing about target server, it sends request into outer network. As long as everything is determined in the computer networks, every target already exists in the world wide web.

Any request goes via hundred routers, where each of routers can read the content of the request. Here the problem comes. As long as we don’t control those routers, we cannot be sure that information is secure to send it in its original form. Therefore, it is invented cryptographical algorithm to encrypt request in a way, so only target server can decrypt it. It’s called asymmetric cryptography or cryptography with public key. It means that

anybody or anything can encrypt a request, but only target server can decrypt the request.

The main idea for cryptography is to change content of a request so it can be reverted back using public and private keys accordingly. Besides it's absolutely recommended to make all messages with the same length so to avoid giving any hints to the crackers [1]. So currently all encrypted messages look almost the same. Even though it makes data exchange more secure, it burdens the data exchange traffic in the world web. In other words, it sends 10Kb over a recipient in lieu of 1Kb or less. It works well for a single request but does not work for millions of requests. In addition, the problem of routing is still present. It means that any two requests might reach the same target using different routes. One of them can be longer, another one – shorter. How does routing impacts world traffic? Every request which goes via different routes is accumulated in those routers until it's sent further. A router is not black box, which sends requests impeccably and quickly. It has its own bounded resources like RAM and CPU. It can accumulate only certain number of requests, but not more. In case it cannot handle more requests, it denies all upcoming. It means the requests for google.com may not work at all.A router goes through the list of requests one by one, parses each one for source and target destinationsand sends it further according to its route table. So, it's highly important to make all requests simple and small [2].

Moreover, there is routing problem. A router checks target destination of a request and tries to find the matching rule in its route table. If the one was found, it sends the request further. A router does not verify whether the target destination is sent with the shortest path or not. It picked up just the first matched one.

In this article is being reviewed different approaches to optimize data exchange in a network using cryptography - to make any date secure, compression – to make any data smaller and accelerated routing, which is based on meta information about target destination: how far they are, how loaded they are.

2. Formulation of the problem

There are 3 networks with 10 serverseach. And there are 100,000 users who use the networks. Each user has an ability to save any encrypted information: music, movie, picture (later "object")in one of those serversand to request this information from the server. It's required to determine how to optimize (accelerate, simplified, reduce) data exchange.

3. Encryption of the data

Any information such as music, movie, picture is just a set of bytes, which consists of bits. Let's assume we have 1Kb of $x_o$that is required to be encrypted into x_e.

$$x_o(n_1, n_2, \dots, n_k) \rightarrow x_e(m_1, m_2, \dots, m_k) \quad (2)$$

where x_o – original message, x_e – encrypted message, k – number of bits, n and m are 1 or 0.

It's a one-to-one mapping. Each bit is transformed into another bit by algorithmic rule. It means that the key with the size k can encrypt only $k - padding\ length$ of the original message. There are various key sizes: 128-bit (16 bytes), 256-bit (32 bytes), 512-bit (64 bytes), 1024-bit (128 bytes), 2048-bit (256 bytes), 4096-bit (512 bytes). Since we have 1Kb message, it's impossible to encrypt with any of those keys [3].

Therefore, there is the first approach.We have to split the original message at first. It's practical to split into parts, where 64 bytes each. It means we get 16 parts with 64 bytes each.

$$x_o(n_1, n_2, \dots, n_{k1}) \rightarrow x_e(m_1, m_2, \dots, m_{k1})$$
$$x_o(n_1, n_2, \dots, n_{k2}) \rightarrow x_e(m_1, m_2, \dots, m_{k2}) \quad (1)$$
$$\dots$$
$$x_o(n_1, n_2, \dots, n_{k16}) \rightarrow x_e(m_1, m_2, \dots, m_{k16})$$

where x_o – original message, x_e – encrypted message, k*i* – number of bits, n and m are 1 or 0 for each part of the message. All $k_i$values are the same.

$$\Sigma_k\big(x_o(n_1, n_2, \dots, n_k)\big) \leftrightarrow \Sigma_k(x_e(m_1, m_2, \dots, m_k)) \quad (3)$$

To encrypt 64 bytes of the message, we have to use key with the size bigger than 64 bytes. We may use 1024-bit key. As long as the key is bigger than a single part, there is a solution to solve it – padding. It allows you to add up dummy data into part to match the length of the key. Indeed, this dummy data adds up load to the Internet traffic. And it does not even provide any useful information [4].

Here is a rough calculation of the size of the encrypted message that is sent over the Internet from one user to another one:

The original message with the size of 1024 bytes is converted into 128 bytes of key * 16 (number of splits for original message). An equation looks like the following:

$$1024\ bytes \rightarrow 128\ bytes\ of\ the\ key * 16 = 2048\ bytes \quad (4)$$

As we see, the encrypted message weighs double bigger than the original one. It means that every encrypted message that is intended to be sent over the Internet will be double bigger. It loads the Internet traffic significantly [5].

To avoid that,there is a 2 approach. We have split message into such parts, so each one should be almost the same as a key size.It's worth to mention that it's not practical to encrypt 64 bytes with 512-bit key, because it's highly recommended to use padding to fix weaknesses of public key encryption. The padding may be fully randomized, which increases entropy of the original message.So, two exact the same messages can have different encrypted outcome. It does increase security facet [6]. Padding should be at least the size of 12 bytes to make it work.So, let's do backpropagation starting from key size of 512-bit (64 bytes).

$$64 - 12 = original\ message\ length\ (in\ bytes)$$
$$\frac{1024\ bytes}{52\ bytes} = 19 + padding\ for\ the\ last\ part \quad (5)$$

So now it has the following equation (all in bytes):

$$1024\ original\ message \rightarrow 64\ (key\ size) * 19 + 0.69 * 64 = 1260.30\ bytes$$

The ratio of difference: (6)

$$\frac{1260.30\ bytes}{2048\ bytes} * 100\% = 61.53\%$$

For now, we succeeded to diminish the resulting encrypted message in 61.53% of the previous result. It means that the traffic with encrypted data will be in 61.53% lighter [7]. It leads to accelerating the data exchange. Besides, routers/hubs/switches can afford to operate on more requests. Here is a comparing table between 2 approaches for cryptography:

Table 1. Comparing sizes of data

Key size	Original message (bytes)	Encrypted message using 1 approach (bytes)	Encrypted message using 2 approach (bytes)	How effective the second approach is

128-bit (16 bytes)	1024	2048	4096	-100%
256-bit (32 bytes)	1024	2048	1638.4	80%
512-bit (64 bytes)	1024	2048	1260.31	61.5%
1024-bit (128 bytes)	1024	2048	1129.93	55.2%

There is an exceptional case when the key size is 128 bits. It's because the padding affects significantly due to being much bigger than the original message is. Nevertheless, we can see the tendency that the bigger file, the better 2 approach works.

As a conclusion for the encryption it's highly recommended to use padding for public key encryption. Splits of the any data must be flexible and be in size relative to the key size including padding. In addition, the bigger data, the less encrypted data we send over the Internet.

Table 2. Comparing sizes of data

Key size	Original message (bytes)	Encrypted message using 1 approach (bytes)	Encrypted message using 2 approach (bytes)	Ratio between first and second one
128-bit (16 bytes)	1024	2048	4096	-100%
256-bit (32 bytes)	10240	20480	16384	80%
512-bit (64 bytes)	102400	204800	126030	61.5%
1024-bit (128 bytes)	1024000	2048000	1129931	55.2%

4. Compression of the data

Compression allows to shrink data and expand it back. It's responsible to convert one data form into another one. It works with 2 options:

- with loss
- without loss

$$x_o(n_1, n_2, \dots, n_k) \rightarrow x_c(m_1, m_2, \dots, m_z) \quad (7)$$

where x_o – original message, x_c – compressed message, k – number of bits of the original message, z – number of bits of the compressed message.

It's worth to note that the original message can be either encrypted or original one. The matter is about transferring less data via routers. Let's assume we use encrypted message as an original one. Its size is 1Kb.

So, the point of lossless compression algorithms is to transform data in a way it can be understood, read, decompressed by others meanwhile it weighs less than the original one. It means that compression works every time differently. It depends on characters that stand in a row, which are being swapped by compressing algorithm. Here is a vanilla example which illustrates how compression works:

$$AAAABBBBBBCCC \rightarrow A\backslash 3B\backslash 5C\backslash 2 \quad (8)$$

It's just a basic example, which shows algorithm, which is based on repeated elements. One problem that should be mentioned – the less data to compress, the more complicated to compress. The idea that comes around different existing algorithms is about learning past data to compress future data.Usually two third is compressed by algorithms. So, in case of 1KB data, we will obtain approximately 0.335KB. As long as we

are trying to optimize the data exchange which goes inside the networks, a well determined and tested compressing algorithm can reduce a load of any network between users.

In addition, compression and encryption layers require to have opposite operation – decompression and decryption. Thus, it reduces load for networks to pass data, but it increases time to be able to read data that is being sent over the Internet.

5. Routing of the data

Besides implementing optimizations for encryption and decryption, there is one more valuable optimization. It's data routing in the networks. As far as we know, that any networks consist of a bunch of routers, hubs, switchers, it means that a path between any 2 vertices must be the shortest. It will allow to pass data without any additional operations of extra routers.[8]

It can be implemented in many different ways. As a starting point, we have to assume that currently routers just work based on their route tables. They find the matched pattern for request's destination in their tables and send it to another target router that is responsible for part of network. All routers, hubs, switches are discoverable using BGP protocol [9]. It unites all of them into a single big network.The internet would not live if BGP didn't exist. Here is an example of routing table for any router [10].

#	Destination	Gateway	Ping
1	0.0.0.0/0	192.168.0.253	100ms
2	0.0.0.0/0	192.168.0.255	90ms
3	127.0.0.0/16	127.0.0.1	1ms
4	35.2.84.0/8	74.52.178.12	54ms
5	54.21.31.0/8	0.0.0.0	10ms
6	172.168.0.0/16	127.0.0.1	2ms

Table 3. Simple route table

#	Destination	Gateway
1	0.0.0.0/0	192.168.0.253
2	127.0.0.0/16	127.0.0.1
3	35.2.84.0/8	74.52.178.12
4	54.21.31.0/8	0.0.0.0
5	172.168.0.0/16	127.0.0.1

Table 4. Twisted route table

To optimize route-finding, we may rely on BGP.For now, routers work straightforwardly. To make it better, we have to implement a smart router. Its duties are searching, analyzing other routers and creating routing table that is based on how far routers are, how loaded they are. To do so, it requires to add up a new software layer, which must be highly optimized as well. It should preferably be a binary, which has the lowest footprint and works as fast as possible. This is important because it's a layer that adds up operational costs, and the time to send packet further is increased because of that.

The column with Ping results shows us how far and how loaded the destination point is. For now,a router can redirect requests to low loaded routers. In the networks, distance which is physical is not as important as a ICMP response of the router from another end. Thus, we are good to use only ICMP requests like ping, to be about to determine how far and how loaded a router is. ICMP requests and its response can be assessed in ms. It represents a time within the requester gets its response. The bigger time, the more loadedor further a router is. Any routing problem can be illustrated as a graph. There are a lot of ready-to-go solutions, which solve different problems. One of the most popular and problematic tasks is a pathfinding.The edges of the graph can be represented with weights. Literally, weights are ICMP response time. It's worth to note that a router itself does ICMP requests to all its known endpoint to figure out how far and how loaded they are.

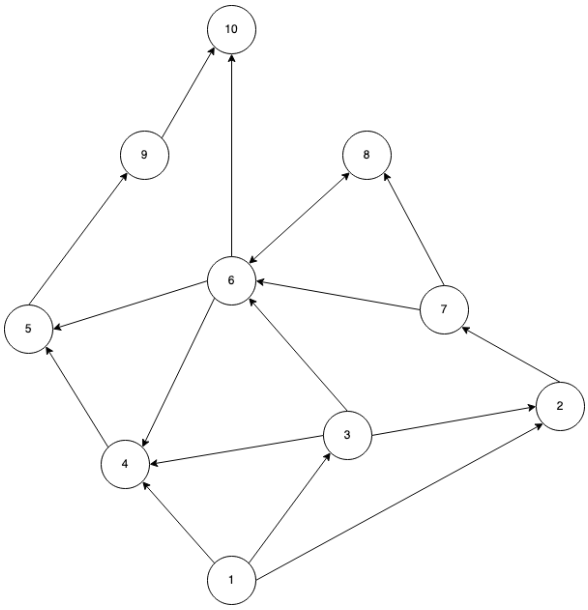


Figure 1 Routing graph

It's a hard problem to solve. Any router might lead to nowhere, so a packet will return to its own sender with an error of "Error destination". Routing itself works impeccably, it sends a packet to a subnet. There is a router for each subnet. The router must know where to send packet to. The shortest path from 1 to 10 will be the following:

$$1 \rightarrow 4 \rightarrow 5 \rightarrow 9 \rightarrow 10 \qquad (9)$$

The total time to send data to a destination would take undefined time, because there is no awareness about current state of the network.

Although if there are weights for edges (connections between routers), then the outcome will be different.

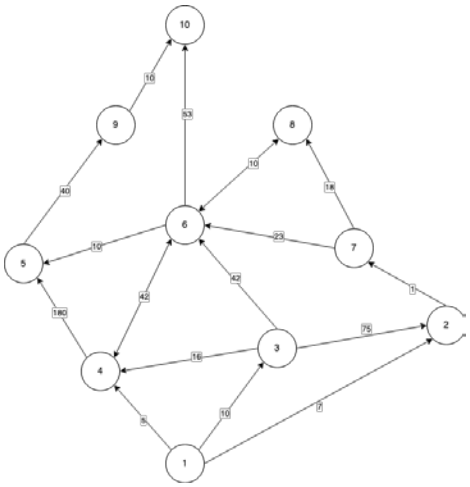


Figure 2 Routing graph with latencies

After adding latencies between routers, the shortest path is now totally different:

$$1 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 9 \rightarrow 10 \qquad (10)$$

The total time to send data would take 107 ms. For instance, the previous example would take 235 ms.

According to the greedy algorithm, we may find the shortest path if only we stick to the shortest latencies. This algorithm works well only when latencies is descending with the following router.

There are heuristic algorithms, which analyzes a few steps ahead. Literally, it makes recursively request to different routers and assess their assessment to their routers. It allows to accumulate more information and find out which router works the best to send request to.

Let's assume we use this algorithm with 1 step ahead; the shortest path will be the following:

$$1 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 9 \rightarrow 10 \qquad (11)$$

With the heuristic algorithm, it would take only 91 ms.

As far as we see, it's totally different from the results above. The only problem that to get information about other routers, it takes some time, so it's highly dynamical approach to solve the shortest path in networks. The shortest path may go via more routers with the new approach in contrast to previous approaches. A path is based on latencies between routers, but not physical distance. It may have different quality of fiber channel between routers and so on. So, the system must be dynamical, but not statical, and rely on current state of their neighbors.

Table 5. Comparing algorithms

Plain algorithm	Greedy algorithm	Heuristic algorithm
235ms	107ms	91ms

As it's shown in the Table 3, it points that current routing in the networks is quite problematic and takes much more time than the one which is heuristic.

Conclusion

We reviewed different approaches how to optimize data exchange in networks. It helps significantly a lot when it all comes together. Encryption now consumes less memory and reduces load in networks. Compression shrinks data, which allows to transform 1KB data into tiny data, which is easier to operate with and to be sent to other recipients. In addition, sending data itself is a compulsory part of optimization as well. It was found out that the algorithms with heuristics work much better than greedy ones. It was assessed time within the packet is delivered to the destination.

In result of this analysis it was found out that current routing system in the Internet works worse than the results of proposals mentioned in this article.

Taking all these factors into account, it can be claimed that the Internet may work much faster and securer than it works currently in spite of the fact that decryption, decompression are expensive operations.

REFERENCES

1. RSA algorithm [Electronic resource]. https://www.di-mgt.com.au/rsa_alg.html [in English].
2. RSA: Start validating your cybersecurity effectiveness. [Electronic resource].https://www.verodin.com/post/rsa-2020-validating-cybersecurity-effectiveness [in English].
3. RSA key length [Electronic resource]https://www.javamex.com/tutorials/cryptography/rsa_key_length.shtml [in English].
4. How compression works for video materials [Electronic resource]. https://www.maketecheasier.com/how-video-compression-works/ [in English].
5. Lossy compression [Electronic resource]. https://en.wikipedia.org/wiki/Lossy_compression [in English].
6. Lossless compression [Electronic resource]. https://en.wikipedia.org/wiki/Lossless_compression [in English].
7. Difference between lossy and lossless compression [Electronic resource]. http://www.cvisiontech.com/resources/jbig2-compression-primer/lossless-lossy-perceptually-lossless-compression.html [in English].
8. A* path finding [Electronic resource]. http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html [in English].
9. A comprehensive study on pathfinding techniques for robotics and video games [Electronic resource] https://www.hindawi.com/journals/ijcgt/2015/736138/ [in English].
10. Border Gateway Protocol [Electronic resource] https://en.wikipedia.org/wiki/Border_Gateway_Protocol [in English].

УДК 004.94

ПЕДОРЕНКО А.В.
СТЕЦЕНКО І.В.

ПАРАЛЕЛЬНИЙ АЛГОРИТМ ОБЧИСЛЕНЬ ПЕТРІ-ОБ'ЄКТНИХ МОДЕЛЕЙ

У даній статті розглянуто паралельний алгоритм обчислень Петрі-об'єктних моделей. Вивчається класифікація та побудова паралельного алгоритму. Наведено результати експерименту порівняння швидкодії послідовного на паралельного алгоритмів.

СТОХАСТИЧНІ МЕРЕЖІ ПЕТРІ, ПАРАЛЕЛЬНІ ОБЧИСЛЕННЯ, АЛГОРИТМ ІМІТАЦІЇ

The subject of this article is a parallel algorithm for computing Petri-object models. It studies the classification and construction of the parallel algorithm. Provided the results of the experiment of comparing the speed of sequential to parallel algorithms.

STOCHASTIC PETRI NET, PARALLEL CALCULATIONS, SIMULATION ALGORITHM

1. Вступ

Потужним інструментом моделювання є Петрі-об'єктні мережі, однак через їх сильну деталізацію розмір моделі швидко розростається, що призводить до підвищення часу роботи симуляції. Оскільки у комп'ютерах, на яких проводяться дослідження моделей все збільшується кількість центральних процесорів, розпаралелювання алгоритму симуляції призведе до значних заощаджень часу, витраченого на дослідження, а значить і витрат бізнесу на моделювання своїх процесів. Ця ідея призвела до створення даної роботи.

2. Класифікація підходів до паралелізму

Основна мета паралелізму — розбиття послідовної роботи алгоритму на менші частини, які можуть обчислюватися одночасно. Паралельний алгоритм імітації можна представити у вигляді декількох послідовних, які обмінюються між собою повідомленнями з вказаним часом їх виникнення.

Є декілька підходів до паралельної імітації дискретно-подійних моделей, які згідно з Джейсоном Лю [1] поділяються на чотири класи:

• тиражування випробувань: до цього типу паралелізму приписують одночасний запуск декількох спроб імітації моделі. Ефективність цього підходу висока, оскільки не потрібно синхронізуватися між процесами імітації. З іншого боку, мінімальний час обчислення обмежений часом обчислення однієї спроби;

• функціональна декомпозиція: цей підхід пропонує паралельно виконувати частини алгоритму імітації в межах однієї ітерації. Масштабованість цього підходу низька, оскільки різні частини однієї ітерації зазвичай сильно залежать одна від одної. Та все ж, в деяких моделях можна, наприклад, паралельно виконувати оновлення стану моделі;

• часова декомпозиція: цей підхід пропонує розділяти час імітації на менші інтервали, які імітуються паралельно, а потім глобально синхронізувати стан. Цей тип паралелізму добре масштабується, але його ефективність обмежена розміром інтервалу паралельної імітації та частотою й складністю синхронізації;

• просторова декомпозиція: в цьому випадку модель розбивають на менші частини, імітація на яких відбувається паралельно. Події, що відбуваються на межі частини, повинні оброблятися сусідньою частиною у відповідний час імітації, а конфлікти між частинами мають вирішуватися за допомогою синхронізації. У цього підходу великий потенціал, але його ефективність залежить від вибору методу синхронізації та від природи моделі, що імітується.

Методи імітації просторово декомпонованих моделей можна розділити на синхронні та асинхронні. В синхронних методах події на межах синхронізуються в глобальні часові кроки, на яких всі потоки зупиняють свою імітацію та межовим станом частини моделі з сусідніми частинами моделі. Якщо часові кроки у всіх частин моделей рівні, то синхронізація виконується після кожного такого кроку. У випадку, коли час кроку частини відрізняється від часу глобального кроку, методи ігнорують синхронізацію проміжних інтервалів та синхронізуються в глобальні часові кроки. Зазвичай це призводить до чисельної помилки в обчисленні моделі. В асинхронних методах межові конфлікти вирішуються асинхронно та одночасно з локальною імітацією, під час якої сусідні частини моделі постійно обмінюються їх межовим станом. Отже, такі методи можна використовувати для точного паралельного імітування моделей, у яких у кожній частині моделі часові кроки просування стану відрізняються.

Проблема синхронізації між частинами моделі в асинхронному методі просторової декомпозиції визначає ефективність отриманого в результаті паралельного алгоритму. При занадто частій синхронізації алгоритм буде неефективним, а якщо виконувати синхронізацію рідко, результат буде не точним, або зростуть затрати пам'яті. У роботі [2] наведено два способи синхронізації:

• консервативний: виконання кожної частини моделі блокується, поки не можна гарантувати, що з сусідніх частин ніколи не надійде подія с часовою позначкою меншу за визначену. Це досягається шляхом повідомлень, які частини моделі посилають одна одній під час виникнення межових подій. Оскільки повідомлення надсилаються тільки під час виникнення внутрішніх межових подій, у наступних повідомлень позначка часу завжди буде більшою за попередні, тому що час у моделях просувається тільки в одну сторону. На жаль, такий підхід призводить до дедлоку, коли частини моделей утворюють цикл і кожна чекає на повідомлення від попередньої, щоб визначити інтервал часу, в якому гарантовано не буде зовнішніх подій. Для вирішення цієї проблеми частини моделі повинні додатково надсилати повідомлення з часовою відміткою їх найближчої внутрішньої події. Оскільки час найближчої події не більший за межову подію, сусідня частина моделі може безпечно імітуватися в цьому інтервалі. Однак, повідомлення з часом найближчої внутрішньої події приводять до неефективності, тому що стрімко зростає частота синхронізації між частинами моделі;

• оптимістичний: виконання кожної частини моделі не блокується, але при надходженні повідомлення від сусідньої частини, якщо часова позначка в ньому менша за поточний час в частині, виконується відкат. За час роботи частини моделі в неправильному порядку могли відбутися два типи подій: зміна внутрішнього стану та надсилання повідомлення сусіднім частинам моделі. Для відкату стану частини моделі необхідно зберігати знімки її станів у певний час. Щоб розв'язати проблему з повідомленнями потрібно надіслати так зване анти-повідомлення, яке знищить початкове повідомлення, якщо воно ще не було опрацьовано, або запустить відкат у цій сусідній частині. Рекурсивно ця процедура повністю прибере ефект від помилки. Цей спосіб має дві центральні проблеми: відкат може займати дуже багато часу, якщо якась частина моделі встигла дуже далеко просунути свій локальний час, та на зберігання знімків моделі витрачається дуже багато пам'яті.

3. Опис паралельного алгоритму обчислень Петрі-об'єктних моделей

Алгоритм імітації Петрі-об'єктних моделей розглянуто у роботі [3]. В цьому алгоритмі використовується підхід просторової декомпозиції, а синхронізація виконується консервативним способом. Петрі-об'єктна модель розділяється на Петрі-об'єкти, які самі просувають свій локальний час. У наведеного алгоритму є декілька обмежень, що не дозволяють йому бути універсальним:

• кожен Петрі-об'єкт може мати максимум один сусідній об'єкт, який надсилає йому повідомлення;

• кожен Петрі-об'єкт може мати максимум один сусідній об'єкт, якому він надсилає повідомлення;

• Петрі-об'єкти в моделі не мають утворювати цикл.

Для подолання перших двох обмежень необхідно замість одного буферу зовнішніх подій мати на Петрі-об'єкті декілька таких буферів, по одному на кожне з'єднання з іншим Петрі-об'єктом, який буде надсилати повідомлення. При такому розширенні алгоритму потрібно змінити правила визначення межі безпечного інтервалу моделювання та тимчасове зупинення роботи при перевищенні ліміту зовнішніх подій.

Межа безпечного інтервалу буде визначатися за наступними правилами:

1) серед усіх буферів зовнішніх подій знайти першу подію з буфера з мінімальною часовою позначкою;

2) якщо подія знайдена і її часова позначка менша за час моделювання, то межа моделювання встановлюється в цю часову позначку інакше в час моделювання.

Ліміт зовнішніх подій має встановлюватися на мінімальний розмір з буферів зовнішніх подій, у який Петрі-об'єкт надсилає повідомлення.

Щоб подолати обмеження з утворенням циклу потрібно перед обчисленням Петрі-об'єктної моделі провести її перебудову. Оскільки Петрі-об'єктна модель утворює орієнтований граф, задачу можна звести до пошуку компонентів сильної зв'язності

орієнтованого графа. За допомогою алгоритму Косараджу можна знайти усі компоненти сильної зв'язності, а потім об'єднати Петрі-об'єкти, що утворюють ці компоненти у більші Петрі-об'єкти.

Окремо необхідно розглянути проблему Петрі-об'єктів об'єднаних через спільні позиції. Між двома Петрі-об'єктами, які поєднані через спільну позицію відбуваються два види взаємодії: вихід маркерів в спільну позицію та забирання маркерів зі спільної позиції. Повідомлення про ці взаємодії об'єкти також мають передавати один одному в буфер повідомлень. Однак, при визначенні безпечного інтервалу роботи об'єкти будуть блокуватися один на одному саме через те, що вони повинні передавати повідомлення один одному, а це утворює цикл. Тому Петрі-об'єкти, які поєднані через спільну позицію мають сприйматися як такі, що утворюють цикл один з одним, та розв'язуватися так само, як й інші цикли.

4. Порівняння послідовного та паралельного алгоритмів

Для порівняння швидкодії послідовного та паралельного алгоритмів імітації було створено набір Петрі-об'єктних моделей за схемою зображеною на рисунку 1.

Першим до Петрі-об'єктної моделі додається Петрі-об'єкт генератора, а потім від п'яти до п'ятдесяти з кроком п'ять Петрі-об'єктів, які складаються з десяти СМО. Кожен перехід, що відображає СМО виконує вихід маркерів у чергу наступної СМО. Таким чином, загальна кількість переходів змінюється від п'ятдесяти до п'ятисота з кроком п'ятдесят. Результати експериментів наведені у вигляді графіка на рисунку 2.

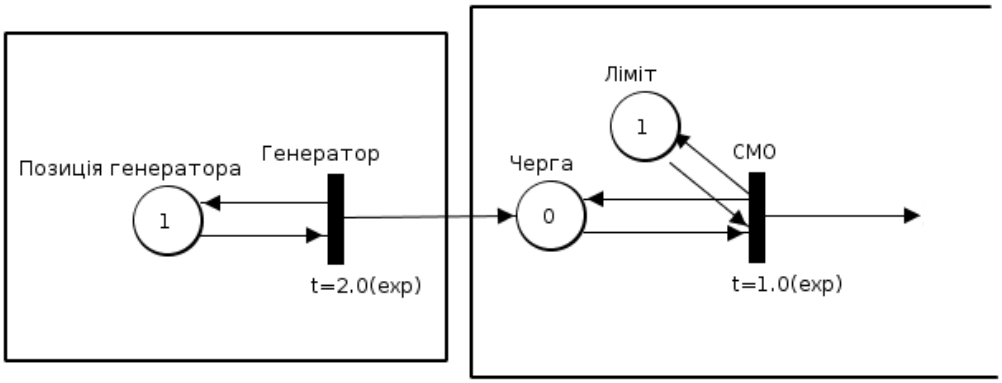


Рис 1. Схема Петрі-об'єктної мережі для порівняння часу роботи послідовного та паралельного алгоритмів

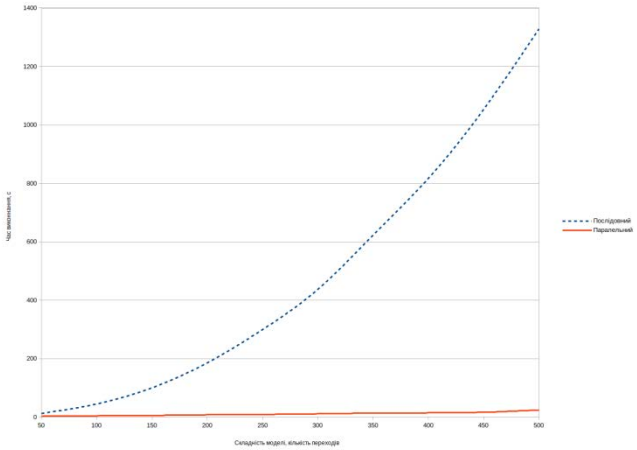


Рис 2. Результати експерименту порівняння швидкодії послідовного та паралельного алгоритмів

З графіка видно, що для побудованої Петрі-об'єктної моделі час роботи послідовного алгоритму кубічно залежить від кількості переходів, а час роботи паралельного алгоритму – лінійно. Для п'ятисот переходів паралельний алгоритм працює швидше послідовного у п'ятдесят п'ять разів.

Експеримент проводився на обладнанні з процесором Intel Core i7 сьомого покоління та з об'ємом оперативної пам'яті 32 Гб.

Висновки

Паралельний алгоритм обчислень Петрі-об’єктних моделей здатен значно облегшити процеси моделювання у багатьох сферах, зменшити час проведення імітацій та підвищити ефективність моделювання як інструменту для роботи у сферах виробництва.

Список літератури

1. Jason L. Parallel Discrete-Event Simulation / Liu Jason., 2010.
2. Fujimoto Richard M. Parallel and distributed simulation // Proceedings of the 2015 Winter Simulation Conference. - IEEE, 2015. - С. 47-52.
3. Стеценко І.В. Паралельний алгоритм імітації Петрі-об’єктної моделі / Інна Вячеславівна Стеценко. // Кібернетика і системний аналіз. - 2017. - №53(4). - С. 130-140.

УДК 004.912

Якимчук О.А.,
Олійник Ю.О.

ПРОГРАМНА БІБЛІОТЕКА ОБРОБКИ ТЕКСТОВОЇ ІНФОРМАЦІЇ ДЛЯ APACHE SPARK

В цій статті розглянуто задачу обробки текстової інформації. Сформульовано цілі та актуальність дослідження. Проаналізовано основні етапи та існуючі математичні методи обробки. Розглянуто існуючі програмні рішення, проведено їх порівняння, наведено переваги та недоліки.

КЛЮЧОВІ СЛОВА: обробка тексту, лематизація, векторизація тексту, реферування, текстовий потік.

This article is about natural text processing problems. Article contains formulation of the goals and relevance of research. The main stages and existing mathematical methods of text processing are analyzed. The existing software solutions are considered, their comparison is made, the advantages and disadvantages are given.

KEYWORDS: natural text processing, lemmatization, vectorization, summarization, text stream.

1. Вступ

Бурхливе зростання кількості електронних документів, що спостерігається в даний час, наочно показує, що традиційні механізми обробки електронних документів не спроможні впоратись з потребами користувачів. Ця тенденція помітна як в мережі Інтернет, так і у корпоративних мережах. Зростання кількості інформації призводить до наступних проблем:

- Більшість технологій роботи з текстовими документами орієнтовані на організацію зручної роботи з інформацією для людини, але практично відсутні можливості для передачі смислового змісту тексту, тобто відсутнє семантичне індексування.
- Не структурована інформація становить значну частину сучасних електронних текстових документів.

При сучасному темпі зростання обсягів інформаційних масивів, не важко уявити, якими надмірно трудомісткими процесами будуть, як класифікація всього фонду електронних текстових документів вручну, так і його кластеризація. Допомогти у вирішенні даної проблеми здатні програмні засоби, які автоматично

виконують інтелектуальну обробку даних. Насамперед, мова йде про текстові документи, які можуть бути представлені у вигляді, придатному для автоматичного аналізу за допомогою програмних засобів. Не зважаючи на те, що Інтерес до даної проблеми, не тільки не згасає, але в останні два десятиліття є підвищеним - досі відсутнє рішення для української та російської мов.

2. Попередня обробка тексту

Процес попередньої обробки вхідних даних є важливим та необхідним етапом в задачах обробки природної мови. Передобробка використовується для виокремлення необхідної інформації з неструктурованих текстових даних.

Попередня обробка тексту включає в себе декілька етапів[1].

Токенізація. Це найперший крок в обробці тексту - розбиття великих шматків вхідного тексту на дрібні частини, токени. Дотокенів відносяться як реченнята слова, так і знаки пунктуації.

Нормалізація. Нормалізацією називають процес трансформації тексту, в єдину канонічну форму, в якій він міг не бути спочатку. В результаті нормалізації всі слова приводяться до одного регістру, видаляються знаки пунктуації та не значимі символи форматування, розшифровуються скорочення, числа приводяться до текстового вигляду, та приведення слів до початкової форми[8].

Останній пункт вимагає розуміння того, як текст буде оброблятися далі і в залежності від цього застосовують один або декілька з наведених нижче методів.

Стемінг.Стемінг – це процес вилучення «зайвого» від кореня і виділення основної частину слова, часто це призводить до втрати словотворчих суфіксів так як основа не обов'язково збігається з морфологічним коренем слова[8].

Лематизація.Лематизація – процес проведення слова до нормальної форми, який використовує словник і морфологічний аналіз[9]. В результаті лематизації від словоформи відкидаються флективні закінчення і повертається основна або словникова форма слова. Наприклад результатом лематизації для дієслова буде інфінітив, а для іменника чи прикметника називний відмінок однини.

Відмінність в тому, що стемер діє без знання контексту і, відповідно, не розуміє різницю між словами, які мають різний зміст в залежності від частини мови. Однак у стемера є і свої переваги: його простіше впровадити і він працює швидше. Також варто зауважити, що більш низька «точність» може не мати значення в деяких випадках.

Видалення стоп-слів. Стоп-слова – це слова, які викидаються з тексту до/після обробки тексту. Коли ми застосовуємо машинне навчання до текстів, такі слова можуть додати багато шуму або створити аномалії, тому необхідно позбавлятися від нерелевантних слів. Під стоп-словами розуміють нецензурну лексику, вигуки, сполучники і т.д., які не несуть сенсу чи значеннях яких потрібно враховувати. Їх ще називають шумовими словами. При цьому потрібно розуміти, що не існує універсального списку стоп-слів, все залежить від конкретного випадку.

3. Виділення додаткових ознак

Після попередньої обробки для кожного знайденого терму виділяються ознаки, за якими можна оцінити ступінь його важливості. Виділені ознаки можна розділити на 3 категорії: синтаксичні, статистичні, та структурні.

До **синтаксичнихознак** відносять ознаку частини мови, яка одержується за допомогою POS-розміток (part-of-speech tagging), та ознаки, отримані за допомогою різних онтологій та словників. Виявлення цих ознак є мовно залежним[2].

До **статистичнихознак** відноситься обчислення частотності слова у документах і в корпусі документів його схожості з іншими словами.

Застосування **структурнихознак** обумовлено тим, що важливість терміна для даного документа залежить від його місця розташування. В якості чисельної структурної ознаки можна використовувати нормалізовану відстань між словом і початком документу (відношення довжини ланцюжка слів від початку документа до поточного слова, поділене на кількість слів у документі)[3].

Наразі однією з найпопулярніших та перевірених таких ознак є статистична міра **TF-IDF** [4]. Співвідношення TF до IDF – статистичний показник, який використовується переважно для оцінювання важливості конкретного слова(терміна) в контексті всього документа, що входить в загальну колекцію.

TF або частота слова – це відношення кількості входження конкретного терміна до сумарного набору слів в досліджуваному тексті (документі). Цей показник відображає важливість (вагомість) слова в рамках певної статті / публікації.

$$tf(t,d) = \frac{n_t}{\sum_k n_k},$$

Де n_t – число входжень слова t в документ, $\sum_k n_k$ – загальна кількість слів у документі.

IDF або зворотна частота документа – це інверсія частотності, з якої певне слово фігурує в колекції текстів (документів). Завдяки даним показникам можна знизити вагомість найбільш широко використовуваних слів (прийменників, сполучників, загальних термінів і понять).

$$idf(t,D) = log\frac{|D|}{|\{d_i \in D \mid t \in d_i\}|},$$

Де D – кількість документів у колекції, $|\{d_i \in D \mid t \in d_i\}|$ – кількість документів із колекції D, в яких зустрічається слово t.

Таким чином міра TF-IDF є добутком двох множників:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D)$$

Відповідно до TF IDF вагомість певного слова (терміна) прямо пропорційна до кількості разів його використання в конкретному тексті і обернено пропорційна від числа використання даного слова в інших документах.

4. Реферування тексту

Реферування або підсумовування – це процес скорочення набору даних для створення підмножини (резюме), що представляє найважливішу або релевантну інформацію в межах оригінального вмісту.

Існує два загальні підходи до автоматичного реферування: вилучення (екстрактивний підхід) та абстрагування (абстрактивний підхід)[9].

При **вилученні** вміст витягується з вихідних даних, але витягнутий вміст жодним чином не змінюється. Приклади вилученого вмісту включають ключові фрази, які можна використовувати для "тегування" або індексації текстового документа, або ключових позицій (включаючи заголовки), які в сукупності містять абстрактні, і репрезентативні зображення або відео сегменти.

Абстрактні методи будують внутрішнє смислове подання оригінального тексту, а потім використовують це уявлення для створення реферату, ближчого до того, що може виразити людина.

Абстракція працює краще, ніж вилучення. Однак алгоритми узагальнення тексту, необхідні для абстрагування, надзвичайно складні у розробці; тому використання вилучення все ще популярне.

Основними методами вилучення є ранжування і відсікання. Зазвичай застосовують один з двох підходів – або використання будь-яких евристичних формул, які дозволяють визначити, чи є слово ключовим, або використання методів машинного навчання. Варто зазначити, що для машинного навчання з учителем необхідний попередньо розмічений корпус документів з виділеними ключовими словами.

Латентний семантичний аналіз

Латентний семантичний аналіз (LSA)[5] – це алгебраїчно-статистичний метод, який виділяє приховані семантичні структури слів та речень. Це непідконтрольний підхід, який не потребує жодної підготовки чи додаткової інформації. LSA використовує контекст вхідного документа та витягує інформацію про те, які слова вживаються разом, а які загальні слова видно в різних реченнях. Велика кількість загальновживаних слів серед речень свідчить про те, що речення є семантично пов’язаними. Значення речення визначається словами, що містяться в ньому, а значення слів – з речень, які їх містять.

LSA має кілька обмежень. Перше – це те, що він не використовує інформацію про порядок слів, синтаксичні відношення та морфології. Цей вид інформації може бути необхідним для з’ясування значення слів та текстів. Друге обмеження полягає в тому, що він використовує не глобальні знання, а лише інформацію, яка існує у вхідному документі. Третє обмеження пов'язане з роботою алгоритму. З більшими та неоднорідними даними продуктивність різко знижується.

TextRank

TextRank[6] – це алгоритм ранжування на основі графів. Він працює на основі PageRank алгоритму.

PageRank[7] – сімейство алгоритмів оцінки важливості веб-сторінок за допомогою розв'язання систем лінійних рівнянь. Для кожної сторінки обчислює дійсне число, чим більше це число – тим «важливіша» сторінка.

Замість прямого підрахунку кількості посилань PageRank інтерпретує посилання сторінки А на сторінку Б як голос сторінки А на користь сторінки Б. Після цього PageRank оцінює рейтинг сторінки відповідно до кількості отриманих голосів.

PageRank також враховує значимість кожної сторінки, що отримала голос, адже голоси деяких сторінок є важливішими, і відповідно до цього підвищується значущість сторінки, посилання на яку вони містять. Важливі сторінки отримують більш високу оцінку PageRank і відображаються на перших позиціях результатів пошуку.

TextRank використовується для реферування одного документа чи набору документів. Підхід має обмеження, наприклад, використовується лише частота слів, що відкидає семантичні особливості тексту.

5. Аналіз існуючого програмного забезпечення

На даний момент існує досить багато програмних бібліотек для обробки природної мови. Найвідоміші з них:

- NLTK (NaturalLanguageToolKit) – найвідоміша NLP бібліотека, створена дослідниками в цій галузі. Надає інструменти для символьної та статистичної обробки природних мов англійською мовою, написаних мовою програмування Python. В силу того, що бібліотека повністю написана на Python вона доволі повільна та не підходить для обробки великих наборів даних. Також недоліком є відсутність підтримки українськовомних текстів;
- SpaCy – це вдосконалена бібліотека для обробки природних мов на Python та Cython. Він побудований на найновіших дослідженнях і був розроблений з першого дня для використання в реальних продуктах. Він постачається з попередньо навченими статистичними моделями та векторами слів, і в даний час підтримує токенізацію для більше 49 мов. Українська та російська мови заявлені в списку підтримуваних мов, але наразі значна частина функціоналу для цих мов не реалізована. Також суттєвим мінусом є непристосованість бібліотеки до масштабування та розподілених обчислень;
- ApacheOpenNLP – це практична та доступна бібліотека з відкритим кодом. Написана на мові Java і використовує бібліотеки Java NLP з декораторами Python. Надає рішення для

найпоширеніших завдань NLP: виявлення мови, токенізація, сегментація речень, позначення частини мови, виділення іменованих сутностей, шматування, синтаксичний аналіз.

- Бібліотеки CoreNLP, SparkNLP для технології ApacheSpark. Невід'ємним плюсом бібліотеки є повна підтримка бібліотеки Spark ML – однією з найпопулярніших на даний момент бібліотеки для машинного навчання технології ApacheSpark. Це означає легку масштабованість та підтримку розподілених обчислень. Проте бібліотека має і мінус попередників – відсутність підтримки для української мови (є лише підтримка російської мови). Існує підтримка мов програмування Python, Java та Scala.

За результатами наведеного аналізу можна зробити висновок, що не існує програмних бібліотек для одночасної обробки україномовних та російськомовних текстів з підтримкою високопродуктивних обчислень.

Розробка бібліотеки для ApacheSpark

Для реалізації обрано мову Python. Вимоги до бібліотеки:

- Підтримка RDD та потоків даних.
- Реалізація функцій попередньої обробки даних (токенізація, очистка, розбиття на речення, лематизація).
- Реалізація функцій реферування тексту.
- Реалізація функцій екстракції ознак TF-IDF та розрахунку косинуса подібності.

Нормалізація виконується за рахунок інтеграції з морфоаналізатором PyMorphy2[10]. Програмна архітектура бібліотеки[11] наведена на рис. 1.

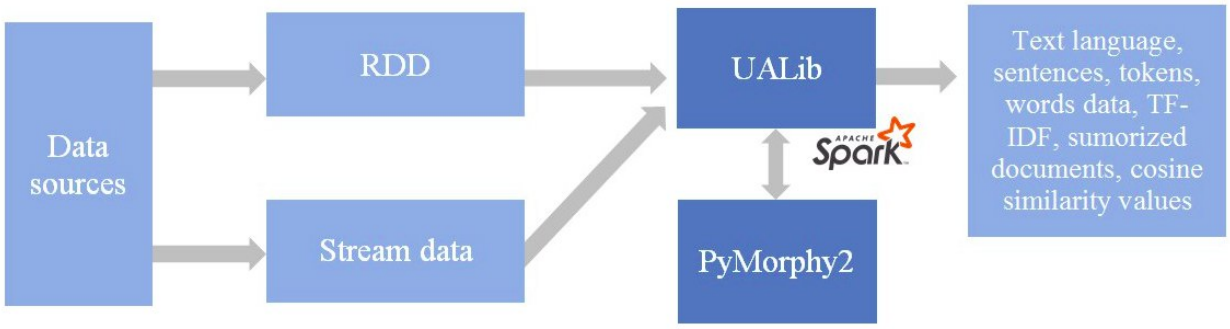


Рис. 1.Схема обробки текстових даних

Висновки

Постійне збільшення обсягу інформації призвело до необхідності розробки засобів автоматичної обробки неструктурованих текстових даних. В рамках даної статті було досліджено проблему автоматизації обробки текстових даних, описано етапи попередньої підготовки тексту, наведено рішення для виявлення ключових слів та вивчено підходи до реферування тексту. Також було розглянуто наявні технічні рішення, що дозволяють виконати обробку текстової інформації, та виявлено наступні такі недоліки як відсутність одночасної підтримки українськомовних та російськомовних документів та незастосовність до обробки надвеликих масивів інформації. Розроблено програмну бібліотекуобробки текстових даних для ApacheSpark з підтримкою обробки текстів українською та російською мовами.

Список літератури

1. Vishal Gupta , Gurpreet S. Lehal «A Survey of Text Mining Techniques and Applications» Journal of Emerging technologies in webintelligence, vol,1 no1 August 2009.
2. Kazi Saidul Hasan, VincentNg. Automatic Keyphrase Extraction: A Survey of the State of the Art. University of Texasat Dallas, Human Language Technology Research Institute.

3. Недільченко О. С. Етапи та методи автоматичного вилучення ключових слів / О. С. Недільченко. – журнал «Молодий вчений». – 2017. – № 22. – с. 60-62.
4. Shahzad Qaiser, Ramsha Ali Text Mining: Useof TF-IDF to Examine the Relevance of Words to Documents, International Journal of Computer ApplicationsVolume 181 – No.1, July 2018
5. J. Steinbergerand K. Jezek. Using Latent Semantic Analysis in Text Summarization and Summary Evaluation// Proc. of ISIM. – 2004. – С. 93–100.
6. Rada Mihalcea, Paul Tarau. Text Rank: Bringing Orderinto Texts, University of North Texas, 2004. Режим доступу: http://www.aclweb.org/anthology/W04–3252
7. Nan Maa, Jiancheng Guan, YiZhaoc. Bringing Page Rankto the citation analysis. // Information Processing&Management. – 2008. – №44. – p. 800-810.
8. Гавриленко О.В., Олійник Ю. О., Ханько Г. В. Огляд та аналіз алгоритмів TEXT MINING. Управління проектами, системний аналіз і логістика. 2017. Випуск 19. Частина 1. С. 32-41.
9. Аршакян Г.Д. Олійник Ю.О. Огляд підходів та методів автоматичного реферування тексту. Інформаційні системи та технології управління: матеріали всеукр. наук.-практ. конф. молодих вчених та студентів, (м. Київ, 26 лист. 2019 р.). Київ, 2019. С. 44-48.
10. Korobov, M. (2015, April). Morphological analyze rand generator for Russian and Ukrainian languages. In International Conference on Analysis of Images, Social Networks and Texts (pp. 320-332). Springer, Cham.
11. Ukrainian NLP Library for Apache Spark.URL: https://github.com/oliyura/UANLP/ [Назва з екрана]

УДК 004.02

ТЕРЕЩЕНКО А.С.,
ОЛІЙНИК Ю.О.

АНАЛІЗ ЕКГ СИГНАЛІВ ЗА ДОПОМОГОЮ МОДЕЛІ WORD2VEC

В цій статті розглянуто задачуаналізу ЕКГ за допомогою моделі Word2Vec, яка створена на основі окремих хвиль ЕКГ.Сформульовано цілі та актуальність дослідження. Проаналізовано підхід до вирішення даної задачі, розглянуто основні етапи попередньої обробки ЕКГ сигналів, наведено приклади обробки ЕКГ.
КЛЮЧОВІ СЛОВА: ЕКГ, модель Word2Vec, векторне представлення слів, методи обробки природньої мови

This article is about ECG analysis problem using the Word2Vec model, which is based on extracted ECG waves. The goals and relevance of the research are formulated. The approach to the decision of the given problem is analyzed, the basic stages of preliminary processing of an ECG of signals are considered, examples of processing of an ECG are resulted.
KEYWORDS: ECG, Word2Vec model, word embedding, NLP methods

1. Задача аналізу ЕКГ

Серцеві захворювання займають значний відсоток серед причин смертності як в Україні так і в більшості країн світу. Для прикладу щороку в Україні понад 68% осіб помирають через серцево-судинні хвороби. Важливим фактором в боротьбі з хворобою є профілактика та виявлення захворювання на ранніх стадіях. Безперечно чим раніше виявляються серцево-судинні захворювання, тим ефективнішим буде лікування. Одним із основних методів діагностики серця є електрокардіографія, тому дуже важливо швидко та точно зробити аналіз електрокардіограми (ЕКГ).

З розвитком машинного навчання вдалося автоматизувати досить складні процеси: розпізнавання об’єктів, прогнозування подій, аналіз різноманітних даних. Було б корисно створити модель, яка змогла аналізувати ЕКГ та виявляти чи прогнозувати захворювання.

Аналіз існуючих підходів до вирішення подібних проблем допомагає більш детально дослідити предметну область та спробувати створити удосконалену модель для

покращення швидкості та точності виявлення тих чи інших серцевих захворювань. Одним з таких підходів може стати розгляд ЕКГ сигналу аналогічно до речення в мові.

Подібно до природних мов сигнал ЕКГ складається з послідовностей з трьох або чотирьох різних хвиль, включаючи P-хвилю, комплекс QRS, T-хвилю та U-хвилю [1]. ЕКГ сигнал є послідовністю серцебиття (подібно до речень в природній мові), і кожне серцебиття складається з сукупності хвиль (подібних до слів у реченні) різної морфології.

Аналогічно обробці природної мови (NLP), яка використовується, щоб допомогти комп'ютерам зрозуміти та інтерпретувати природну мову людини, можна розробити методи NLP, що допоможуть комп'ютерам глибше зрозуміти сигнали електрокардіограми. Техніка аналізу ЕКГ, зосереджена на розширенні можливостей комп'ютерів розуміти сигнали ЕКГ в так, як це роблять лікарі.

2.Лінгвістичний метод представлення ЕКГ

Основною концепцією даного методу [2] є співвідношення числового проміжку до певної літери. Довжина числового проміжку може бути підібрана різною відповідно до розподілу, обраного символьного алфавіту та самого діапазону значень числового ряду.

При використанні лінгвістичного підходу варто зазначити необхідність отримання початкового промаркованого набору з аномаліями для їх пошуку в подальшому. При пошуку аномалій в справжніх даних використовуються алгоритми для пошуку відстані між рядками : відстань Левеншейна ,відстань Геммінга, Джаро-Вінклера , Дамерау-Левенштейна.

Недоліком даного методу є представлення хвилі через лінгвістичний ланцюжок. Але так як хвилі одного типу мають різні сигнали то і різні лінгвістичні ланцюжки, що ускладнює їх обробку.

Етапи обробки ЕКГ сигналу

Опишемо основні етапи обробки мови ЕКГ[3].

1. Виявлення піків - етап включає виявлення R-піків поточного сигналу ЕКГ або виявлення комплексів хвиль QRS. Для цієї мети використовуються один з алгоритмів Пана – Томпкінса [4]. Червоними колами на рисунку 1зображені R-зубці ЕКГ-сигналу.
2. Розбиття та сегментаці хвиль - етап включає поділ безперервного ЕКГ-сигналу на певну послідовність серцебиттів та розбиття кожного серцебиття на окремі одиниці, які називаються хвилями. Після виявлення R-хвиль, наявність інших складових хвиль (тобто P, QRS і хвилі T) в ЕКГ-сигналі можуть бути вилучені за допомогою адаптивних вікон пошуку. Для виконаннясегментації сигналу на серцебиття ідентифікуємоодин сегмент як фіксовану кількість екземплярів зняття сигналу до розташування піку R та до фіксованої кількості екземплярів після розташування піку R або від початку хвилі P до зміщення послідовної хвилі T. На малюнку рисунку 2 зображена сегментований ЕКГ сигнал, розфарбований R-хвилями, P, QRS і T-хвилями.
3. Створення словника - етап включає побудову словника хвиль на основі вилучених хвиль з ЕКГ сигналів. Шляхом групування хвиль формуємо середнє значення кожної групи як вхід до словника. Це може бути здійсненошляхом подачі всіх хвиль на вхідалгоритмівкластеризації, такихяк K-середніх, спектральна кластеризація або агломеративні алгоритми кластеризації [5,6]. Після кластеризації хвиль середнє значення кожного кластера може представляти окрему хвилю словникового запасу. На рисунку3 зображено кластеризацію набору даних ЕКГ-сигналуз використанням техніки t-розподіленого стохастичного сусідства (t-SNE) [7].
4. Розподілення хвиль - процес сегментації розбитих хвиль створює послідовність хвиль для кожного сигналу ЕКГ. Потім, кластер кожної хвилі в послідовності ідентифікується за допомогою виходу попереднього кроку (тобто попереднього етапу конвеєра). Іншими словами, він присвоює унікальнийсимвол (який відповідає

певномукластеру) кожній хвилі в послідовності. Таким чином, кожен ЕКГ сигнал кодується послідовністю символів так, що кожнийсимвол представляє певну хвилю (або кластер) у словниковому запасі.

5. Вбудування та векторизація хвиль.На цьому етапі приймаємо кодований словник виділенних хвиль і будуємо вектор вбудовування (тобто вектор заданої довжини) для кожної хвилі словникового запасу. Основна причина вбудовування слів полягає в тому, що це дозволяє нам застосовувати вдосконалену модель, що спирається як на штучні нейронні мережі так і на цілочисельні кодовані сигнали ЕКГ для конкретного завдання. За допомогою NLP можемо використовувати кілька підходів, такі як граф векторизатор, у якому послідовність хвиль перетворюється у вектор фіксованої довжини із розміром словникового запасу. Значення в кожному положенні у векторі буде підрахунком кожної хвилі в закодованому сигналі, або підхід Word2Vec, який використовує нейромережеві методи для представляють хвилі у векторному просторі. Останній підхід є більш ефективним, оскільки він розпізнає контекст, відношення та подібність між хвилями [8].

6. Тренування та тестування результатів - етап передбачає використання методів машинного навчання та глибокого навчання для створених моделей, які зможуть виконувати будь-які навчальні завдання, включаючи класифікацію, прогнозування тощо.

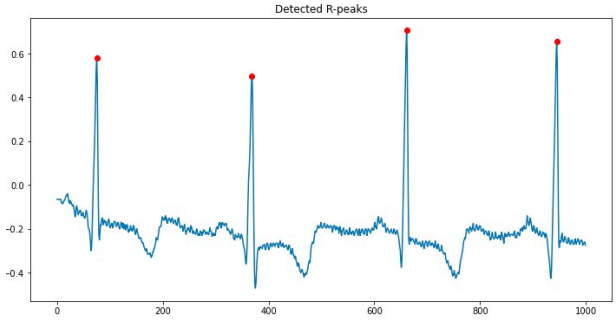


Рис.1.Виявлення R піків в кожному серцебитті

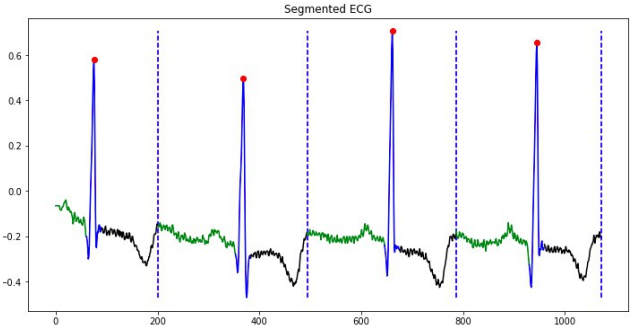


Рис.2. Сегментація хвиль в ЕКГ сигналі

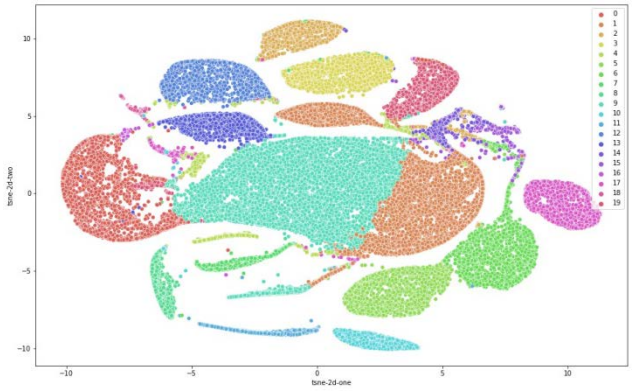


Рис.3. Візуалізація результату кластеризації вилучених хвиль з ЕКГ

4. Побудова моделі Word2Vec

Word2Vec - це техніка для обробки природної мови. Алгоритм word2vec [9] використовує модель нейронної мережі для вивчення асоціацій слів із великого корпусу тексту. Після навчання така модель може виявляти слова синоніми або пропонувати додаткові слова для часткового речення. Як випливає з назви, word2vec зіставляє кожне окреме слово з певним списком чисел, який називається вектором. Вектори вибираються ретельно таким чином, щоб проста математична функція (косинусова подібність між векторами) вказувала на рівень семантичної подібності між словами, представленими цими векторами.

Даний алгоритм реалізує дві основні архітектури – ContinuousBagofWords (CBOW) і Skip-gram. Обидві архітектури володіють власними перевагами. наприклад, підхід skip-gram найбільш ефективний при обробці невеликого корпусу навчальних даних. Більш того, за допомогою нього добре описуються слова, що рідко зустрічаються. З іншого боку, CBOW працює швидше і краще

обробляє часто вживані слова. Однак навчання вихідного вектора алгоритмів CBOW і skip-gram є одним з найбільш великих обмежень цих моделей, так як це може бути важким завданням, що вимагає великих обчислювальних витрат. Для вирішення цієї проблеми можна використовувати два алгоритми.

Перший - негативний семплінг. Основна ідея алгоритму полягає в обмеженні кількості вихідних векторів, які потрібно оновлювати. Таким чином, тільки деякі вектори оновлюються випадковим чином. Цей розподіл шуму є імовірнісним і використовується в процесі семплінгу.

Другий алгоритм - ієрархічний softmax. Він заснований на дереві Хаффмана. Фактично, це двійкове дерево, яке представляє всі терміни на основі їх частоти появи в корпусі тексту. Потім кожен крок від кореня до мети нормалізується. Кожен алгоритм має переваги в порівняно з іншим в залежності від навчальних даних. Наприклад, негативний семплінг більш ефективно працює з векторами малої розмірності і часто вживаними словами. Проте, ієрархічний softmax показує себе краще в роботі з рідко вживаними словами.

В контексті переведення сигналу ЕКГ в модель Word2Vec відбувається представлення слів (хвиль серцебиття) у векторній структурі. При цьому враховується близькість різних тактів серцебиття.

CBOW

ContinuousBagofWords (CBOW) дозволяє передбачати поточне слово, ґрунтуючись на його контексті, який визначається сусідніми словами у вікні. У CBOW використовуються три шари. Вхідний шар відповідає контексту. Прихований шар - проекції кожного слова з вхідного шару в вагову матрицю, яка проектується в третій вихідний рівень. Останнім етапом моделі є порівняння її виведення з самим словом, щоб скорегувати його уявлення, засноване на зворотному поширенні градієнта помилки. Тому, метою нейронної мережі CBOW є максимізація рівняння показаного формулою:

$$\frac{1}{V}\sum_{t=1}^{V}\log p\left(m_t\,|m_{t-\frac{c}{2}}\dots m_{t+\frac{c}{2}}\right),$$

де V - об'єм словника, c - розмір вікна для кожного слова.

Skip-gram

Метод skip-gram вирішує обернену задачу: на підставі одного слова передбачається контекст. Останній крок алгоритму - порівняння виведення з кожним словом в контексті з метою коригування уявлення, заснованого на зворотному поширенні градієнта помилки. даний метод виконує максимізацію наступного рівняння:

$$\frac{1}{V}\sum_{t=1}^{V}\sum_{j=t-c,j\neq 1}^{t+c}\log p(m_j|m_t),$$

де V - об'єм словника, c - розмір вікна для кожного слова.

В контексті нашого дослідження модель Word2Vecдозволить використати методиNLP для аналізу ЕКГ, наприклад, класифікації серцевиххворобчи прогнозування аритміїза допомогою визначення подібностісерцебиттів, шляхом навчання моделей розміченими даними.

Результати досліджень

У результаті обробки сигналу ЕКГ було виділено піки методом Пана – Томпкінса та проведено розбиття на хвилі за типами. Методом k-meansхвилі були розділені на 20 кластерів, де у відповідність кожному кластеру проставлено символ. Таким чином серцебиття можливо представити в вигляді слів, що складаються з символів, яківідповідають певним кластерам. На основі перетвореного сигналу ЕКГ в слова побудовано модель word2vec. Використання даної моделі відкриває шлях до використання методів машинного навчання для аналізу ЕКГ.

Висновки

Отже, в даній статті розглянутовирішення задачі аналізу ЕКГ за допомогою моделі Word2Vec. Проведено процес розбиття ЕКГ сигналу на окремі серцебиття та їх сегментацію різними типами хвиль. Проведено кластеризацію виділених хвиль та створення словника, в якому хвилі групуються відповідно до кластера якому вони належать. За допомогою лінгвістичного методу переведено кожну хвилю зі словника в окремий символ, тим самим

кожне серцебиття представлено у вигляді слова, а ЕКГ сигнал у вигляді речення. Переведенні слова подаються на вхід Word2Vec моделі для її тренування одним із методівCBOW, Skip-gram. Навчену модель можна використовувати для виявлення рівня семантичної подібності між серцебиттями (словами), представленими цими векторами, а також аналізу ЕКГ методами NLP.

Список літератури

1. J. W. Hurst, “Naming of the waves in the ecg, with a brief account of their genesis,” Circulation, vol. 98, no. 18, pp. 1937–1942, 1998.
2. Baklan I., Mukha I., Oliinyk Y., Lishchuk K., Nedashkivsky E., Gavrilenko O. (2020) Anomalies Detection Approach in Electrocardiogram Analysis Using Linguistic Modeling. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. Springer, Cham; pp 513-522, DOI - https://dx.doi.org/10.1007/978-3-030-16621-2_48; (Scopus)
3. Sajad Mousavi, Fatemeh Afghah, Fatemeh Khadem, and U. Rajendra Acharya ECG Language Processing (ELP): New Technique to Analyze ECG Signals, 2020, pp. 2-4
4. J. Pan and W. J. Tompkins, “A real-time qrs detection algorithm,” IEEE Trans. Biomed. Eng, vol. 32, no. 3, pp. 230–236, 1985.
5. T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 7, pp. 881–892, 2002.
6. A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in Advances in neural information processing systems, 2002,pp. 849–856
7. L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” Journal of machine learning research, vol. 9, no. Nov, pp. 2579–2605, 2008.
8. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Advances in neural information processing systems, 2013, pp. 3111–3119.
9. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // Proc. of Workshop at ICLR. 2013. P. 1301-3781

УДК 004.031.43

КАС’ЯНЧУК А.С.

ІНФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА ДЛЯ ПІДБОРУ КРЕДИТНИХ ПРОПОЗИЦІЙ КЛІЄНТАМ ТОРГОВЕЛЬНИХ МЕРЕЖ

В роботі розглянуто модель продажі кредитних продуктів в торгових мережах, без безпосередньої участі, використовуючи інформаційні технології. Описана схема роботи інформаційної системи там методи прогнозування відповідей від банків на базі особистої скорингової системи.

КЛЮЧОВІ СЛОВА: БРОКЕР, ПРОГНОЗУВАННЯ, СКОРИНГ, РЕГРЕСІЯ, КРЕДИТНИЙ ПРОДУКТ.

This article discusses the model of selling credit products in retail chains, without direct participation, using information technology. The scheme of work of information system is described there methods of forecasting of answers from banks on the basis of personal scoring system.

KEYWORDS: BROKER, FORECASTING, SCORING, REGRESSION, CREDIT PRODUCT.

1. Вступ

На сьогодні на ринку банківських послуг дуже велика конкуренція, а вимоги клієнтів тільки ростуть. Споживачі, перед тим, як обирати кредитний продукт, хочуть порівняти між собою усі пропозиції на ринку та обрати найкращий серед запропонованих кредитних продуктів [1]. Це зробити досить складно, потрібно відвідати кожен банк або переглянути відповідний веб-сайт банку, щоб зрозуміти, що він може запропонувати.

Особливо гостро ця проблема стоїть в сфері продажу товарів та послуг під споживчий кредит або так звану розстрочку. В Україні та у світі дуже мало торгових мереж, які можуть особисто видавати такі споживчі кредити. Загалом, магазини, в таких випадках, користуються послугами банків, проте без налагодженої інфраструктури, співпрацювати з більше ніж одним банком дуже складно і через це, з часом, обраний банк стає монополістом і починає завищувати процентні ставки на споживчий кредит. В такому випадку магазини починають втрачати потенційних клієнтів і гроші.

Також є проблема швидкості видачі продукті. Клієнт хоче прийти в магазин і максимум за 30 хвилин отримати кредитний продукт на купівлю обраного товару без відвідування банку. Модель збільшить кількість клієнтів і пропускну спроможність магазину.

У даному дослідженні розглядаються модель продажі кредитних продуктів у автоматичному режимі, без безпосередньої участі банку, використовуючи інформаційні технології. Пропонується створити інформаційну систему, яка дозволяє переглядати запропоновані кредитні продукти від усіх підключених до системи банків в одному місці. Система не тільки підбиратиме кредитні продукти для клієнта, а і в онлайн режимі надсилатиме запити на отримання кредиту в усі банки по вибраним кредитним продуктам.

2. Система для підбору кредитних пропозицій клієнтам торговельних мереж

Як було сказано вище, ринок потребує інформаційну систему-конфігуратор, яка б об’єднувала у себе всі банки.

Модель передбачає, що клієнт особисто або за допомогою продавця на торговій точці, заповнить анкету з необхідними даними для банків. Анкету передбачається поділити на наступні кроки: основні дані, контакти, фінанси, робота. Після заповнення анкети, внутрішня скорингова система, на базі попередніх відповідей, аналізує анкету і надає клієнту прогнози на те, отримає клієнт кредит чи ні. Після аналізу, якщо система поверне негативну відповідь, буде виведено на екран поля в анкеті, після зміни яких, можливо буде отримати позитивну відповідь [2].

Після отримання прогнозу від внутрішньої скорингової системи, анкета відправляється в банк, для аналізу банківською скоринговою системою. Протягом декількох хвилин банк повертає в систему результат роботи зовнішньої скорингової системи. Схема роботи зображена на рисунку 1.

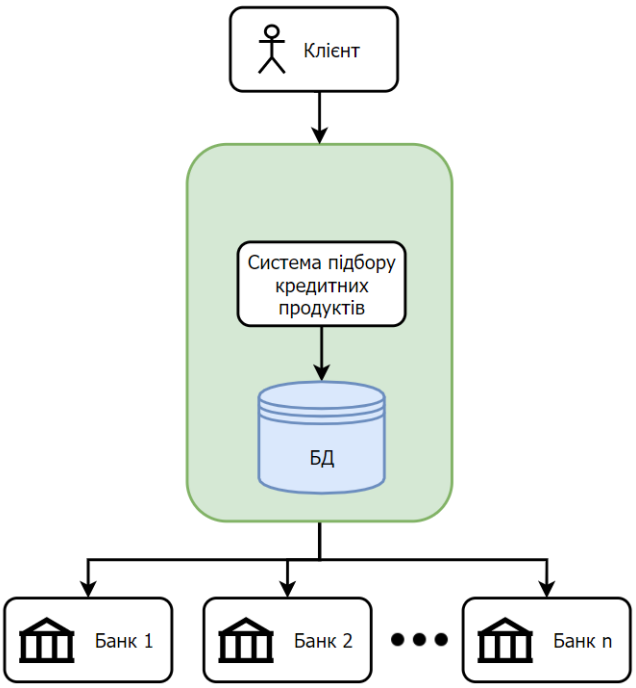


Рис 1. Схема роботи системи

У відповіді очікується результат (так або ні) та при позитивній відповіді множина кредитних продуктів.

Після отримання результатів, клієнту потрібно обрати банк, серед тих, що дали позитивну відповідь та обрати кредитний

продукт. По кожному кредитному продукту буде представлена повна аналітика та буде рекомендовано найвигідніший продукт.

Вибравши кредитний продукт, система надсилає в банк відповідь, після чого система банку генерує пакет документів та надсилає в внутрішню систему. Клієнт підписує угоду, після чого в систему банку надсилається повідомлення про фіналізацію угоди.

3. Постановка задачі прогнозування результатів скорингових систем

Система підбору кредитних продуктів працює в тісній інтеграції зі скоринговими системами банків. Тому, було б зручно, в системі використовувати прогнозування результатів скорингових систем банків, для підвищення ймовірності отримати позитивну відповідь. Система прогнозування для кожного банку буде оцінювати ймовірність надання позитивної відповіді.

Для вирішення задачі прогнозування відповіді від скорингової системи можна існує декілька методів.

На базі накопичених заявок і відповідей можна побудувати дерево рішень. І завдяки дереву рішень отримати прогнозований результат. Недоліком такого рішення є те, що дерево в результаті повертає бінарне рішення (так або ні).

Іншим способом є побудова власної скорингової моделі. Щоб сформувати відповідь клієнту, банки використовують скорингові моделі для оцінки платоспроможності клієнта позичальника. За допомогою скорингу можна отримати класифікаційну статистично-математичну модель по різним групам [3].

Під кожен банк буде будуватись своя внутрішня скорингова модель.

Для реалізації скорингової моделі використовують різні моделі. На ринку кредитування найчастіше використовуються наступні методи: лінійний регресійний аналіз, логістичний регресивний аналіз, нейронні мережі та дерева рішень.

Для вирішення даної задачі було вирішено використовувати скорингові карти на базі логістичної регресії.

Скорингова карта – це пул ознак клієнта-позичальника (стать, вік, дохід, рухоме і нерухоме майно і т.д.). На базі вказаних в анкеті даних, формується скоринг бал. І порівнюючи цей скоринг бал, банк дає рішення.

Скоринг карта підходить, як і під бінарні ознаки (стать), так і значимих ознак (дохід).

4. Алгоритм прогнозування результатів скорингових систем

Поділимо алгоритм на два етапи.

Eman 1

Основною задачею побудови скорингової таблиці є оцінка коефіцієнтів значущості ознак клієнтів. Підсумкову оцінку скорингу можна підрахувати за наступною формулою[4]:

$$Score = k_1 x_1 + k_2 x_2 + \cdots + k_i x_i$$

де k_i – ваговий коефіцієнт логістичної регресі і-ї ознаки клієнта;

x_i- значення *i*-ї ознаки.

Зі всього масиву даних про клієнтів, потрібно обрати найбільш значущі ознаки. Зазвичай, оцінку ознаки визначають за допомогою WOE (Weight of Evidence).

За своєю суттю метод вагомості ознаки представляє статистичний метод, заснований на теоремі Байеса [5].

Цей показник розраховується як:

$$\boldsymbol{WOE} = \ln(\frac{\boldsymbol{d_i^{(1)}}}{\boldsymbol{d_i^{(2)}}})$$

де $\boldsymbol{d_i^{(1)}}$ кількість або частка погоджених запитів на кредит по і-й ознаці клієнта;

$\boldsymbol{d_i^{(2)}}$ кількість або частка не погоджених запитів на кредит по і-й ознаці клієнта.

Тепер нехай деяка номінальна ознак під номером і приймає значення *{1,...,n}.* Для цього потрібно розрахувати *IV (Information Value)* за формулою:

$$\boldsymbol{IV} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{k}} \left(\boldsymbol{d_i^{(1)}} - \boldsymbol{d_i^{(2)}}\right) * \boldsymbol{WOE_i},$$

Відповідно до значення *IV* оцінюється значущість ознаки, чим більше значення, тим більш значимим є поле.

За допомогою *WOE* оцінюються категоріальні ознаки (наприклад тип доходу).

В внутрішній скоринговій моделі показник *IV* буде використовуватись як ваговий коефіцієнт k_i.

Етап 2

Оцінивши скоринг бал для кожного банку, клієнт може прийняти рішення відправити заявку на банки або, якщо клієнта не влаштовує прогнозуючий скоринг бал, то він може відредагувати заявку і повторити виконання алгоритму за *етапу 1*.

5. Збір даних

Основним етапом побудови скорингової моделі є накопичення анкетних даних клієнтів та відповіді банків. Точність прогнозу повністю залежить від чистоти даних, тому до цього етапу потрібно підійти відповідально. Потрібно використовувати надійні джерела для побудови початкової прогнозуючої моделі.

Дані мають бути як мінімум за останній рік, так як скорингові моделі банків постійно змінюються, а клієнту потрібно отримати актуальний прогноз.

6. Аналіз результатів

Згенеруємо чотири анкети з різними анкетними даними для аналізу роботи моделі. В таблиці 1 представлено деякі анкетні дані.

В таблиці 2 представлені результати оцінки параметрів клієнтів. В останньому рядку для кожного з них описаний скоринг бал, сформований за допомогою побудованої скоринг моделі.

Таблиця 1 – Анкетні дані клієнтів

Ознака	***Анкета 1***	***Анкета 2***	***Анкета 3***	***Анкета 4***
Прострочені платежі по кредитах за рік	Ні	Ні	Так	Ні
Відсутність пропуску платежів	Так	Так	Ні	Так
Дохід	32000	7000	16000	8000
Стаж роботи	6 років	1 рік	17 років	Не працює
Вік	27	19	45	67
Освіта	Вища	Незакінчена вища	Вища	Середня спеціальна
Є діти?	Ні	Ні	Так	Ні

Таблиця 2 – Категоризовані дані

Ознака	***Вага ознаки***	***Анкета 1***	***Анкета 2***	***Анкета 3***	***Анкета 4***
Прострочені платежі	-0,8	0	0	1	0
Відсутність пропуску платежів за рік	0,7	1	1	0	1
Дохід	1	1	0,3	0,7	0,3
Стаж роботи	0,5	0,5	0,3	1	0
Вік	0,7	0,7	0,2	1	0,1
Освіта	0,7	1	0,3	1	0,5
Є діти?	-0,5	0	0	1	0
Скоринг бал		3,14	1,5	1,3	1,42

Для *i*-го банку усередненим скоринг балом, при якому банк дає позитивну відповідь є ***2,55***.

Візьмемо Анкету 1 для прикладу і проведемо аналіз результатів *m* скорингових моделей для *m* банків. В таблиці 4 наведено результати скорингу для 3-ьох банків. Як видно із скоринг балу, найбільші шанси отримати кредит від банку під номером 2, а найменші від банку під номером 3.

Таблиця 4 – Анкетні дані клієнта в розрізі 3-ьох банків

Ознака	***x***	***Вага ознаки банку 1***	***Вага ознаки банку 2***	***Вага ознаки банку 3***
Прострочені платежі	0	-0,8	-0,5	-0,8
Відсутність пропуску платежів за рік	1	0,7	0,9	0,5
Дохід	1	1	1,2	0,8
Стаж роботи	0,5	0,5	0,5	0,6
Вік	0,7	0,7	0,6	0,7
Освіта	1	0,7	0,6	0,6
Є діти?	0	-0,5	-0,2	-0,4
Скоринг бал		3,14	3,37	2,69

Висновки

В рамках системи представлена реалізована ідея використання інформаційно-пошукової системи підбору кредитних продуктів клієнтам торгових мереж. В процесу роботу над системою було розглянуто основні підходи для прогнозування відповідей банків на отримання кредиту, використовуючи побудовану скорингову модель.

Список літератури

1. Риков І. Н. Ринок нових кредитних продуктів: проблеми і перспективи, – М.: Фінанси і кредит, 2007. – с. 1-4
2. Звєрєв О.А. Система продаж банківських продуктів як невід’ємна елемент ринкового механізму в банківській сфері. – М.: Фінанси і кредит, 2004. – С. 1-5.
3. Сорокін А.С Побудова скорингових моделей з використанням логістичної регресії, - М: Науковедение - 2014, - с. 1-4
4. Самойлова С.С., Курочка М.А. Скорингові моделі оцінки кредитного ризик, - М: Социально экономические явления и процессы - 2012, - с. 1-8
5. Лагарєв Д.Г., Бондарєва И.В. Особливості побудови скорингової моделі на основі аналітичної платформи Deductor, - М: Научно-технологический вестник Брянского государственного университета, №1 - 2014, - с. 1-4

УДК 004.042

ШУЛІКОВ Д.Д.

ВИБІР ПЛАТФОРМИ ДЛЯ ОБРОБКИ ПОТОКОВИХ ДАНИХЗ СЕНСОРІВ МОРСЬКИХ СУДЕН

У цій статі я розповім про типи та аспекти обробки потоків даних загалом, а потім порівняю найпопулярніші платформи з відкритим кодом. Я спробую пояснити як вони працюють, випадки їх використання, сильні сторони, обмеження, подібності та відмінності.

SPARK STREAMING,APACHEFLINK,APACHE STORM,KAFKA STREAMS

I will describe the types and aspects of streaming processing in general, then compare the most popular open source platforms. I will try to explain how they work, cases of their use, strengths, limitations, similarities and differences.

SPARK STREAMING,APACHEFLINK,APACHE STORM,KAFKA STREAMS

1.Вступ

Згідно з нещодавнім звітом IBM, «Дев'яносто відсотків даних у світі було створено лише за останні два роки. З появою нових пристроїв, датчиків та технологій, зростання даних прискориться ще більше».

Технічно це означає, що обробка великих масивів даних стане складнішою. Багато випадків (наприклад, оголошення для мобільних додатків, виявлення шахрайства, бронювання, моніторинг пацієнтів тощо) потребують обробки даних у режимі реального часу для швидкого прийняття рішень. Ось чому розподілена потокова обробка стала дуже популярною у світі великих даних.

Сьогодні доступна низка фреймворків для потокової обробки з відкритим кодом. Цікаво, що майже всі вони досить нові і були розроблені лише за останні кілька років, тож новачку досить легко заплутатися в розумінні цих платформ.

2. Обробка потокових даних

Найелегантніше визначення, яке я знайшов, це "тип механізму обробки даних, який розроблений з урахуванням нескінченних наборів даних".

На відміну від пакетної обробки, де дані обмежуються початком і кінцем у роботі, а робота закінчується після обробки цих даних - потокова обробка призначена для опрацювання необмежених даних, що надходять у режимі реального часу, безперервно протягом днів, місяців та років.

Існує кілька важливих характеристик та термінів, пов’язаних із обробкою потоку, про які ми повинні знати, щоб зрозуміти сильні сторони та обмеження будь-якої структури потоку:

- Гарантія доставки (Delivery Guarantees). Це означає гарантію того, що незважаючи ні на що, певний вхідний запис у потоковому механізмі буде оброблений. Це може бути як atleast-once (буде оброблено принаймні один раз, навіть у разі відмов), atmost-once (може бутине оброблено у разі відмов) або exactly-once (буде оброблено один і рівно один раз, навіть у випадку відмов). Очевидно, що режим exactly-once найбажаніший, але його важко досягти в розподілених системах, потрібно знаходити компроміси з продуктивністю.

- Відмовостійкість (Fault Tolerance). У разі відмов, таких як відмови вузлів, відмови мережі тощо, фреймворк повинен мати можливість відновлення та починати обробку знову з того місця, де він зупинився. Це досягається за допомогою регулярногозбереження стану потокової обробки на постійне сховище.

- Управління станом (State Management). У випадку вимог до обробки даних, де нам потрібно підтримувати деякий стан застосунку (наприклад, кількість входжень кожного окремого слова із записів), фреймворк повинен мати можливість забезпечити якийсь механізм збереження та оновлення інформації про стан.

- Продуктивність (Performance) включає затримку (як швидко запис може бути оброблений), пропускну здатність (оброблені записи / секунду) та масштабованість. Затримка повинна бути якомога меншою,

тоді як пропускна здатність повинна бути якомога більшою. Важко отримати обидва показники одночасно.

- Розширені функції. Функції, якінеобхідні, якщо вимоги до обробкипотоку є складними. Наприклад, обробка записів на основі часу, який був сформований у джерелі та агрегація даних у часових вікнах.

- Зрілість. Важливо, якщо інструмент вже перевірений та випробуваний великими компаніями. Більше шансів отримати хорошу підтримку у мережі та допомогу на stackoverflow.

3. Два типи обробки потоків даних

Знаючи терміни, які щойно були описані, тепер можна зрозуміти, що існує два підходи до впровадження обробки потоків даних:

Нативний стрімінг - кожен вхідний запис обробляється як тільки він надходить, не чекаючи інших. Є кілька безперервно запущених процесів, які ми називаємо операторами, і кожен запис проходить через ці процеси для обробки. Приклади: Storm, Flink, Kafka Streams, Samza.

Мікробатчінг - означає, що вхідні записи через кожні кілька секунд групуються, а потім обробляються в одній міні-партії із затримкою в кілька секунд. Приклади: Spark Streaming, Storm-Trident.

Обидва підходи мають деякі переваги та недоліки:

Нативний стрімінгє більш природним, оскільки кожен запис обробляється відразу після надходження, що дозволяє фреймворку досягти мінімально можливої затримки. Але це також означає, що важко досягти відмовостійкості без шкоди для пропускної здатності, оскільки для кожного запису нам потрібно зберігати контрольну точку після обробки. Крім того, управління станом є простим, оскільки існують тривалі процеси, які можуть легко підтримувати необхідний стан.

Мікробатчінг навпаки, є цілком протилежним. Толерантність до несправностей надається безкоштовно, оскільки це, по суті, пакет.Пропускна здатність також велика, оскільки обробка та перевірка буде виконуватися одним пострілом для групи записів. Але це буде відбуватись з деякоюзатримкою і це не буде природний потік даних.

4. Фреймворки для потокової обробки даних

- ApacheStorm - найстаріший фреймворк для потокової обробки з відкритим кодом і один із найбільш зрілих та надійних. Це нативна потоковаобробка. Підходить для простих випадків прийняття рішень на основі подій.

- SparkStreaming - Spark виявився справжнім спадкоємцем hadoop в пакетній обробці та першим фреймворком, який повністю підтримує лямбда архітектуру (де використана як пакетна, так і потокова обробка; пакетна для коректності, потокова для швидкості). Цей фреймворкдуже популярний, зрілий і широко прийнятий. Spark Streaming безкоштовно надається разом із Spark і використовує мікробатчінг для потокової обробки. До випуску 2.0 Spark Streaming мав деякі серйозні обмеження продуктивності, але з новим випуском 2.0+ він називається структурованою потоковоюобробкою і оснащений багатьма додатковими функціями, такими як користувацьке управління пам'яттю (аналог flink), водяними знаками, підтримкою обробки часу подій тощо. Також структурований потік є набагато абстрактнішим, і є можливість переключення між мікро-пакетним режимом та режимом безперервної потокової обробки у випуску 2.3.0. Режим безперервного потокової обробки обіцяє зменшити затримку у сторону Storm та Flink, але він все ще перебуває в початковій стадії розробки з багатьма обмеженнями в операціях.

- Fink,як і Спарк, також має академічний бекграунд. Спарк прийшов з UC Berkley, Флінк - з Берлінського університету TU. Як і Spark, він також підтримує лямбда-архітектуру.Але реалізація повністю протилежна реалізації Spark.Flink - це, по суті, справжній потоковийпроцесінг. Хоча API в обох фреймворках схожі, але вони не мають жодної подібності в реалізаціях. У Flink кожна функція, така як map, filter, reduce тощо реалізована як тривалий оператор (аналогічно Storm).

- KafkaStreams на відміну від інших фреймворків, -невелика бібліотека для потокової обробки даних KafkatoKafka. Немає відповідності з точки зору

продуктивності з Flink, але також не потребує окремого кластера для запуску. Дуже зручний і простий фреймворк у розгортанні та початку роботи.Однією з головних переваг KafkaStreams є те, що обробка повідомлень відбувається чітко один раз (exactlyonce). Це можливо, оскільки джерелом, а також місцем призначення є Kafka, а з версії Kafka 0.11, випущеної приблизно в червні 2017 року, підтримується exactlyoncесемантика. Щоб увімкнути цю функцію, нам просто потрібно увімкнути флаг у налаштуваннях.

- Samza схожа на Kafka Streams. Є багато подібностей. Обидва ці фреймворки були розроблені тими самими розробниками, які впровадили Samzав Linked In, апотімзаснували Confluent, де вони написали Kafka Streams. Обидві ці технології тісно пов'язані з Kafka: беруть необроблені дані з Kafka, а потім повертають оброблені дані назад до Kafka. Самза – це свого роду масштабована версія Kafka Streams. Поки Kafka Streams – це бібліотека, призначена для мікросервісів, Samza – це повноцінна кластерна обробка.

5. Як обрати найкращий фреймворк

Важливо мати на увазі, що жоден механізм обробки не може бути срібною кулею для кожного випадку використання. Кожен фреймворк має деякі сильні сторони та деякі обмеження. Якщо варіант використання простий, немає необхідності шукати найновіші та найкращі фреймворки, якщо це складно вивчити та впровадити. Багато залежить від того, скільки ми готові вкласти, скільки хочемо взамін. Наприклад, якщо це проста система сповіщення на основі подій IOT, з Storm або Kafka Streams цілком чудово працювати.

У той же час нам також слід усвідомлено розглянути, якими будуть можливі випадки використання в майбутньому? Чи можливо, що в майбутньому можуть виникнути вимоги до розширених функцій, таких як обробка часу подій, агрегування, приєднання потоків тощо? Якщо відповідь позитивна, тоді краще піти на вдосконалені фреймворки потоковоїобробки, такі як Spark Streaming або Flink. Після інвестування в одну технологію - її важко змінити пізніше. Якщо ми добре розуміємо сильні сторони та обмеження фреймворків разом із необхідними варіантами використання, тоді легше обрати або хоча б фільтрувати наявні варіанти. Нарешті, завжди добре мати POC, коли буде обрано пару варіантів.

Висновки

Простір Apache Streaming розвивається настільки швидкими темпами, що ця публікація може застаріти з точки зору інформації вже через пару років. В даний час Spark і Flink є важкоатлетами, що знаходяться попереду, з точки зору розвитку подій, але новийучасник все ще може прийти і приєднатися до гонки.

Список літератури

1. WhatisStreamProcessing? [Електронний ресурс] – Режим доступу до ресурсу: https://www.ververica.com/what-is-stream-processing
2. SparkStreamingProgrammingGuide [Електронний ресурс] – Режим доступу до ресурсу: https://spark.apache.org/docs/latest/streaming-programming-guide.html#spark-streaming-programming-guide
3. ApacheFlinkDocumentation [Електронний ресурс] – Режим доступу до ресурсу: https://ci.apache.org/projects/flink/flink-docs-stable/
4. WhatisApacheStorm? [Електронний ресурс] – Режим доступу до ресурсу: https://intellipaat.com/blog/what-is-apache-storm/
5. The easiest way to write mission-critical real-time applications and microservices [Електронний ресурс] – Режим доступу до ресурсу: https://kafka.apache.org/documentation/streams/

УДК004.43

ВАЩЕНКОЮ.О.
ФІНОГЕНОВ О.Д.

ЗАСТОСУВАННЯ СТАТИЧНОГО АНАЛІЗУ КОДУ НА ПРИКЛАДІ DATA-FLOWАНАЛІЗУ

У даній статі розглянуто підходи для статичного аналізу коду. Демонструєтьсяможливості використання та інформація, яку можна дізнатись в разі виконання. Наведено приклади використання різних методів та аналіз на прикладі коду C#.
STATIC CODE ANALYSIS, DATA FLOW ANALYSIS, TEXT ANALYSIS

This article discusses approaches to static code analysis. Demonstrates usability and information that can be learned in case of execution. Examples of the use of different methods and analysis on the example of C # code is given.
STATIC CODE ANALYSIS, DATA FLOW ANALYSIS, TEXT ANALYSIS

1. Вступ

В наші часи програмне забезпечення все більше та більше входить в різні сфери суспільного життя, беручи на себе велику кількість зобов’язань. Це системи керування медичними приладами, складною технікою, літаками. В таких проектах успішність виконаних операцій та навіть життя людей повністю покладаються на якість розробленого програмного забезпечення. Одним із способів перевірки коректності програмного коду є статичний аналіз.

Статичний аналіз програм - це аналіз комп'ютерного програмного забезпечення, який виконується без фактичного запуску програм, на відміну від динамічного аналізу, який є аналізом, що виконується програмами під час їх виконання. У більшості випадків аналіз виконується для якоїсь версії вихідного коду, а в інших випадках - для якоїсь форми об'єктного коду. Одним із видів статичного аналізу є аналіз потоку даних.

Аналіз потоку даних - це техніка для збору інформації про можливий набір значень, обчислених у різних точках комп'ютерної програми. Графік потоку керування програмою (CFG) використовується для визначення тих частин програми, на які може поширюватися певне значення, присвоєне змінній.

В наступних розділах буде розглянуто підходи для проведення аналізу потоку даних в програмному коді.

2.Постановка задачі

На вхід кожного з видів аналізу подається програмний код на мові програмування C#, що потрібно проаналізувати та знайти в ньому всі можливі потоки виконання, зокрема місця коду, які в результаті не будуть виконані.На виході кожного з аналізів отримується структура даних в якій містяться позиції коду, в яких буде стан змінних буде проініціалізований.

3. Основні принципи

Аналіз потоку даних намагається отримати конкретну інформацію в кожній точці процедури. Зазвичай достатньо отримати цю інформацію на межах основних блоків програмного коду, оскільки з цього легко обчислити інформацію в точках основного блоку. При прямому аналізі потоку стан виходу блоку є функцією стану входу блоку. Ця функція є об’єднанням виконання операторів у блоці. Стан входу блоку є функцією станів виходу його попередників. Маємо набір рівнянь потоку даних:

Для кожного блоку b:

$$out_b = trans_b(in_b)$$

$$in_b = join_{p \in pred_b}(out_p),$$

де $trans_b$ – передавальна функція блоку b.Після вирішення цього набору рівнянь стану входу та / або виходу блоків можна використовувати для отримання властивостей програми на межі блоків. Функція передачі кожного висловлення

окремо може бути застосована для отримання інформації в точці всередині основного блоку. Кожен конкретний тип аналізу потоку даних має свою особливу функцію передачі та операції приєднання. Деякі проблеми аналізу потоку даних вимагають аналізу зворотного потоку. Він працює так же, за винятком того, що функція передачі застосовується до стану виходу, що дає стан входу, а операція приєднання працює над станами входу наступників, щоб отримати стан виходу.

4. Ітераційний алгоритм

Найбільш поширеним способом вирішення рівнянь потоку даних є використання ітераційного алгоритму.

Починається з апроксимації стану кожного блоку. Потім вихідні стани обчислюються шляхом застосування передавальних функцій у внутрішніх станах. З них внутрішні штати оновлюються, застосовуючи операції об'єднання. Два останні кроки повторюються, поки ми не досягнемо так званої точки фіксування : ситуації, в якій внутрішні (і, як наслідок, зовнішні стани) не змінюються.

Основним алгоритмом вирішення рівнянь потоку даних є ітераційний круговий ітераційний алгоритм:

для *i* ← 1 до *N*
ініціалізувати вузол i
while (*набори все ще змінюються*)
  для *i* ← 1 до *N*
перерахувати набори у вузлі i

Щоб бути придатним для використання, ітераційний підхід повинен фактично досягти точки фіксації. Це може бути гарантоване накладенням обмежень на комбінацію області значень станів, функцій передачі та операції об'єднання.

5. Приклади використання

Форвадний аналіз – це аналіз визначень, що обчислює для кожної точки програми набір визначень, які потенційно можуть досягти цієї точки програми. Для прикладу код на Рис. 1, на якому визначальне значення змінної, а у рядку10 - це набір призначень *a = 5*, у рядку 5 та *a = 3* у рядку 7.

```
4  if (b == 4) {
5      a = 5;
6  } else {
7      a = 3;
8  }
9
10 if (a < 4) {
11     //...
12 }
```

Рис. 1. Приклад коду для форвардного аналізу

Аналіз назад – аналіз змінних в реальному часі, що обчислює для кожної точки програми змінні, які потенційно можуть бути прочитані після цього перед наступним оновленням запису. Результат, як правило, використовується шляхом усунення мертвого коду для видалення операторів, які присвоюються змінній, значення якої потім не використовується.

Внутрішній стан блоку - це набір змінних, які працюють на початку його. Спочатку він містить усі змінні в реальному часі (містяться) у блоці, перш ніж застосовуватиметься функція передачі та обчислюватимуться фактичні значення, що містяться. Функція передачі оператора застосовується вбивством змінних, записаних у цьому блоці (вилучіть їх із набору змінних, що працюють). Зовнішній стан блоку - це набір змінних, які працюють в кінці блоку, і обчислюється об'єднанням внутрішніх станів наступників блоку. Розглянемо початковий код – Рис. 2.

```
1  let b = 4;
2  let a, x, d, c = 5;
3
4  b1: a = 3;
5      b = 5;
6      d = 4;
7      x = 100;
8      if(a > b) {
9  b2:     c = a + b;
10         d = 2;
11 b3: }
12     c = 4;
13     return b * d + c;
14
```

Рис.2. Приклад коду для аналізу назад

Проаналізуємо цей приклад. Вхідний стан b3 містить лише b і d , оскільки c написано. Зовнішній стан b1 - це об'єднання внутрішніх станів b2 і b3. Визначення c з b2 можна вилучити, оскільки c не діє безпосередньо після оператора.

Вирішення рівнянь потоку даних починається з ініціалізації всіх вхідних та вихідних станів до порожнього набору. Робочий список ініціалізується шляхом вставки точки виходу (b3) у робочий список (типово для зворотного потоку). Його обчислюваний внутрішній стан відрізняється від попереднього, тому його попередники b1 та b2 вставляються, і процес продовжується.Зверніть увагу, що b1 було внесено до списку перед b2, що примусило обробляти b1 двічі (b1 було повторно введено як попередник b2). Вставка b2 перед b1 могла б дозволити більш раннє завершення.

Ініціалізація порожнім набором є оптимістичною ініціалізацією: всі змінні починаються як мертві. Слід зазначити, що зовнішні стани не можуть зменшуватися від однієї ітерації до наступної, хоча зовнішній стан може бути меншим, ніж внутрішній. Це видно з того факту, що після першої ітерації зовнішній стан може змінитися лише шляхом зміни внутрішнього стану. Оскільки in-state починається як порожній набір, він може зростати лише в подальших ітераціях. Відобразимо наглядно прикладдля цього блоку на Рис.3.

```
1
2   // y: {}
3   b1: a = 3;
4       b = 5;
5       d = 4;
6       x = 100; // x ніколи не використовується пізніше, отже, не в наборі {a, b, d}
7       if (a > b) {
8   // out: {a, b, d} // об'єднання всіх (не) наступників b1 => b2: {a, b} та b3: {b, d}
9
10  // y: {a, b}
11  b2: c = a + b;
12      d = 2;
13  // вихід: {b, d}
14
15  // y: {b, d}
16  b3: }
17      c = 4;
18      return b * d + c;
19  // вихід: {}
```

Рис. 3. Приклад станів системи при проведенні «аналізу назад»

Висновки

Статичний аналіз коду є досить корисним інструментом для більшості розробників та просто необхідним для деяких галузей в яких інженер-програміст не має право на помилку. Було розглянуто основні підходи в статичному аналізі, зокрема ітеративний. До нього відносять «аналіз вперед» та «аналіз назад», які мають можливість знаходити код, що не може бути виконаний, отримати можливі значення змінних по блокам коду, а також отримати різна деталі по можливим помилковим станам програми під час виконання. Також dataflowаналіз може застосовуватись в компіляторах для оптимізації результуючих збірок.

Список літератури

1. Static Program Analysis [Електронний ресурс] // Режим доступу до ресурсу: https://cs.au.dk/~amoeller/spa/.
2. Static Analysis of Numerical Algorithms[Електронний ресурс] // Режим доступу до ресурсу: http://www.lix.polytechnique.fr/~putot/Publications/sas06.pdf
3. Data flowan alysisin compilers[Електронний ресурс] // Режим доступу до ресурсу: https://www.geeksforgeeks.org/data-flow-analysis-compiler/

УДК 004.089

Медведніков Д.С.
Олійник Ю.О.

Геопросторовий сентимент-аналіз текстових потоків даних

В даній роботі розглянуто підходи до аналізу тональностей тексту. Сформульовано цілі та актуальність дослідження. Описані складові системи геопросторового сентименту-аналізу текстових потоків даних та принципи її роботи, а саме задача аналізу потоку повідомлень з новинних каналів та візуалізація сентименту за країнами, що згадуються, на карті.

КЛЮЧОВІ СЛОВА: СЕНТИМЕНТ-АНАЛІЗ, ГЕОПРОСТОРОВИЙ АНАЛІЗ, АНАЛІЗ ТОНАЛЬНОСТЕЙ, ТОНАЛЬНІСТЬ

In this article the main approaches of sentiment analysis are considered. The article contains formulations of the goals and relevance of research. The system of geospatial sentiment analysis of text data streams and principles of it work are described including the task of sentiment analysis of text data stream from news channels and visualization of mentioned countries on the map.

KEYWORDS: SENTIMENT ANALYSIS, GEOSPATIAL ANALYSIS,

1. Вступ

Більшість даних у світі неструктуровані та неорганізовані, велика купа текстових даних (електронні листи, чати та бесіди в соціальних мережах, статті, документи, опитування тощо) створюється щодня, але її важко проаналізувати, зрозуміти та сортувати, не кажучи вже про трудомісткість та дороговартісність. Тому в часи швидкого всебічного розвитку інтернет-павутини дуже важливим є структурування даних про почуття клієнта чи сортування великої кількості повідомлень за настроєм.

Аналіз тональності тексту - це область дослідження, що аналізує погляди, оцінки, думки та емоції людей по відношенню до таких об'єктів як: продукти, послуги, організації, особисті питання, події, теми, характеристики. Аналіз тональності знаходить своє практичне застосування в таких областях як: політологія, маркетинг, соціологія, медицина та психологія.

На сьогоднішній день, коли людина хоче купити продукт, вона вже не обмежується тим, щоб запитати думку друзів чи родини, тому що в Інтернеті вже є багато відгуків та дискусій на публічних форумах щодо продукту. А для організацій немає необхідності проводити опитування суспільної думки та фокус-групи для збору громадської думки, тому що такої інформації в достатку є у відкритому доступі. Кожен сайт, як правило, містить величезний обсяг тексту думок, який не завжди легко розшифровується в довгих блогах і повідомленнях на форумі. Середньостатистичному користувачу буде важко визначити відповідні сайти, а також отримати і узагальнити думки, що містяться на них. Таким чином, необхідні автоматизовані системи аналізу настроїв.

2. Постановка задачі аналізу потоку текстових повідомлень

Дано потік повідомлень D:

$D = \{d_1, d_2, \dots d|D|\}$

Функція класифікації тональності лексичних одиниць, що складають потік повідомлень $\Phi(d_j)$. приймає значення від -5 до -1, якщо оцінка негативна, 0, якщо оцінка нейтральна та від 1 до 5 якщо оцінка позитивна.

Необхідно визначити значення цільової функції f, яка являє собою суму оцінки лексичних тональностей одиниць $\Phi(d_j)$. Тобто:

$$f(D) = \sum_{i=1}^{D} \Phi(d_j).$$

Додатнє значення функції f являє собою позитивну оцінку, від'ємне значення - негативну. Якщо значення функції дорівнює нулю, то текст є тонально нейтральним.

3. Підходи до розв'язання задачі

Перед застосуванням будь-яких із зазначених далі алгоритмів, необхідно виконати попередню обробку даних.

Зазвичай, вона складається з двох частин - сегментації та токенізації та лематизацію, що виконуються послідовно[1].

Сегментація - це процес розділення тексту природною мовою на компоненти-речення. При цьому необхідно зважати на неподільні токени, які містять крапки, пробіли та інше. Наприклад, крапка може бути використана не тільки у кінці речення, а й у якості скорочення.

Токенізація - процес розділення речень на компоненти-слова. В українській мові у якості роздільника дуже зручно використовувати пробіл, але треба також обробляти виключення, коли слова складаються з декількох частин, наприклад "сентимент-аналіз".

Зазвичай, слова у реченнях зустрічаються у різних формах і з різними закінченнями. Мета лематизації привести токени до більш словникового вигляду. Розглянемо речення "Діти грають на вулиці", після лематизації воно буде приведено до вигляду "діти грати на вулиця", що надасть можливість більш ефективно шукати ці токени по словниках.

Крім лематизації також виділяють стемінг. Стемінг є більш "грубим" процесом нормалізації тексту, якій просто відкидує усе зайве.

В сучасних системах автоматичного визначення емоційної оцінки тексту частіше за все використовується одновимірний емотивний простір: позитив або негатив.

Найбільш поширеними є наступні методи аналізу тональностей[2]:

- методи, що засновані на словниках та правилах
- машинне навчання з вчителем
- машинне навчання без вчителя
- методи, що засновані на теоретико-графових моделях

Метод словників та правил базується на пошуку емотивної лексики в тексті по заздалегідь складеним тональним словником, та правилам з використанням лінгвістичного аналізу[3]. За сукупністю знайденої емотивної лексики текст можна оцінити за шкалою. Для того щоб оцінити текст за цим методом необхідно кожному слову в тексті присвоїти значення його тональності зі словника, а потім порахувати загальну тональність, підсумувавши значення тональностей кожного окремого речення. Варто зазначити, що крім слів та словосполучень, певне емоційне навантаження можуть також мати емодзі (англ. emoji), хештеги та знаки пунктуації (наприклад "!") [4].

Основною проблемою методів, заснований на правилах та словниках, вважається трудоємність процесу підготовки словника. Для того, щоб отримати інструмент для класифікації тональності документа з високою точністю, необхідний словник, що буде мати оцінку термінів адекватну предметній області документу. Наприклад, слово "величезний" може бути позитивним щодо розміру пам'яти жорсткого диску та негативним щодо розміру мобільного телефону. Також даний метод потребує великих трудовитрат для складання великої кількості правил, необхідних для коректної роботи системи.

Машинне навчання з вчителем є найбільш широко використовуваним. Суттю таких методів є те, що на першому етапи навчається машинний класифікатор на заздалегідь розмічених текстах, а потім отриману модель використовують для аналіза нових документів. Приклад алгоритму[10]:

1. Спочатку збирається колекція документів, на основі яких навчається машинний класифікатор
2. Кожен документ розкладається у вигляді вектора аспектів, за якими він буде досліджуватися
3. Вказується правильний тип тональності для кожного документа
4. Відбувається вибір алгоритму класифікації та методу для навчання класифікатора
5. Отриману модель використовуємо для визначення тональності нової колекції документів

Головною перевагою даного підходу є те, що навчанні моделі мають більш гнучкі можливості щодо обробки текстових даних. Наприклад, вони можуть розпізнавати семантично однакові речення, сарказм, пропаганду та інше.

В основі цього **методу навчання без вчителя** покладено те, що терміни, які найчастіше зустрічаються у тексті і в той

же час присутні у меншій кількості в інших документах цієї колекції, мають найбільшу вагу в тексті. Виділивши дані терміни, а потім визначивши їх тональність, можна зробити висновок щодо тональності тексту взагалі[10].

В методах, що засновані **на теоретико-графових моделях**[5] використовується припущення, що не всі слова в текстовому корпусі документа рівнозначні. Деякі слова мають більше вагу та сильніше впливають на тональність тексту. При використанні цього методу, аналіз тональності розбивається на декілька етапів:

1. Побудова графа на основі досліджуваного тексту
2. Ранжування його вершин
3. Класифікація знайдених слів
4. Підрахування результату

Для класифікації слів використовується тональний словник. Для отримання кінцевого результату необхідно визначити значення двох оцінок: позитивною складової та негативної. Для того щоб знайти позитивну складову тексту необхідно знайти суму тональностей всіх позитивних термінів в тексті. Аналогічно знаходиться негативна складова. Кінцева тональність тексту визначається як співвідношення за формулою:

$$T = \frac{P}{N},$$

де T - загальна тональність тексту, P - позитивна складова, N - негативна складова.

Таким чином текст, для якого значення T близько до 1, вважається нейтральним. Для якого T незначно перевищує 1 - позитивним, якщо значно перевищує 1 - сильно позитивним. Так само і для негативних.

1.5. Оцінка якості сентимент-аналізу

Точність та якість системи сентиментального аналізу оцінюється тим, наскільки хорошо вона узгоджується з думкою людини відносно емоційної оцінки досліджуваного тексту. Для цього можуть використовуватись такі метрики, як точність та повнота. Формула для знаходження повноти:

$$R = \frac{O_c}{O_T},$$

де R - повнота аналізу, Oc - кількість коректно визначених думок, O_T - загальна кількість думок (як знайдених системою, так і не знайдених).

Точність визначається за формулою:

$$P = \frac{O_c}{O_{TS}},$$

де P - точність, Oc - кількість коректно визначених думок, O_{TS} - загальна кількість думок, знайдених системою.

Таким чином, точність показує кількість досліджуваних текстів, речень або документів, в оцінці яких думка системи сентиментального аналізу збігається з думкою експерта.

4. Обробка текстових потоків даних

Потік - умовно нескінченна послідовність елементів даних, що стають доступними протягом часу. Кажучи про аналіз тональності повідомлень, потоком можуть бути повідомлення з соціальних мереж, новинних сайтів та месенджерів.

Існує багато технологій та засобів для обробки потоків даних, найбільш поширеними серед них є наступні:

- Apache Hadoop;
- Apache Storm;
- Splunk;
- Apache Spark;
- Apache Samza;
- Apache Kafka;

Для реалізації системи геопросторового аналізу потоків текстових даних було використано сервер повідомлень Apache Kafka**.** Базова топологія[6] Kafka складається з таких компонент:

- producer;
- consumer;
- broker;
- Zookeeper.

За зберігання даних відповідає брокер (англ. broker). Всі дані зберігаються в бінарному вигляді, і брокер мало знає про те, що вони з себе представляють і яка їхня структура.

Кожен логічний тип подій зазвичай знаходиться в своєму окремому топіку (англ. topic). У свою чергу, кожен топік розбивається на одну і більше партіції (англ. partition). Саме в партіції в результаті потрапляють події. Якщо в кластері більш одного брокера, то партіції будуть розподілені по всьому брокерам рівномірно

(наскільки це можливо), що дозволить масштабувати навантаження на запис і читання в один топік відразу на кілька брокерів.

Zookeeper виконує роль сховища метаданих та координатора.

Producer - це найчастіше сервіс, який здійснює безпосередню запис даних в Apache Kafka. Producer вибирає topic, в якому будуть зберігатися його тематичні повідомлення, і починає записувати в нього інформацію.

Consumer відповідає за отримання даних з Apache Kafka. У тих випадках, коли одного consumer недостатньо (наприклад, потік нових подій дуже великий), можна додати ще кілька consumer, зв'язавши їх разом в “consumer group”. “Consumer group” логічно представляє з себе точно такий же consumer, але з розподілом даних між учасниками групи. Це дозволяє кожному з учасників взяти свою частку повідомлень, тим самим масштабуючи швидкість читання.

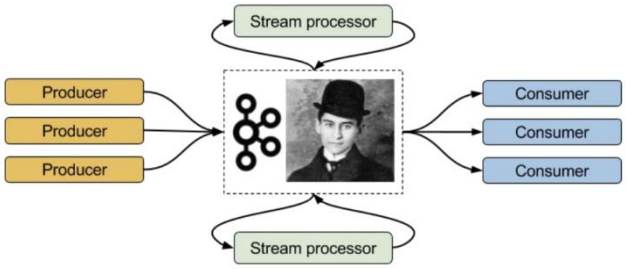


Рис. 2. Архітектура Apache Kafka

Отже, Apache Kafka збирає дані у продюсерів (producer), зберігає дані у себе в розподіленому сховищі по топікам (topic) і роздає консьюмерам (consumer) за підпискою. Іншими словами, Kafka - це гібрид розподіленої бази даних і черги повідомлень.

5. Візуалізація тональності потоків даних

Візуалізація тональності - задача з області візуалізації інформації та візуальної аналітики для аналізу сентименту, знайденому у тексті. Застосунки з візуалізації сентименту можуть використовуватись для моніторингу думки спільноти у соціальних мережах, аналізу літератури та іншого.

На сьогодні існують сотні технік[7] візуалізації сентименту та їх кількість постійно зростає. Для того щоб обрати техніку, що задовольняє задачам дослідження необхідно зважати на наступні фактори:

- Область даних (соціальні мережі, новини, художні твори, відгуки, научна література)
- Джерело даних (документ, потік)
- Властивості даних (геопросторові, часові)
- Тип аналітичної задачі (аналіз полярності, opinion mining, аналіз емоцій)
- Тип задачі візуалізації (кластеризація, порівняння, огляд, моніторинг)
- Візуальна складова (колір, форма, розмір, позиція)

Виходячи з цього, найбільш відповідними до нашої задачі є такі техніки як Sentiment Map[8] та Weibo Sentiment Visualization[9].

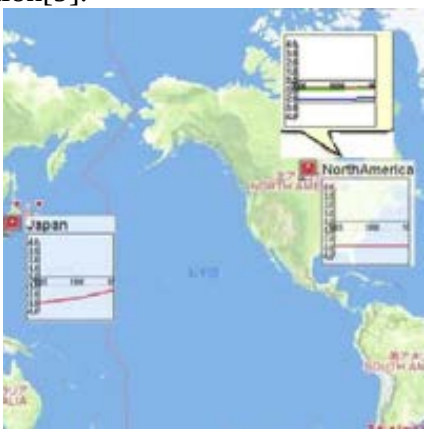


Рис. 1.. Візуалізація Sentiment Map

Рис. 2. Візуалізація Weibo Sentiment Visualization

6. Система геопросторового аналізу текстових даних

Реалізована система являє собою серверну частину, що включає в себе NodeJS застосунок на брокер повідомлень Apache Kafka, та клієнтську частину, що

представлена веб-застосунком, створеним з використанням фреймворку VueJS та бібліотеки Leaflet для візуалізації результатів аналізу на карті.

Таким чином, система “підписується” на новинні канали у месенджері Telegram, після чого кожне повідомлення з цих каналів потрапляє до Apache Kafka, а потік повідомлень обробляється застосунком на NodeJS, де визначається сентимент кожного окремого повідомлення, а також з тексту виділяються країни чи інші географічні назви, що були згадані у цьому повідомленні. Веб-застосунок відображає результати сентимент-аналізу на карті світу, змінюючи забарвлення (від зеленого до червоного відповідно) кожної країни в залежності від контексту, в якому вони були згадані, у реальному часі.

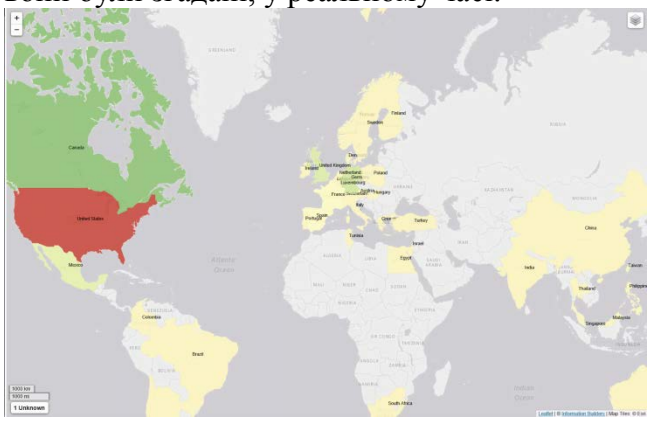


Рис. 3. Візуалізація результатів геопросторового аналізу текстового потоку.

Висновки

В процесі дослідження було розглянуто основні алгоритми сентимент аналізу, описані їх переваги та недоліки. Розглянуто методи та засоби роботи з потоками текстових даних та розроблено алгоритм геопросторового сентимент-аналізу україномовних текстів. Розроблено програмне забезпечення геопросторового аналізу текстових потоків даних з візуалізацією даних за країнами.

Список літератури

1. Олійник Ю.О., Афанасьева О.Є., Аршакян Г.Д. Деякі аспекти аналізу потоків текстових даних — ІТММ 2020 — с. 366-369
2. Степанюк Є.Ю. Олійник Ю.О. Дослідження методів аналізу тональності тексту. Інформаційні системи та технології управління: матеріали всеукр. наук.-практ. конф. молодих вчених та студентів, (м. Київ, 26 лист. 2019 р.). Київ, 2019. С. 32-39
3. Анна Пазельская, Алексей Соловьев. Метод определения эмоций в текстах на русском языке // The international conference on computational linguistics and intellectual technologies “Dialogue 2011” : конференция. — Москва, 2011. — с. 516-522.
4. Olga Artemenko, Volodymyr Pasichnyk, Nataliia Kunanets, Khrystyna Shunevych — Using Sentiment Text Analysis of User Reviews in Social Media for E-Tourism Mobile Recommender Systems — pp. 6-11
5. М. В. Клековкина, Е.В. Котельников. Метод автоматической классификации текстов по тональности, основанный на словаре эмоциональной лексики (рус.) // RCDL-2012, Переславль-Залесский, Россия : конференция. — 2012.
6. Apache Kafka [Електронний ресурс] – Режим доступу до ресурсу: https://kafka.apache.org/.
7. Kostiantyn Kucher, Andreas Kerren, Carita Paradis — The State of the Art in Sentiment Visualization // Computer Graphics Forum — pp. 1-7
8. Jianwei Zhang, Yukiko Kawai, Tadahiko Kumamoto, and Katsumi Tanaka. A Novel Visualization Method for Distinction of Web News Sentiment. Proceedings of the International Conference on Web Information Systems Engineering (WISE), pp. 181-194, 2009
9. Zhiwen Yu, Zhitao Wang, Liming Chen, Bin Guo, and Wenjie Li. Featuring, Detecting, and Visualizing Human Sentiment in Chinese Micro-Blog. ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 10, no. 4, 2019
10. Бартков'як, А. Ю. Метод автоматизованої класифікації коротких новинних текстів з використанням іменованих сутностей : магістерська дис. : 121 Інженерія програмного забезпечення / Бартков'як Андрій Юліанович. - Київ, 2018. - 104 с.

УДК 004.089

МИГАЛЬ Д.С.
ОЛІЙНИК Ю.О.

МЕТОДИ ТА ЗАСОБИ СЕМАНТИЧНОГО АНАЛІЗУ ТЕКСТІВ

В цій статті розглянуто задачу семантичного аналізу україномовних текстів. Сформульовано цілі та актуальність дослідження. Проаналізовано існуючі математичні методи тематичного моделювання текстів. Розглянуто різні програмні засоби, описана ефективність цих застосунків.

КЛЮЧОВІ СЛОВА: тематичне моделювання, семантичний аналіз, LSA, україномовні тексти

1. Вступ

Автоматична обробка текстових документів набуває все більшої необхідності у сучасному світі. Попит на аналіз великої кількості документів зростає щодня, адже це обумовлено накопиченням текстової інформації в мережі за рахунок мільйонів веб-сайтів та застосунків. За статистикою розмір World Wide Web станом на травень 2020 року оцінюється близько 5.47 мільярдів сторінок [1]. Тому це очевидно, що без допомоги автоматичної обробки, аналізувати інформацію такого об’єму неможливо. За оцінюванням топ десяти мільйонів веб-сторінок, україномовний контент містить близько 0.3%, що в десятки разів менше ніж російськомовний – 8.6% та значно менше ніж відсоток англомовного контенту – 59.6% [2]. Саме тому на даний час не існує багато програмних рішень автоматичного семантичного аналізу української мови, порівняно з англійською та російською.

Семантика - розділ лінгвістики, що вивчає смислове значення одиниць мови. Семантичний аналіз дає точне або словникове значення зі структур, створених синтаксичним аналізом. Основна мета семантичного аналізу - мінімізувати структури синтаксису та знайти її значення. Семантичні моделі тексту, що є результатом комплексного аналізу, дозволяють оцінити коректність тексту, в наочній формі, візуально уявити структуру сюжету, взаємозв'язок об'єктів і процесів тексту, їх атрибути. Послідовність моделей простих речень тексту і результуюча візуальна модель тексту дозволяють реалізувати зворотний зв'язок "вплив на модель - реакція в тексті", завдяки чому можна в інтерактивному режимі налагоджувати процеси аналізу текстів і докази об'єктивності та однозначності тлумачення текстів на природних мовах.

З огляду на вищевикладене, створення нових методів семантичного аналізу текстів відкриє нові можливості та дозволить істотно просунутися у вирішенні багатьох завдань комп'ютерної лінгвістики, таких як машинний переклад, автореферування, класифікація текстів і т.п. Не менш актуальною є розробка нових засобів та інструментів, що дозволяють автоматизувати семантичний аналіз. Подібні методи аналізу дозволяють збирати основну інформацію про тематику, спрямованості і настрої текстів, що в подальшому спрощує автоматизовану роботу з ними, таку як каталогізація, пошук і порівняння.

2. Мета роботи

Підвищення можливостей семантичного аналізу україномовних текстів за рахунок моделей знань про граматику української мови і про предметну область тексту, а також шляхом вдосконалення семантичних моделей оброблюваного тексту.

3. Основна частина

Тематичне моделювання - це спосіб побудови моделі корпусу текстів, що відображає перехід від сукупності документів, сукупності слів в документах до набору тем, що характеризують зміст даних документів. Наразі існують реалізації моделей тематичного моделювання для англійської та російської мов, проте не існує таких для української.

Найбільш популярні методи тематичного моделювання можна розділити на дві основні групи - алгебраїчні та ймовірнісні. До алгебраїчних моделей належать стандартна векторна модель тексту VSM і латентно-семантичний аналіз LSA, а серед імовірнісних найбільш популярними є

імовірнісний латентно-семантичний аналіз PLSA і латентне розміщення Діріхле LDA [3].

Серед розвинутих засобів семантичного аналізу українськомовних текстів є морфоаналізатор pymorphy2 [4], який написаний на мові Python і розміщений у відкритому репозиторії github. Містить функції приведення слова до нормальної форми (леми), наприклад "люди -> людина", або "гуляв -> гуляти". Також є можливість поставити слово в потрібну користувачу форму, наприклад у множину, міняти відмінок слова тощо. Pymorphy2 також дозволяє отримувати граматичну інформацію про слово – його число, рід, відмінок, частину мови. У роботі pymorphy2 використовується словник OpenCorpora, а для незнайомих слів будуються гіпотези. Бібліотека є досить швидкою, зараз швидкість роботи - від декількох тисяч слів до більше ніж 100 тисяч слів за секунду в залежності від виконуваної операції, інтерпретатора та встановлених пакетів.

Результатом роботи повинна стати модель, що проводить тематичне моделювання корпусу україномовних текстів, і яка використовує вищезгадані методи та засоби.

На вхід моделі подається корпус україномовних текстів, кожен з яких повинен мати заголовок та контент, який і буде проаналізований. Корпус повинен містити достатню кількість документів на різну тематику, щоб модель могла виділити декілька різних тем. Корпус не повинен містити дублікати документів.

Також необхідно передати два параметри обмежувачі.

Перший - це максимальна кількість тем, які ми хочемо отримати на виході. Оскільки модель не може визначити чи достатньо розподілити корпус на певну кількість тем, то користувач повинен встановити обмеження. Це не означає що модель визначить рівно стільки тем, як і число в обмежувачі, оскільки деякі теми можуть включати в себе інші.

Другий - це максимальна кількість слів, які повинні містити всі теми в загальному. Цей параметр необхідний в більшості для візуалізації результату у вигляді графу, оскільки занадто складні структури можуть бути тяжкими для сприйняття.

На виході з моделі отримуємо певну кількість тем, які були виділені з корпусу текстів. Кожна тема - це множина термів, сортовані за релевантністю до цієї теми за спаданням.

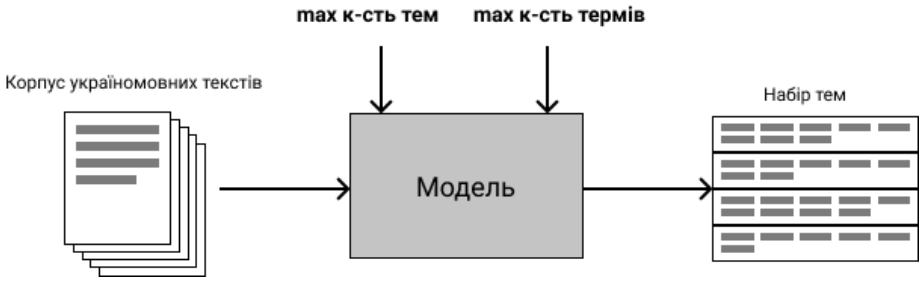


Рис.1. Діаграма вхідних і вихідних даних моделі

Розглянемо як працює модель всередині. Насамперед необхідно провести попередню обробку всіх документів. Перш за все це видалення усіх знаків пунктуації та спеціальних знаків. Також необхідно виключити, так звані, стоп-слова або шумові слова — це слова, які не несуть змістовного навантаження, тому їх користь та роль для пошуку не суттєва, наприклад всі прийменники, суфікси, дієприкметники, вигуки, цифри тощо. Список шумових слів для української мови можна знайти в репозиторії українського аналізатору пошукової системи Lucene.

Необхідно привести всі слова до нормальної форми, а також визначити частину мову цього терму за допомогою морфоаналізатором pymorphy2. Після нормалізації з масиву слів видаляються усі дублікати - так ми отримуємо вектор термів для кожного документу.

З векторів термів усіх документів формується bag of words - вектор що містить терми, які зустрічаються в усіх документах. Після цього формується частотна матриця

термів (Term Frequency) та вектор зворотної частоти документів (Inverse Document Frequency). Їх добуток дає нам TF-IDF матрицю, яка надає більші значення термам, які зустрічаються частіше в документі, але не часто трапляються у всіх інших документах, і відповідно навпаки [5].

Наступним етапом є розкладання матриці методом сингулярного розкладання (Singular value decomposition, SVD). Сингулярне розкладання - це математична операція, за допомогою якої матрицю розкладають на 3 складових. Сингулярне розкладання можна уявити у вигляді формули:

$$A = U \times S \times VT$$

де A - вихідна матриця, U і VT - ортогональні матриці, а S - діагональна матриця, значення, на діагоналі якої називаються сингулярними коефіцієнтами матриці A і які упорядковані за спаданням.

Сингулярне розкладання дозволяє виділити ключові складові вихідної матриці. Основна ідея полягає в тому, що якщо в якості матриці A використовувалася TF-IDF матриця, то матриця A*, що містить тільки *k* перших лінійно незалежних компонент, відображає основну структуру різних залежностей, присутніх у вихідній матриці. Структура залежностей визначається ваговими функціями термів. Це означає що передавши обмеження кількості тем як параметр *k* ми отримаємо матрицю тем A*, в якій стовпці відповідають темам, а рядки термам. Значення матриці - це релевантність належності певного терму до певної теми [5].

Після цього можемо співставити кожному значенню матриці слово, зробити сортування по релевантності та повернути набір тем, що складається з масиву термів.

Також кожен з термів ми можемо пропустити через NER-аналізатор. Named-entity recognition (NER, з англійської - Розпізнавання іменованих сутностей) це підзадача видобування інформації, яка намагається знайти і класифікувати іменовані сутності в заздалегідь визначені категорії, такі як імена людей, організації, місця, медичні коди, час, кількості, грошові значення, відсотки тощо. Так завдяки цьому аналізатору ми можемо отримати більше інформації про кожен терм (що є іменником) на виході.

Отримані дані можна візуалізувати у вигляді певної структури, наприклад графу, який наглядно демонструє важливість термів в документах і темах та їх зв'язки між собою. В даній роботі був реалізований графічний інтерфейс у вигляді веб-застосунку, де користувач може інтерактивно взаємодіяти з графом термів, списком документів та тем, а також змінювати вхідні значення моделі

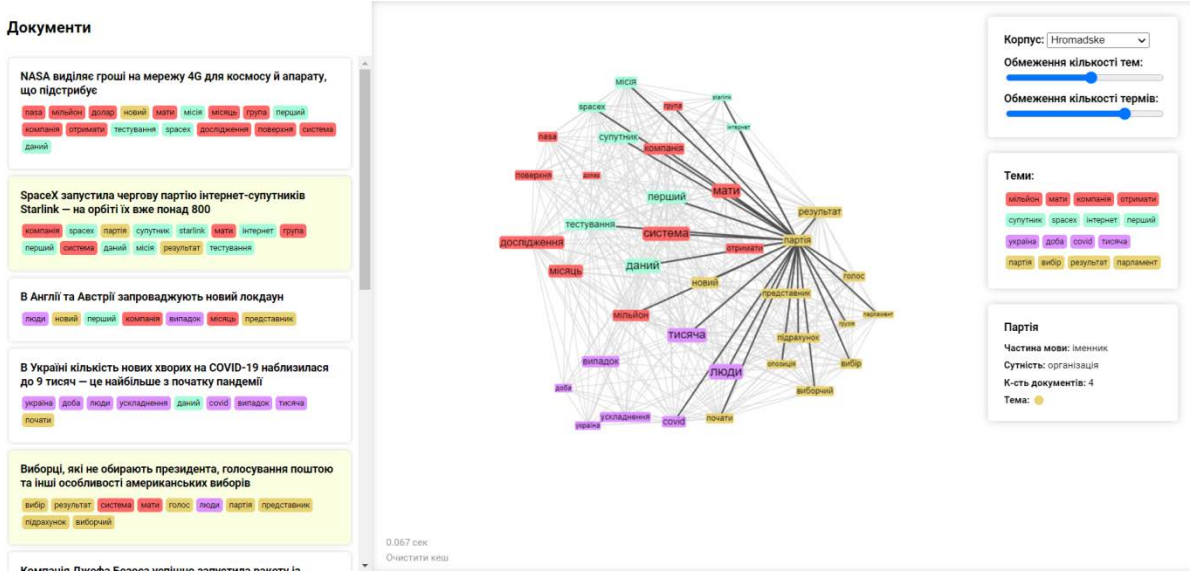


Рис. 2. Зображення візуалізації результатів тематичного аналізу а вигляді графу

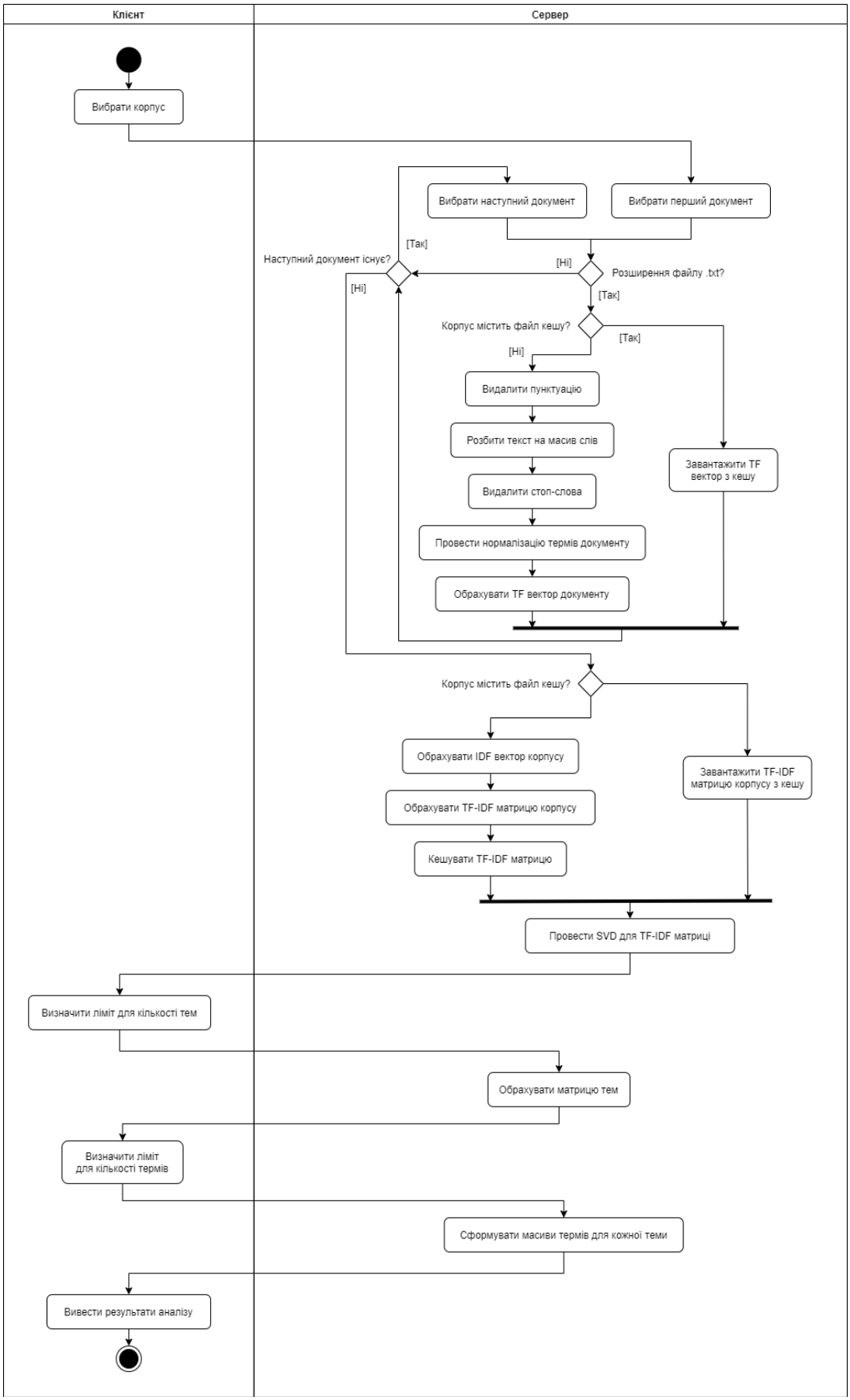


Рис. 3. Діаграма діяльності моделі

Висновки

Розроблено модифікований метод семантичного аналізу тексту, який дає можливість аналізу українськомовного та російськомовного контенту шляхом застосування морфоаналізатора Pymorphy2. Збільшено семантичні можливості завдяки використанню NER-моделі. Програмну реалізацію виконано за допомогою технології Node.js. Також використовувалась мова python для взаємодії з морфоаналізатором pymorphy2.

Для збільшення швидкодії методу, модель запам'ятовує нормалізовані терми та TF-IDF матриці кожного проаналізованого корпусу. Завдяки цьому швидкість аналізу корпусу з 12 документів скорочується з однієї секунди до однієї десятої секунди.

В подальшому розвитку засобів необхідно реалізувати вибір різних алгоритмів для тематичного моделювання, а також проводити детальніший аналіз унікальних термів. Також для більшої швидкодії об'ємних корпусів можна запровадити паралельну обробку даних.

Список літератури

1. worldwidewebsize.com [Електронний ресурс] : [Веб-сайт] – Режим доступу: www.worldwidewebsize.com (дата звернення: 28.05.2020)
2. w3techs.com [Електронний ресурс] : [Веб-сайт] – Режим доступу: www.w3techs.com/technologies/overview/content_language (дата звернення: 28.05.2020).
3. Глушков, Н. А. Анализ методов тематического моделирования текстов на естественном языке / Н. А. Глушков. — Текст : непосредственный // Молодой ученый. — 2018. — № 19 (205). — С. 101-103. — Режим доступу: https://moluch.ru/archive/205/50247/ (дата звернення: 28.05.2020).
4. pymorphy2.readthedocs.io [Електронний ресурс] : [Веб-сайт] – Режим доступу: www.pymorphy2.readthedocs.io (дата звернення: 28.05.2020).
5. Information retrieval document search using vector space model in R — 2017. — [Електронний ресурс]. Режим доступу: www.datasciencecentral.com/profiles/blogs/information-retrieval-document-search-using-vector-space-model-in (дата звернення: 28.05.2020).
6. Topic Modeling with LSA, PLSA, LDA & lda2Vec — 2018. — [Електронний ресурс]. Режим доступу: www.medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05 (дата звернення: 28.05.2020).

УДК 004.04

ОЛІЙНИК А.О.
БАКЛАН І.В.

ВИКОРИСТАННЯ ЛІНГВІСТИЧНОЇ МОДЕЛІ ДЛЯ ПОПЕРЕДНЬОЇ ОБРОБКИ ДАНИХ ЕКГ

В даній статті розглядаєтьсяметод попередньої обробки даних сигналів з використанням лінгвістичного моделювання. Описуються етапи виконання перетворення ЕКГ у вигляді числового ряду в лінгвістичний ланцюг з використанням апарату нечітких множин та розглядається етап створення словників на основі результатів.
Ключові слова:ЕКГ, лінгвістичне моделювання, попередня обробка даних, нечітка множина, лінгвістичний ланцюг, словник даних, сегментація сигналів, числовий ряд

In this article, themethod of pre-processing of signal data using linguistic modeling are reviewed. The phases of ECG conversion from time seriesto linguistic chains with fuzzy sets are described. The stage of creating dictionaries based onreceived results is considered.
Keywords:ECG, linguistic modeling, data preprocessing, fuzzy set, linguistic chain, data dictionary, signal segmentation, numerical series

Вступ

Аритмія серця є найпоширенішою та значною причиною захворюваності та смертності серед серцевих захворювань. Це група станів, які можуть характеризуватися порушеннями серцевого ритму. Рання діагностика має вирішальне значення для забезпечення вчасного лікування пацієнтів, які страждають цією хворобою. Традиційно діагностика проводиться шляхом огляду електрокардіограми (ЕКГ) кардіологом.

З розвитком технологій було зробленобагато досліджень, запропоновано та вдосконалено алгоритми для аналізу та класифікації сигналу ЕКГ. Методами та алгоритмами класифікації, які були запропоновані протягом останнього десятиліття, можна назвати цифровий аналіз сигналів, використання апарату нечітких множин, штучні нейронні мережі, приховану модель Маркова, генетичний алгоритм, опорні векторні машини, метод Баєса та інші.Усі вони демонструють власні переваги та недоліки. Такі алгоритми дозволяють проводити діагностування швидше та без впливу людського фактору. Проте для отримання точних результатів необхідно попередньо обробити дані, що будуть класифікуватися.

ЕКГ − найпоширеніший інструмент для вивчення функціональних можливостей роботи серця та діагностики кількох аномальних аритмій. Сигнал являє собою послідовність різних хвиль, таких, як P-хвиля, комплекс QRS, T- хвиля та U-хвиля. ЕКГ може містити кілька різних різновидів кожної хвилі (наприклад, комплекс QRS може мати різні види). В кожній мові використовуються слова і речення для обміну інформації Так само можна представити хвилі сигналу у вигляді слів, а їх сукупність, тобто повний сигнал, як речення. [1].

Виявлення інтервалів

Один цикл ЕКГ, який називається серцебиттям, окреслюється розташуванням сигналів P, QRS-комплексу та T, а також сегментами PQ та ST. Кожен сегмент відтворює певну стадію збудження міокарда: зубець P виникає при порушенні передсердь, комплекс QRS – скорочення шлуночків, зубець T з'являється при виході міокарда зі стану підйому.Аналіз ритму серця включає визначення регулярності та частоти серцевих скорочень, джерела збудження, а також оцінку функції провідності.

Існують різні форми попередньої обробки, необхідні для підвищення ефективності алгоритмів класифікаціїдля діагностування хвороби. Було прийнято кілька ключових способів попередньої обробки даних ЕКГ: пікове виявлення QRS та сегментація сигналів серцебиття.

Оскільки дані збираються протягом великого проміжку часу (8, 24, 48 годин), вони подаються у вигляді великого числового ряду сингалу ЕКГ. Тому необхідно поділити їх на інтервали. QRS - це найвизначніша форма сигналу на ЕКГ, яка є основою для майже всіх автоматичних алгоритмів діагностики ЕКГ. Оскільки вона ілюструє електричні імпульси всередині серця здебільшого під час скорочення шлуночків, цінні дані про фактичний стан серця дають частота його існування разом з його формою. Майже кожне виявлення піку R еквівалентно одному виявленню серцебиття.

Отже, перший етап – це виявлення Rпіків. Вони мають найбільше значення частот. Тому потрібно виділити моменти часу, які відповідають за R піки або наближені до їх значень. Далі вимірюється інтервал R-R - відстань між сусідніми R піками [1].

Наступним етапом є виявлення сегментація сигналів. Необхідно детально розбити сигнал на менші одиниці. Тобто, виявити положення P, QRS і T-хвиль. Для цього використовується Після виявлення R-піків, наявність інших будівельних хвиль (тобто, P, QRS і T-хвиль) в сигналі ЕКГ може бути виділена за допомогою адаптивних вікон пошуку. Для сегментації серцебиття можна ідентифікувати сегмент як фіксовану кількість зразків до розташування піку R до фіксованої кількості зразків після розташування піку R або від початку зубця P до зміщення послідовного T- хвиля.

Приклад визначення R-R інтервалу та сегментації показано на Рис.1.

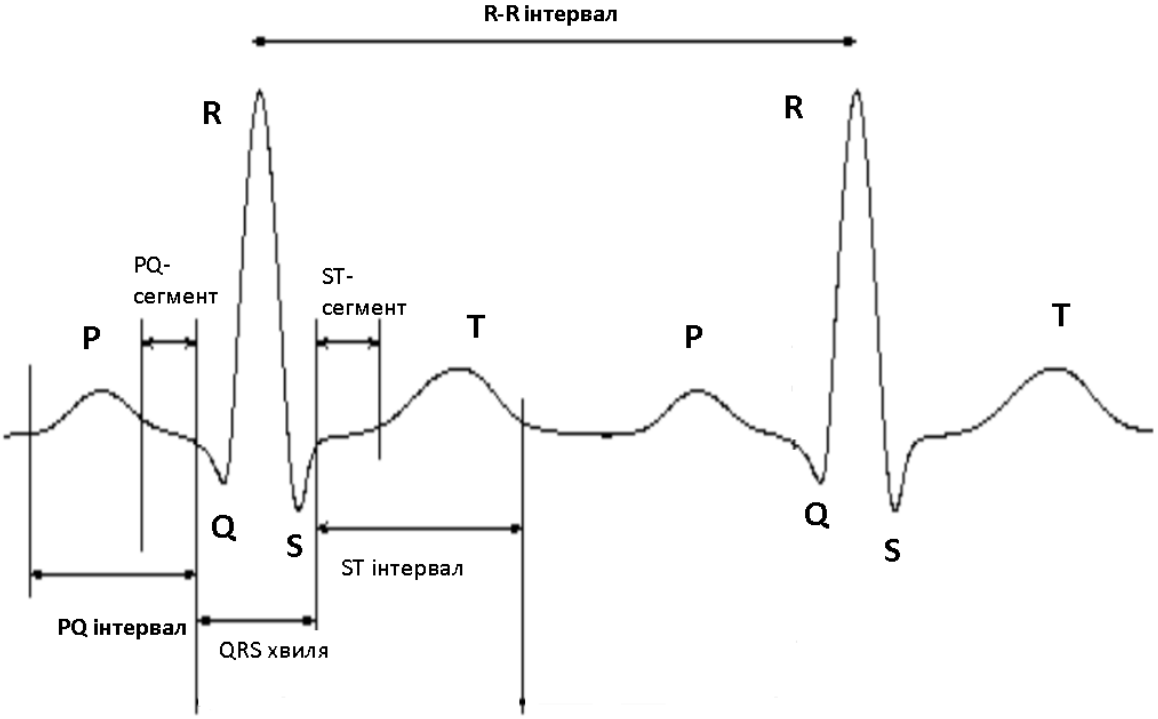


Рис.1. Визначення R-R інтервалу та сегментація сигналів ЕКГ[2]

Використання лінгвістичного моделювання

Щоб перетворити ЕКГ сигнал на структуру, схожу на речення природної мови, використовується метод лінгвістичного моделювання. Головною його метою є трансформація даних, представлених як числові ряди, в лінгвістичний ланцюг. Після цього робиться висновок про формальну граматику мови, якій ця послідовність належить, для подальшого проведення аналізу або створення словників [3][4]. Останнє дозволить значно прискорити порівняння даних та виявлення аномалій.

Для побудови лінгвістичної моделі необхідно виконати наступні етапи [5]:

1. попередньо обробити дані, звільнити від шумів, виявлення протиріч та дублікатів, отримання числового ряду даних, розбиття на проміжки;
2. розбиття ряду даних на рівні інтервали меншої довжини;
3. перетворення інтервалів на лінгвістичні ланцюги;
4. відновлення формальної граматики.

Перші кроки описані в попередньому пункті дослідження.

Для виконання другого етапу необхідно ввести певні поняття. Нехай маємо множину символів (алфавіт) $A = \{a_1, a_2, a_3, \dots, a_m\}$, за допомогою якого будемо складати «слова» сигналів.Даний крок полягає у розбитті великих проміжків на інтервали кількість N, де $N = |A|$. Отриманий інтервал позначається як$X = \{x_1, x_2, x_3, \dots, x_n\}$, рівні якого виміряні через однакові проміжки часу.

Науковою новизною даного алгоритму є додання процесу фазифікації (нечітких множин) в даний етап побудови лінгвістичної моделі. Математична теорія нечітких множин була запропонована Л. Заде, американським математиком і логіком з українським корінням,яка дозволяє описувати нечіткі поняття та знання, оперувати цими знаннями та робити нечіткі висновки [6].

Для виконання третього кроку необхідно визначити, що являє собою кожна нечітка множина. Вона являє собою набір правил, які визначають чи належить елемент множині, чи ні. Нехай такий набір позначається як P. Тоді загальний вигляд вона має такий:

$$U = \{(x, \mu_P(x)) | x \in P\} \qquad (1)$$

, де $\mu_P(x)$ – це функція приналежності елементу xнечіткій множині U.

$$\mu_P(x) = \begin{cases} 0, x \notin U \\ (0,1), x \text{ частково } \in U \\ 1, x \in U \end{cases} \quad (2)$$

Функція приналежності має задовольняти таким умові: $0 \leq \mu_P(x) \leq 1$.Якщо елемент не належить нечіткої множини, то функція набуває значення 0. Якщо елемент належить та зустрічається частіше інших в інтервалі, то вважається, що він повністю належить. Функція приналежності його дорівнює 1. Інші елементи, що належать чіткій множині, але зустрічаються рідше, будуть мати функцію, що належать проміжку (0,1).

Кожному символу алфавіту відповідає своя нечітка множина з відповідною функцією приналежності. Кожний елементx з інтервалу X з функцією приналежності $\mu_X(x)$ співставляємо символ з множини A. Таким чином, ряд $X = \{x1, x2, x3, \dots, xm\}$ перетворюється на ряд $L = \{l1, l2, l3, \dots, lm\}$, де $li \in A$. Результат перетворення сигналів в лінгвістичний ланцюг показано на Рис.2.

Останнім етапом є створення формальної граматики. На даному кроці виконується виявлення стійких фрагментів лінгвістичних ланцюгів. За допомогою цього формуються словники певних сигналів ЕКГ або навіть інтервалів чи масивів. Також виконується виявлення набору правил, що відображають стійкі зв'язки в отриманих лінгвістичних ланцюгах.

Створення словника

Останнім етапом для створення своєї мови обробки сигналів ЕКГ є формування словника. Даний крок виконується після розбиття числового ряду на інтервали, та перетворення їх в лінгвістичний ланцюг. Після виконання попередніх етапів робиться виявлення стійких фрагментів в лінгвістичних ланцюгах. Вони групуються за певними ознаками та зберігаються в словниковий запас. Для витягування даних достатньо лише виконати пошук інтервалу сигналу за певними ознаками. Якщо виконати вище описані етапи на сигналах ЕКГ, які діагностую як відхилення від норми, то можна створити словник аномальних даних. Далі для класифікації даних пацієнтів та діагностування достатньо буде лише порівняти надходженні сигналі зі словником аномалій. Це значно прискорить алгоритми та методи подальшої обробки інформації, а також зменшить ресурси, що необхідні для автоматичного діагностування [7].

Результати дослідження

Формалізовано процес попередньої обробки даних ЕКГ з виділенням максимальних значень R піків та сегментацію сигналу на менші хвилі. В результаті чого великий масив даних поділено на інтервали. Створено лінгвістичну модель шляхом відтворення елементів кожного інтервалу в певний символ на основі апарату нечітких множин, що відрізняє модель від попередніх досліджень [8]. Тобто, з певного числового проміжку отримуємо лінгвістичний ланцюг. На основі отриманих ланцюгів формуємо словник для подальшої обробки даних.

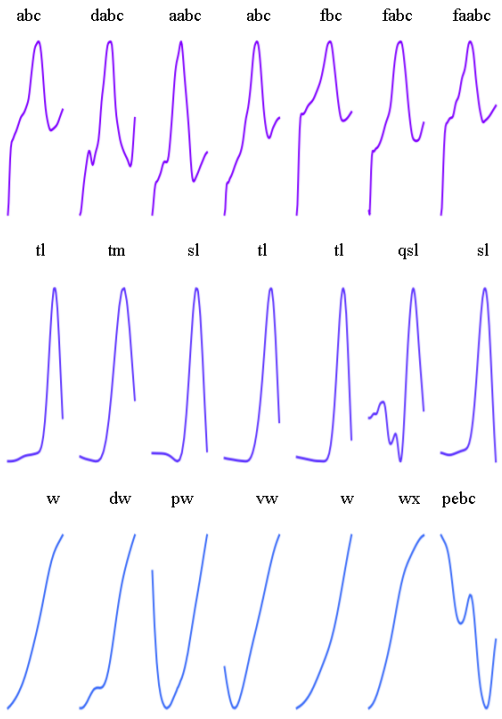


Рис.2. Перетворення сигналів ЕКГ в лінгвістичний ланцюг

Висновок

У цьому дослідженні було запропоновано метод попередньої обробки даних сигналів ЕКГ, який складається з трьох основних етапів:

1. розбиття великого масиву даних на інтервали;
2. відтворення отриманих інтервалів в лінгвістичний ланцюг;
3. створення словників стійких лінгвістичних ланцюгів.

Новизною дослідження є додання процесу фазифікації лінгвістичної моделі. Кожній літері алфавіту відповідає своя нечітка множина с відповідною функцією приналежності.

У результаті експерименту запропонований алгоритм відтворення сигналів ЕКГ за допомогою апарату нечітких множин в лінгвістичний ланцюг. Також створені словники на основі стійких фрагментів результатів для прискорення подальшої обробки даних та діагностування серцево-судинні хвороби.

Список літератури

1. Sajad Mousavi, Fatemeh Afghah, Fatemeh Khadem, and U. Rajendra Acharya ECG Language Processing (ELP): New Technique to Analyze ECG Signals, 2020, pp. 2-4
2. S.Karpagachelvi, Dr.M.Arthanari, M.Sivakumar. ECG Feature Extraction Techniques - A Survey Approach – 2010. arXiv preprint arXiv: 1005.0957
3. Baklan I. Linguistic approach for a time series anomaly detection – Slovac International Scientific Journal. – 2019. – №35, Vol. 1. – pp. 16-18
4. Баклан І. В. Лінгвістичне моделювання: основи, методи, деякі прикладні аспекти. Систем. технології. – 2011. – № 3. – с. 10-19.
5. Баклан І. В. Основні етапи побудови лінгвістичної моделі. – 2013. – Вестник ХНТУ № 2. – с. 10-19.
6. Заде Л.А.Понятие лингвистической переменной и его применение к принятию приближенных решений/ – М.: Мир, 1976. – 165 с
7. Nanyu Lib, Yujuan Si, Di Wang, Tong Liuc , Jinrun Yua. ECG Beats Fast Classification Base on Sparse Dictionaries – 2020. arXiv preprint arXiv: 2009.03792
8. Baklan, I., Oliinyk, Y., Mukha, I., Lishchuk, K., Gavrilenko, O., Ocheretianyi, O., & Tsytsyliuk, A. Adaptive Multistage Method of Anomalies Detection in ECG Time Series. COLINS. 2020.

УДК 004.93

ЗАКУПІН Є.О.
МАЖАРА О.О.

РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД УПРАВЛІННЯ ДЛЯ ПРИСТРОЇВ НА БАЗІ ARDUINO

У даному дослідженні проаналізовано підхід до розпізнавання голосових команд в умовах обмежених апаратних ресурсів. Представлено огляд різних підходів до розпізнавання голосових команд за допомогою нейронних мереж, описано їх переваги та недоліки. Розроблено прототипну систему розпізнавання голосових команд управління інвалідним візком. Представлено статистику отриманих результатів тестування системи.
КЛЮЧОВІ СЛОВА: розпізнавання голосу, нейронні мережі, Arduino.

This article analyzes the approach to voice command recognition in conditions of limited hardware resources. An overview of different processes of voice command recognition using neural networks is presented and the advantages and disadvantages of the chosen approach are described. Statistics of the result obtained when testing the system are also presented.
KEY WORDS: voice recognition, neural networks, Arduino.

1. Вступ

Забезпечення якості життя людей з обмеженими можливостями є актуальною проблемою як в усьому світі, так і в Україні зокрема. Провідними країнами створюються новітні розробки для спрощення побуту таких людей. Прикладом є проекти щодо вдосконалення спеціалізованих засобів, що полегшують повсякденні турботи. Нажаль, в Україні таким розробкам не надається значної уваги. В той же час, використання надбань зарубіжних колег часто обмежується фінансовим та/або мовним бар'єром. Необхідними є розробки орієнтовані на локальний ринок та потреби.

Дане дослідження спрямоване на створення рішення щодо спрощення побуту людей з обмеженнями в пресуванні за рахунок системи управління інвалідного візка з допомогою голосових команд. Таким чином утворюється можливість інтеграції машинного навчання у побут людей з обмеженими можливостями.До недоліків рішень, які наразі представлено можна віднести малий розмір словника команд тавикористання застарілих версії апаратних засобів. Проте головноюпроблемою існуючих рішень є те, що вони не передбачають використання української мови.Актуальною є задача створення системи, яка б не тільки використовувала сучасні апаратні та програмні засоби, але йнадавала б змогу підлаштовувати словник розпізнавання команд під людину, що буде її використовувати.

Важливою вимогою до системи є легкість навчання. Доцільним вбачається застосовувати систему, навчену на командах самого користувача.Це зумовлено тим, що система розробляється для людей з обмеженими можливостями, що можуть проявлятися не тільки в нездатності самостійно пересуватися.Нерідко також спостерігається пошкодження моторики верхньої частини тіла, що призводить до складнощів в управлінні візком руками, а також призводить до порушень мови, що ускладнює використання наборів, що були навчені заздалегідь. Також потребу в індивідуальному навчанні нейронної мережі викликає той факт, що більшість рішень має англомовну реалізацію, в той час як це є проблемою для громадян України.

Тому метою даної роботи є створення простого у реалізації рішення щодо розпізнавання україномовних голосових команд для сучасного, але недорогого апаратного забезпечення для подальшої інтеграції в систему керування інвалідним візком.

Для досягнення поставленої мети було виокремлено наступні завдання:

- провести огляд існуючих підходів до розпізнавання голосових команд;
- обґрунтувати засоби програмної та апаратної реалізації;
- обґрунтувати підхід до формування навчальної вибірки та способу класифікації команд;
- реалізувати прототипне рішення для підмножини команд.

2. Огляд існуючих рішень

Згідно з поставленою задачею були розглянуті існуючі методи розпізнавання командна базі Arduino. У загальному випадку існуючі рішення можна розділити на 3 види:

- використання готових модулів для розпізнавання команд. Прикладом такого підходу є модуль VoiceRecognitionModule v3.1, що підключається до плати та бере на себе роль розпізнавача[1].
- використання апаратної частини на базі Arduino у зв'язці з іншою технікою, на якій і буде проводитись розпізнавання (Комп'ютер чи Телефон). Прикладом такого підходу є програма для AndroidBTVOICECONTROLFORARDUINO, що використовує мікрофон телефону, та його ресурси для розпізнавання мови, а з платою спілкується за допомогою Bluetooth [2].
- використання самого контролеру для розпізнавання.В якості прикладу можна зазначити проект інтелектуальних інвалідних колясок MIT, що розробляє робот-інвалідний візок із голосовим керуванням. Цей проект обладнаний не лише системами розпізнавання голосу, але і датчиками для аналізу навколишнього середовища [3].

Через обмежені ресурси контролеру неможливо навчати систему на самому контролері.Тому попередньо до

використання пристрою у роботі з розпізнаванням команд, система має бути навчена, а вже потім отримані у навчанні файли додані до контролеру.

3. Засоби реалізації

В якості апаратної основи для проекту була обрана плата ArduinoNano 33BLE. Цей вибір зумовлений наступними факторами: подібні плати мають відносно низьку ціну, що робить їх доступними, що є важливою перевагою в даній системі; апаратна база Arduinoможе бути легко масштабованою. Це означає, що кінцевий продукт розробки можна розглядати як окремий модуль для використання у більш складних системах. Вищезазначене надає розробці властивості доступності та практичності, адже пристрій можна буде доповнювати іншими частинами, що будуть розширювати можливості загального пристрою, не викликаючи труднощів при компонуванні. В той же час даний підхід накладає певні обмеження на використання програмних засобів, адже подібні плати мають низьку розрахункову потужність, та малі об'єми оперативної та постійної пам'яті. Це призводить до неможливості використання складних систем, а також унеможливлює використання складних алгоритмів оптимізації. [4]

Враховуючи вищезазначене для подальшого дослідження було обрано бібліотеку Keras. Реалізацію розпізнавання пропонується виконувати за допомогою послідовних моделей (Sequential).

4. Послідовна модельKeras

Keras – це бібліотека для побудови моделей нейронних мереж на основі високорівневого програмного інтерфейсу. Вона дозволяє визначати вузли нейронної мережі різних типів та поєднувати їх у шари відповідно до обраної архітектури. Найбільш поширеною моделлю є послідовна - tf.keras.Sequential.

В процесі побудови послідовної моделі користувачеві надаються різноманітні можливості налаштування. Серед них основним параметром, який впиває на ефективність є тип функції активації (activation) В даному дослідження в якості функції активації було обрано relu та softmax.

Після того як модель сконструйована, необхідно налаштувати процес її навчання викликом методу compile. tf.keras.Model,для якого необхідно вказати в якості параметру обраний алгоритм оптимізації (optimizer), функцію витрат (loss) та метрики оцінки якості алгоритму (metrics). [5]

В якості алгоритму оптимізації використовується Adam. Keras надає реалізацію цього оптимізатору з можливістю виклику за ключовим словом 'adam' та оптимізатору (optimizer). Функція витрат представлена наступним чином:

$$J(\theta) = \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} \text{ де } \widehat{m}_t = \frac{m_t}{1-\beta_1^t}, \widehat{v}_t = \frac{v_t}{1-\beta_2^t},$$

Де $J(\theta)$– функція витрат, m_t–функція обновлення ваг,v_t–експоненціальнековзаюче середнє, β_1,β_2–коефіцієнти затухання, □ - ненульове значення.

5. Оптимізатори нейронних мереж

Бібліотека Kerasнадає можливість використати для навчання один з базових оптимізаторів. В роботі розглядались наступні алгоритми та їх особливості.

Стохастичний градієнтний спуск, який підтримує єдину швидкість навчання (так звану альфа) для всіх оновлень ваги.Швидкість навчання не змінюється під час тренування, а підтримується для кожної ваги мережі (параметра) і окремо адаптується по мірі навчання.Метод обчислює індивідуальні адаптивні швидкості навчання для різних параметрів з оцінок першого і другого моментів градієнтів.

Середньоквадратичне поширення (RMSProp) також підтримує швидкості навчання по кожному параметру, які адаптовані на основі середнього значення останніх величин градієнтів для ваги (наприклад, як швидко воно змінюється). Це означає, що алгоритм добре працює онлайн і з нестаціонарними завданнями (наприклад, з шумом).

Адаптивний Градієнтний Алгоритм (AdaGrad)підтримує швидкість навчання за параметром, яка покращує продуктивність при проблемах з розрідженими градієнтами (наприклад, проблеми з природною мовою і комп'ютерним зором).

Адам (Adam) - це алгоритм оптимізації, який можна використовувати замість класичної процедури стохастичного

градієнтного спуску для ітеративного поновлення ваг мережі на основі навчальних даних.Адам також використовує середнє значення других моментів градієнтів (нецентрованого дисперсія).

Для подальшого використання в даному дослідженні було обрано саме Adam алгоритм виходячи з наступних його переваг:

- простота реалізації.
- ефективність
- незначні вимоги до пам'яті.
- зарекомендований для задач, які є можуть вважатися великими з точки зору даних і / або параметрів;
- гіперпараметри мають інтуїтивно зрозумілу інтерпретацію і зазвичай вимагають незначних налаштувань.

6. Представлення даних

В системах розпізнавання звукових сигналів широко зарекомендувало себе представлення навчальної вибірки у вигляді мел-кепстральних характеристик – MFCC (MFCC, Mel-frequency cepstral coefficients).

Мел – це психофізична одиниця висоти звуку, пов’язана з частотою (в герцах) як

$$mel = 2595 \log_{10}(1 + \frac{f}{700}),$$

а кепстр є результатом взяття оберненого перетворення Фур’є від логарифма спектру сигналу, тобто нелінійним «спектром спектру» [6].

Базовою концепцією цього методу є максимальне наближення вхідної інформації системи розпізнавання до тієї, що надходить в слуховий аналізатор мозку людини. До переваг такого представлення також варто віднести стійкість до шумів та інформативність для розпізнавання фонем. Розрахунок коефіцієнтів відбувається за наступними базовими кроками[7]:

- розбиття сигналу на фрейми (вікна)
- зважування кожного фрейму віконною функцією
- розрахунок амплітуди спектру для кожного фрейму шляхом застосування дискретного перетворення Фур’є.
- перехід в мел-шкалу
- згортка трикутними вікнами з перекриттям
- перехід в лінійну шкалу
- дискретне конусне перетворення

Цей процес може мати варіації, наприклад: відмінності у формі або інтервалі вікон, що використовуються для нанесення масштабу.

В дані роботі для формування навчальної вибірки було здійснено аудіозаписи базових команд. Перетворення записів в MFCC виконувалось з використанням бібліотеки Librosa.

7. Реалізація

Таким чином для реалізації було обрано послідовну модель нейронної мережі.В якості оптимізатору використовується алгоритм Adam.

На вхід мережі подаються MFCC характеристики звукового запису. На виході отримується клас команди.

Було вирішено, шо довжина звукової доріжки в навчальній вибірці буде складати дві секунди. Цієї довжини достатньо для вимовляння необхідних команд. Крім того, використання коротших команд зумовлює зменшення використання пам’яті та часу відгуку системи.

В першій реалізації роботи використовуються три класи: «вперед», «назад» та «шум».

Апаратна реалізація складалась з плати Arduino Nano 33BLE, а також електричного двигуна, що імітував при тестуванні інвалідний візок.

Після успішного навчання системи, будуються файли у форматі бібліотеки для середи розробки Aduino, яка в свою чергу підключається до проекту програмного коду, що буде завантажений на плату.

Плата циклічно зчитує звук з вбудованого мікрофону та починає аналіз отриманого звукового ряду.

В результаті тестування система показала свою працездатність. Було отриманостатистичні дані щодо ефективності системи, які представлені у таблиці 1.

В роботі використано та проаналізовано дві навчальні вибірки. Перша мала 15 елементів для кожного класу, друга - 100. Формування вибірок здійснювалось з розрахунку на те, що більшість аудіозаписів належали майбутньому користувачу. Для забезпечення такого навчання в майбутньому пропонується поширювати

розробку за вільною ліцензією з зручним програмним інтерфейсом додаткового навчання моделі.

Дані показують, що зі збільшенням вибірки зростає кількість необхідної пам’яті, при цьому більша вибірка, має більшу точність розпізнавання.

Табл 1. Результати експеременті

Кількість елементів	Точність, %	Втрати	Час взаємодії, с	Максимальна задіяна оперативна пам’ять, kB	Використання постійної пам’яті, kB
15	81.1	0.45	66	12	29.9
100	92.3	0.23	55	15.6	35.5

Висновки

Таким чином, розроблене рішення дозволяє розпізнавати дві голосові команди, використовуючи заздалегідь навчену мережу, що в подальшому використовується на платі з обмеженими ресурсами. Основною перевагоює використання власної вибірки, що дозволяє налаштуватисистему під користувача. Додатковою перевагою є те, що система використовує невелику але сучасну плату, що здатна легко взаємодіяти з іншими платами того ж сімейства, що забезпечує універсальність та масштабованість. Система на практиці показала свою працездатність згідно з статистичними характеристики. В майбутньому система може бути покращена за рахунок збільшення навчальної вибірки, що збільшить точність розпізнавання,а також за рахунок збільшення словника, що позитивним чином вплине на практичність системи.

Список літератури

1. Introduction to Voice Recognition WithElechouse V3 and Arduino[електроннийресурс] – Режимдоступу:https://www.instructables.com/Introduction-to-Voice-Recognition-With-Elechouse-V/

2. Голосовое управление посредством Arduino[електронний ресурс] – Режим доступу:https://future2day.ru/golosovoe-upravlenie-posredstvom-arduino/

3. MIT Intelligent Wheelchair Project: A Voice-Commandable Robotic Wheelchair [електронний ресурс] – Режим доступу:https://techtv.mit.edu/videos/6465-mit-intelligent-wheelchair-project-a-voice-commandable-robotic-wheelchair

4. David Kushner (2011-10-26). "TheMakingofArduino". IEEE Spectrum.

5. Backend utilities[електронний ресурс] –Режим доступу: https://keras.io/api/utils/backend_utils/

6. ТЕОРІЯ ТА МЕТОДИ ОБРОБЛЕННЯ СИГНАЛІВ. // ISSN 1990-5548 Електроніка та системи управління. – 2012. – С. 5–9.

7. АНАЛІЗ ВПЛИВУ ПАРАМЕТРІВ ОБРОБКИ ЗВУКОВОГО СИГНАЛУ НА ЯКІСТЬ РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД. // Вісник Національного технічного університету України «КПІ». – 2014. – С. 34–41.

УДК 004.02

ЯДЕЛЬСЬКИЙ Р.І.
ЛІЩУК К.І.

ВИЯВЛЕННЯ НЕПОЛАДОК В ПРОЦЕСІ 3D ДРУКУ ЗА ДОПОМОГОЮ АВТОМАТИЗОВАНОГО ВІЗУАЛЬНОГО МОНІТОРИНГУ

У даній статті розглянуто види неполадок в процесі 3D друку та запропоновані методи їх автоматичного виявлення, а саме метод з використанням класичних підходів комп'ютерного зору, комбінацію комп'ютерного зору і аналізу3D моделі об'єкта, та виявлення характерних артефактів, що сигналізують про несправність друку з допомогою згорткових нейронних мереж.
КЛЮЧОВІ СЛОВА: 3D ДРУК, МОНІТОРИНГ, КОМП'ЮТЕРНИЙ ЗІР, НЕЙРОННІ МЕРЕЖІ.

This article discusses problems in the process of 3Dprinting and proposes methods of their automatic detection, namely the method using classical approaches of computer vision, a combination of computer vision and analysis of 3D models, detection of specific artifacts that signal a printing failure using convolutionalneural networks.
KEY WORDS: 3D PRINTING, MONITORING, COMPUTER VISION, NEURAL NETWORKS.

1. Вступ

Протягом останніх років настільні 3D-принтери стали достатньо доступними та розповсюдженими. Однак, попри свою поширеність, експлуатація таких 3D принтерів є достатньо складною так як у процесі друку з великою вірогідністю може виникнути несправність. Залежно від розміру, складності моделі та обраної роздільної здатності під час генерації інструкцій для принтеру, процес 3D-друку може зайняти від пари годин до декількох днів. Це означає, що машина іноді працюватимевеликий проміжок часу без людського контролю. За цей час багато речей може піти не так. Наприклад, сопло, через яке видавлюється матеріал, може бути закупоритись, або виріб може відлипнути від друкарського столу.Це може привести до марнування великої кількості матеріалу, значних часових затрати та до пошкодження принтеру. Якщо проблему виявлено на ранньому етапі, багатьох наслідків можна уникнути. Зараз єдиним способом виявлення цих помилок є ручна перевірка процесу друку людиною. Метою цієїроботи є класифікація найпоширеніших проблем у FDM 3D-друці та аналіз кожної з них, щоб перевірити, чи можливо їх автоматичне виявлення, за допомогою візуального моніторингу в режимі реального часу.

2. Аналіз несправностей в процесі 3D друку

В процесі 3D друку задіяна досить велика кількість процесів, коректність проведення яких не відслідковується автоматично, тому часто виникають різного виду неполадки. Для того, щоб їх успішно виявляти слід описати причини їх появи та їх ознаки. Основні типи неполадок перелічені далі.

Найпоширеніша помилка і найпростіша у виявленні - це від'єднання об'єкта від поверхні друку.[1] Об'єкт повинен бути міцно закріплений на поверхні для успішного друку, в іншому випадку переміщення сопла призведе до зміщення об'єкту та неправильної викладки матеріалу. На рисунку 1 ця ситуація представлена двома ескізами. Смугастий прямокутник із закругленими кутами представляє об'єкт для друку та прикріплюється до поверхні друку (представлена товстою чорною смугою під об'єктом).На ескізі відрив видно, оскільки об'єкт перемістився вправо, а початкове положення позначено пунктирними лініями. Зміщення між початковим та фактичним положенням називається помилкою переміщення.

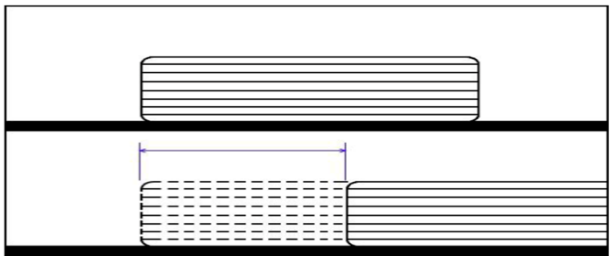


Рис. 1. Відрив об’єкта від поверхні друку

Другою найбільш частою проблемою є зупинка постачання матеріалу в процесі друку. При цьому типі помилки матеріал не протікає через сопло друкуючої головки. Друкуюча головка рухається вздовж попередньо визначеного шляху інструменту без видавлювання нитки матеріалу. Висота об’єкта залишається незмінною, а об’єкт рухається вздовж вісі Z вниз. Об’єкт не завершений і не буде завершений. Якщо потік матеріалу буде перервано лише тимчасово, об’єкт буде надрукований з дефектом, оскільки один або кілька шарів матеріалу відсутні. Якщо потік матеріалу перерваний лише ненадовго, об’єкт може бути повністю надрукований лише з незначними дефектами, оскільки шари можуть компенсувати незначну частину нижнього шару.На рисунку 2 зображено цей тип помилки. Об'єкт відображається у вигляді смугастого прямокутника із закругленими кутами, друкарська поверхня - у вигляді товстої чорної лінії під об'єктом друку, а друкуюча головка або екструдер - у вигляді білого прямокутника із смугастим п'ятикутником внизу. У верхній частині ескізу показано правильне позиціонування екструдера, а у нижній –ситуацію, яка виникає при зупинці подачі матеріалу.

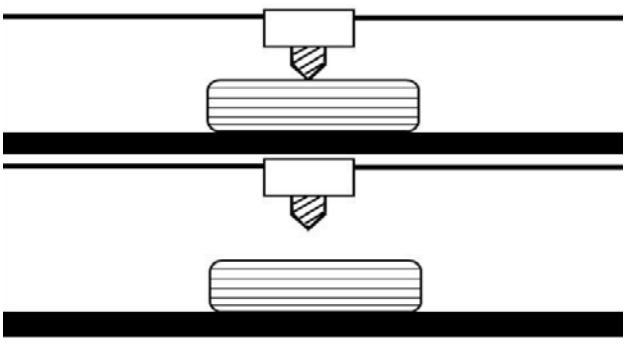


Рис. 2. Припинення подачі матеріалу

Ще одною розповсюдженою несправністю є зсув шарів в процесі друку. [2] Ця проблема виникає при несправності системи принтера, що відповідає за рух екструдера, наприклад заїдання мотора. Приклад такої несправності наведений на рисунку 3. Зверху зображена нормальна укладка матеріалу, а знизу – при зсуві шарів.

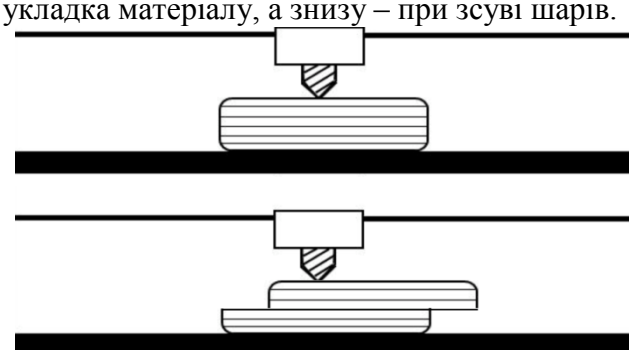


Рис. 3. Зсув шарів

3. Методи виявлення несправностей друку за допомогою аналізу візуальної інформації

Оскільки конструкція 3D принтерів не дозволяє реалізувати механічний контроль якості друку, єдиним доступним способом моніторингу є візуальний контроль з допомогою відео камер.Для підвищення якості моніторингу необхідно використовувати кілька камер, що дозволить використовувати алгоритм триангуляції для розпізнавання позиції об’єкта, що друкується, однак це потребує точного позиціонування системи камер, чого важко досягнути при експлуатації настільних 3D принтерів, тому для виявлення неполадок методом, описаним у даній статті використовується єдина камера та джерело світла, що встановлені статично та напрямлені на поверхню для друку, як зображено на рисунку 4.Кольорова камера знімає зображення, освітленої моделі після додавання кожного нового шару матеріалу. Для того, щоб спростити аналіз зображень,програму управління принтером модифіковано таким чином, щоб при завершенні друку кожного шару друкуюча голівка переміщалась у заздалегідь задану точку у горизонтальній площині, щоб не перекривати модель на зображенні, після чого власне і відбуватиметься створення фото. Це дозволяє позбутись необхідності постійного відслідковування положення друкуючої голівки та нівелювати вплив її руху на аналіз. Таким чином єдина різниця

між зображеннями двох сусідніх шарів буде зумовлена власне викладеним на останньому шарі матеріалом.

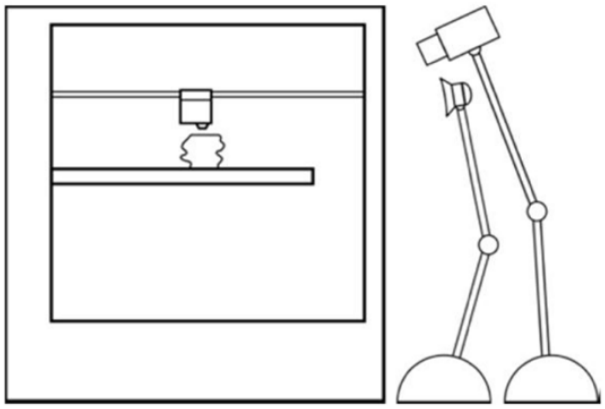


Рис. 4. Встановлення камери та джерела світла для спостереженням за процесом друку

Для виявлення неполадок уданій статті розглянуто наступні методи:

- Використання класичного підходу комп'ютерного зору та виявлення помилок базуючись на аналіз різниці між зображеннями шарів моделі
- Комбінація підходів класичного комп'ютерного зору та аналізу комп'ютерної моделі виробу, що друкується для виявлення різниці між реальними та симульованими девіаціями у зображеннях
- Аналіз зображень з використанням згорткових нейронних мереж для виявлення характерних артефактів, що сигналізують про неполадку в процесі друку.

4. Метод виявлення несправностей друку з використанням класичних підходів комп'ютерного зору

Для коректного виявлення несправності на зображенні необхідно відділити об'єкт від фону зображення. При сегментації об'єкту із захопленого зображення використовується колір матеріалу для друку. Оскільки цей підхід спрямований на недорогі споживчі принтери, матеріалом, як очікується, є монохромна пластмаса PLA. Камера робить кольорові зображення за допомогою техніки фільтра Баєра. Колір кожного пікселя представлений трьома інтенсивностями, по одній для кожного з червоного, зеленого та синього кольорів (RGB). У поєднанні ці три кольори можуть імітувати більшість сприйманих людиною кольорів. Хоча це добре для людей, це кольорове зображення не є ідеальним для алгоритмічної сегментації кольорових предметів, тому для сегментації проводиться перетворення кольорового простору з RGB на відтінок, насиченість і значення (HSV). Простір HSV представляє колір як набір з трьох значень: чистий коліряк кут від 0 ° до360 ° на крузі кольорівінасиченість та інтенсивність як значення від0 до1. [3]

Маючи піксель зі значенням у просторі RGBзі значеннями$r, g, b \in [0, 255]$ спершу відбувається їх нормалізація:

$$r' = r \,/\, 255$$
$$g' = g \,/\, 255$$
$$b' = b \,/\, 255$$

Після цього визначається максимальна та мінімальна компонента і вираховується їх різниця:

$$C_{max} = \max(r', g', b')$$
$$C_{min} = \min(r', g', b')$$
$$\Delta = C_{max} - C_{min}$$

Ці значення використовуються для обчислення компонент в представленні HSV:

$$h = \begin{cases} 60 * \mathrm{mod}\left(\frac{g-b}{\Delta}, 6\right) & C_{max} = r' \\ 60 * \left(\frac{b-r}{\Delta} + 2\right) & C_{max} = g' \\ 60 * \left(\frac{r-g}{\Delta} + 4\right) & C_{max} = b' \end{cases}$$

$$s = \begin{cases} 0 & C_{max} = 0 \\ \frac{\Delta}{C_{max}} & C_{max} > 0 \end{cases}$$

$$v = C_{max}$$

Після того, як зняті зображення переведені в кольоровий простір HSV пластик сегментується по ознаці належності до діапазону$H_t = [h_{min}, h_{max}]$, що відповідає діапазону для заданого матеріалу. Крім того, також відкидуються пікселі зі значенням компонент *s*та *v*менших за порогові значення T_sтаT_vвідповідно для видалення темних пікселів, що не несуть корисної інформації. Таким чином, для створення сегментаційної маски використовується наступна формула:

$$S = \begin{cases} 1 & h \in H_t, s > T_s, v > T_v \\ 0 & \text{в іншому випадку} \end{cases}$$

Коли сегментація проведена, для виявлення несправності в процесі друку визначається середньокватичнадевіація між зображеннями двох сусідніх шарів під номерами*l* та *l-1*:

$$RMSE = \sqrt{\frac{\sum_{x=1}^{X_{max}} \sum_{y=1}^{Y_{max}} (S_{x,y,l-1} - S_{x,y,l})^2}{X_{max} * Y_{max}}}$$

Якщо дане значення більше за зазначене порогове значення E_t, система сигналізує про ймовірну несправність друку:

$$Error = \begin{cases} 1\ RMSE > E_t \\ 0 RMSE \leq E_t \end{cases}$$

5. Метод виявлення несправностей друку з використанням комп'ютерного зору та аналізу комп'ютерної моделі виробу

При використанні методу, описаного у минулому пункті найскладнішою задачею є правильний підбір граничного значення E_t, так як оптимальне його значення залежить від очікуваної кількості матеріалу, що викладається на кожному шарі об'єкта та положення камери. Таким чином для різних моделей, положень камери, і навіть для різних шарів одної моделі це значення повинно змінюватись. Для того, щоб автоматизувати обрахунок оптимального значення E_t можна використовувати комп'ютерну модель виробу, що друкується. Нехай товщина одного шару моделі H_l, тоді висоту частини моделі, що вже надрукована на шарі *l*можна визначити за наступною формулою:

$$h_l = H_l * l$$

Відітнувши верхню частину моделі горизонтальною площиною, що знаходиться на висоті h_l отримуємо комп'ютерну модель виробу, що в даний момент знаходиться на поверхні для друку, яка використовується для створення комп'ютерно генерованого зображення виробу. Щоб створити зображення слід врахувати взаємне положення між камерою та об'єктом після чогодо моделі застосовується перспективна проекція: [4]

$$S = \frac{1}{\tan\left(\frac{fov}{2} * \frac{\pi}{180}\right)}$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{bmatrix} S & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & -\frac{f}{(f-n)} & -1 \\ 0 & 0 & -\frac{f*n}{(f-n)} & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$u = \frac{x'}{z'} v = \frac{y'}{z'}$$

У формулах наведених вище fovце кут огляду камери, *f*та *n*– це відстані до дальньої та ближньої обтинаючоїплощин відповідно, що необхідні для коректної роботи процесу рендерингу зображення з використанням програмного комплексу OpenGL. Оскільки згенероване зображення буде застосовується як бінарна маска для його генерації не застосовується модель освітлення. Для кращого позиціонування моделі у просторі відносно камери використовується порівняння реального та згенерованого зображення виробу.

Після отримання комп'ютерно зненерованого зображення виробу, що друкується на шарі *l*визначається прогнозована девіація та порівнюється з девіацією на реальних зображеннях.Якщо вони відрізняються більш ніж заздалегідь заданий допуск ε, то система сигналізує про помилку друку:

$$E = \frac{\left|RMSE_{cam} - RMSE_{gen}\right|}{RMSE_{gen}}$$

$$Error = \begin{cases} 1\ E > \varepsilon \\ 0 E \leq \varepsilon \end{cases}$$

Крім описаного вище методу, за умови точного позиціонування камери, є можливість визначення девіації між згенерованим та реальним фото, що дозволить виявляти неточності в геометрії виробу.

6. Метод виявлення несправностей друку з згорткових нейронних мереж

При виникненні неполадок у процесі друку часто подача матеріалу не припиняється, натомість він видавлюється просто в повітря утворюючи заплутані клубки характерного вигляду. Ідея цього методу полягає у виявленні таких клубків з допомогою згорткової нейронної мережі. Згорткові нейроонні мережі (англійською

CNN) в машинному навчанні — це клас глибинних штучних нейронних мереж прямого поширення, який успішно застосовувався до аналізу візуальних зображень. [5]ЗНМ складається з шарів входу та виходу, а також із декількох прихованих шарів. На відміну від звичайних нейронних мереж, шари згорткових нейронних мереж є багатовимірними. Нейрони одного шару пов'язані тільки з певною областю наступного шару. Перший вхідний шар мережі має розмірність $w \times h \times d$, де w — шири-на зображення, h — висота, d — кількість каналів кольору. Кожен шар такої мережі трансформує отриманий вхід, використовуючи диференційовану функцію. В згортковій нейронній мережі в основномувикористовуються три типи шарів: згорткові, агрегуючі, повнозв'язні.

Для навчання нейронної мережі необхідна велика кількість зображень, на яких наявні описані вище клубки матеріалу. Оскільки у вільному доступі таких даних немає, вони були створені штучно, з допомогою накладання на відео процесу друку випадково генерованих клубків матеріалу. Для генерації випадковим чином обирається кількість точок ламаноїN, після чого координати кожної точки тривимірної ламаної обираються випадково, однак з умовою їх розміщення у сфері радіусом R.Отримана тривимірну ламана візуалізується у двовимірне зображення з урахуванням тіней від джерела світла. Згенероване зображення накладається на кадр відео з реальної зйомки процесу друку і використовується для навчання нейронної мережі.

7. Результати досліджень

Метод, описаний у секції5, був застосований до тестової друкованої геометрії, що містить літери "DTU", як показано на малюнку 5. Ця геометрія є складною для друку на принтерах на основі FDM через непідтримувані звиси. Формування опорної структури було вимкнено, щоб продемонструвати виявлення несправного друку.

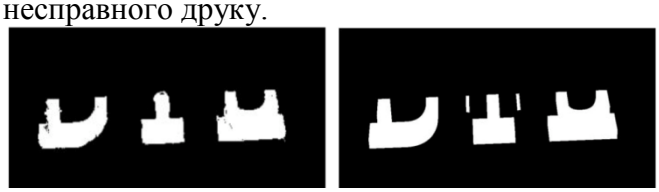


Рис. 5. Результат сегментації реального зображення виробу (зліва) та комп'ютерної моделі (справа)

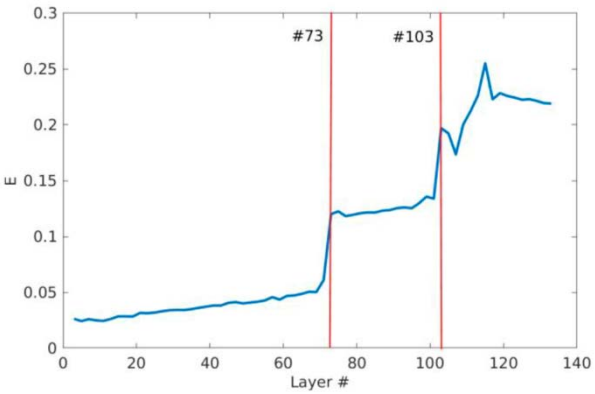


Рис.6.Графік помилки значення в залежності від шару друку

Як видно сегментація відбувається достатньо точно. На графіку зображеному на рисунку 6видно кілька стрибків значення помилки на шарах під номерами 73 та 103, система відреагувала на них успішносигналізувала про помилку друку.

Висновки

У даній роботі було розглянуто основні класи неполадок в процесі 3D друку та методи їх виявлення. Наявні експериментальні результати свідчать про ефективність запропонованих методів. У подальшому планується модифікація та комбінування описаних методів з метою їх практичного застосування для моніторингу 3D друку.

Список літератури

1. Kadir G. Common fdm 3d printing defects / G. Kadir, T. Halit., 2018. – 42 c.
2. Alsoufi M. Warping deformationof desktop 3d printed parts manufactured by open source fused depo-sition modeling (fdm) system. / M. Alsoufi, A. El-Sayed. // International Journal of Mechanical &Mechatronics Engineering. – 2017. – C. 7–16.
3. AR S. Color gamut transform pairs, Computer Graphics / Smith AR., 1978. – 103 c.
4. Hartle R. Multiple view geometry incomputer vision / R. Hartle, A. Zisserman. – Cambridge university press, 2003. – 334 c.
5. Alex K. Imagenetclassification with deep convolutional neural networks / K. Alex, S. Ilya, H. GeoffreyHinton. – Pereira: CurranAssociates, Inc., 2012. – 1305 c.

УДК 004.9

СБОРИК А. Ю.,
ТЄЛИШЕВА Т.О.

РЕКОМЕНДАЦІЙНА СИСТЕМА ПІДБОРУ АВТОМОБІЛІВ ДЛЯ ПРОДАЖУ КЛІЄНТАМ

Проаналізовано існуючі методи знаходження рекомендацій для клієнтів та наведено розроблений алгоритм формування рекомендацій. Побудована рекомендаційна система на основі розробленого алгоритмувикористовується з метою підвищення ефективності процесу створення рекомендацій робітниками компанії з продажу автомобілів. Система дозволяє знаходити рекомендовані автомобілі на основі побажань клієнтів.

КЛЮЧОВІ СЛОВА: АВТОМОБІЛЬ, РЕКОМЕНДАЦІЙНА СИСТЕМА, АЛГОРИТМ ПОШУКУ РЕКОМЕНДАЦІЙ, CRM СИСТЕМА

The existing methods of finding recommendations for clients are analyzed and the developed algorithm of forming recommendations is given. The built recommendation system on the basis of the developed algorithm is used for the purpose of increase of efficiency of process of creation of recommendations by workers of the company on sale of cars. The system allows you to find recommended cars based on customer wishes

KEYWORDS: CAR, RECOMMENDATION SYSTEM, RECOMMENDATION SEARCH ALGORITHM,CRM SYSTEM

1. Постановка проблеми:

Сучасний глобальний ринок автомобілів є однією із важливих складових економіки світу, одним з найконкурентніших ринків. Продаж автомобілів напряму позначається на економічному зростанні та кризових явищах глобальної економіки[1].

Світовий авторинок у вересні поточного року підвищився на 0,5% до 7647000 легкових автомобілів. Найбільшу кількість автомобілів за звітний період реалізували в КНР з показником в 2,487 млн од., що на 9,3% вище за минулорічний результат. Американські автолюбителі придбали 1,337 млн автомобілів, що на 4,4% більше, ніж в 2019 році. Західноєвропейськіавторинки продемонстрували результат в 1,367 млн автомобілів (+1,3%). Реалізація автомобілів в Східній Європі підвищилася на 19,7% до 393,3 тис. шт.[2].

Через високий попит на даний товар автосалони, компанії, що займаються продажами автомобілів отримують велику кількість звернень від клієнтів на покупку автомобілів. Відповідно робітники відділу продажу отримують та аналізують побажання кожного клієнта та, зазвичай, власноруч підбирають найкращі варіанти для клієнтів. Проте такий процес підбору займає досить багато часу і залежить від кваліфікації робітника, що знаходить рекомендації, тобто знайдені рекомендації залежать від людського фактору і не завжди можуть бути правильними. Технології опрацювання інформації використовують найширший та найсучасніший спектр технічних засобів таких, як CRM системи для збереження та управління даними, а також для управління процесом по роботі з клієнтом. Тому обраний напрям дослідження і створення рекомендаційної технології та системи є актуальним.

Аналіз останніх досліджень та публікації, на які спирається автор.

Рекомендаційні системи є досить популярними, тому існує багато робіт присвячених даній тематиці. У статтях [3-5] розглядаються методи знаходження рекомендацій, що базуються на аналізі профілів та попередньої активності користувачів в системі. У розглянутих публікаціях не було досліджено ситуацію, коли потрібно на основі побажань користувача надати рекомендацію серед існуючих авто в системі. Дана ситуація є актуальною, оскільки існує багато випадків, коли користувач не має ні профілю, ні попередньої історії дій та хоче отримати рекомендацію.

Тому виникає необхідність подальшого дослідження в цьому напрямі та в розробці рекомендаційної системи.

Метою проєкту з розробки рекомендаційної системи підбору автомобілів є підвищення ефективності роботи менеджерів, що в цілому збільшить прибуток компанії за рахунок збільшення загальної кількості оброблених заявок і продаж.

2. Концепція системи:

Процес знаходження рекомендацій починається з моменту заповнення клієнтом заявки на знаходження рекомендацій. Клієнт в веб-застосунку заповнює форму заявки, в якій він вказує персональні дані для подальшої комунікації з ним, а також обирає бажані характеристики автомобіля. Після чого дані автоматично потрапляють до CRM системи. Робітник починає роботу з клієнтом і система автоматично знаходить рекомендовані автомобілі і відображає їх у вигляді таблиці.

3. Розробка алгоритму створення рекомендацій:

Існує декілька основних підходів до знаходження рекомендацій:

- фільтрація вмісту даних;
- колаборативна фільтрація даних.

Для побудови рекомендаційної системи використано підхід фільтрації вмісту даних [3]відповідно до його особистих запитів. Елементами в алгоритмі знаходження рекомендацій вважаються автомобілі, наявні в системі, користувачами – запити клієнтів, що містять бажані характеристики автомобілів.

Першим етапом роботи алгоритму є перетворення текстових характеристик бажаних параметрів для автомобілів, отриманих з заявки клієнта, та характеристик автомобілів на вектори слів, індексація та нормалізація отриманих векторів. Перший етап роботи алгоритму наведено кроками 1-5 на блок-схемі роботи алгоритму.

Другим етапом роботи алгоритму є саме створення рекомендацій. Процес створеннярекомендацій відбувається до тих пір поки не знайденакількість автомобілів, які відповідають запиту. Другий етап роботи алгоритму наведено кроками 6-12 на блок-схемі роботи алгоритму.

Для знаходження списку елементів (автомобілів), що підходять користувачеві фільтруємо елементи, що описані векторами атрибутів, таким чином щоб вони задовольняли перевагам конкретного користувача. У цьому випадку основна ідея алгоритму полягає в тому, щоб знайти схожість між об'єктом - користувачем та елементами, що мають різну семантику, але перебувають у тому самому Евклідовому *n*-вимірному просторі.Кореляція між об'єктом - користувачем та векторами атрибутів елементів представлена як відстань. Показник подібності, представлений Евклідовою відстанню між атрибутамизапитів користувачів та атрибутами елементів, обчислюється на основі найбільш подібних атрибутів (наприклад, що мають найближчі відстані між собою)[4]. Фактично отримуємо схожість між елементами-користувачами та елементами, обчислюючи суму квадратів часткових відстаней між окремими атрибутами.

Алгоритм знаходить елементи, що найбільше підходять конкретному елементу-користувачеві, за вектором атрибутів $\overline{U}$(вектором переваг користувача), та формує центр кластеру, в якому конкретний користувач, пов'язаний з кількістю елементів . Подібність користувачів та елементів представлена Евклідовою відстанню між векторами атрибутів користувача $\overline{U}$ та атрибутів елемента $\bar{I}$. Подібність знайденаза формулою Кількість атрибутів вектору користувача та векторів автомобілів є однаковою.

$$S_w^2(\overline{U}, \bar{I}_t) = \sum_{n=1}^{N_p} \left(\frac{1}{r_n} * [i_n^{(t)} - U_n]^2\right), (1)$$

де $S_w^2(\overline{U}, \bar{I}_t)$ - зважена міра подібності між користувачем та елементом, $i_n^{(t)}$- значення *n*-го атрибута *t*-го елемента, U_n- значення *n*-го атрибута для конкретного користувача, r_n- коефіцієнт відповідності між *n*-иматрибутом з вектору атрибутів користувача та вектором елементів, N_p – загальна кількість атрибутів користувачів або елементів.

На відміну від звичайної Евклідової формули відстані, яка дозволяє обчислити фактичну відстань між двома векторами атрибутів, міру подібності отримано у

вигляді зваженого значення відстані, що розраховується, як сума квадратів часткових відстаней між конкретними атрибутами користувача або елемента, помноженого на певний ваговий коефіцієнт r_n. Тобто знаходиться подібність, що базується на відстані та значення якої залежить від більш ніж одного параметра.

У цьому випадку, крім часткової відстані між атрибутами користувача та елемента, використано параметр -лексикографічна відповідність між цими атрибутами r_n. Оскільки реалізується контекстна модель даних, що буде використовувати для створення рекомендаційвикористовуємо міру лексикографічної відповідності r_n, як параметр формули (1)

$$\forall r_n = \sum_j \left(\frac{1}{\min\{L\}} \middle| i_j^{(n)} = u_j^{(n)} \right), \quad (2)$$

де r_n - значення лексикографічної відповідності між *n*-иматрибутом конкретного користувача та елемента,$i_j^{(n)}$–*j*-ийсимвол у рядковому поданні *n*-го атрибута конкретного елемента, $u_j^{(n)}$ - *j*-ийсимвол у рядковому поданні *n*-го атрибутаконкретного користувача, $\min\{L\}$ - фактична довжина найменшого з двох рядків, що представляють *n*-ийатрибут користувача або елемента.

Для обчислення відповідності атрибутів користувача або елемента, будемо використовувати алгоритм лінійного пошуку шляхом ітерації над двома рядками, що представляють атрибут *n*-ий, і для кожної позиції виконуємо порівняння символів, розташованих на одній *j*-ійпозиції. Якщо ці символи $i_j^{(n)}$і $u_j^{(n)}$точно збігаються, то ми додаємо значення $\frac{1}{L}$ до значення відповідності r_n. Параметр L - це фактична довжина найменшого рядка, яка представляє *n*-ийатрибут користувача або елемента.

Крім того, для забезпечення більш гнучких обчислень подібності на відстані за допомогою формул (1,2) модифіковано формулу (1), використовуючи інший додатковий параметр *m*, який представляє фактичну кількість атрибутів користувача або елемента, які точно відповідають значенню актуальності $\forall r_n = 1$. Основноюпричиною використання додаткового параметра є те, що схожість конкретних користувачів та елементів, представлена значенням фактичної відстані між вектором атрибутів користувача та елемента, багато в чому залежить від кількості атрибутів, які є лексикографічно рівними. Тоді формула (1) матиме вигляд:

$$S_w(\bar{U}, \bar{I}_t) = \frac{1}{m} * \sqrt{\sum_{n=1}^{N_p} \left(\frac{1}{r_n} * \left[i_n^{(t)} - U_n \right]^2 \right)} \quad (3)$$

Велика кількість атрибутів з лексикографічною відповідністю гарантує, що Евклідова відстань між двома векторами атрибутів користувача або елемента є невеликою, що забезпечує збільшення подібності між конкретним користувачем і елементом. Однак існують випадки, коли конкретний користувач і деякий елемент не мають атрибутів, які є подібними та значення додаткового параметра $m = 0$. У цьому випадку будемо приймати значення параметра $m = 0.01$ для забезпечення великої відстані між кардинально різними користувачем та елементом, щоб уникнути випадків, коли елементи з атрибутами, що зовсім не відповідають атрибутам користувача, мають ближчу відстань до цього користувача. На рисунку 1 наведено блок-схему роботи алгоритму.

Висновок

У даній статті викладена мета розробки рекомендаційної системи, концепція системи.Виділена основна задача, наведено аналіз існуючих методів розв'язання та виявлено їх основний недолік – відсутність можливості знаходити рекомендації без профіля або попередньої історії клієнта.

Наведений алгоритм дозволяє знаходити рекомендації на основі побажань клієнта без попередньої історії дій та існуючого профілю. Перед знаходженням рекомендацій дані автомобілів, що наявні в системі й параметри, які обрав клієнт перетворюються з текстових описів на вектори слів, проводиться індексація і нормалізації і після чого отримуються вектори характеристик автомобілів та векторпобажань клієнта. Алгоритм знаходить

найбільш подібні автомобілі за допомогою визначення мір відповідності між вектором побажань клієнта та векторами характеристик автомобілів.

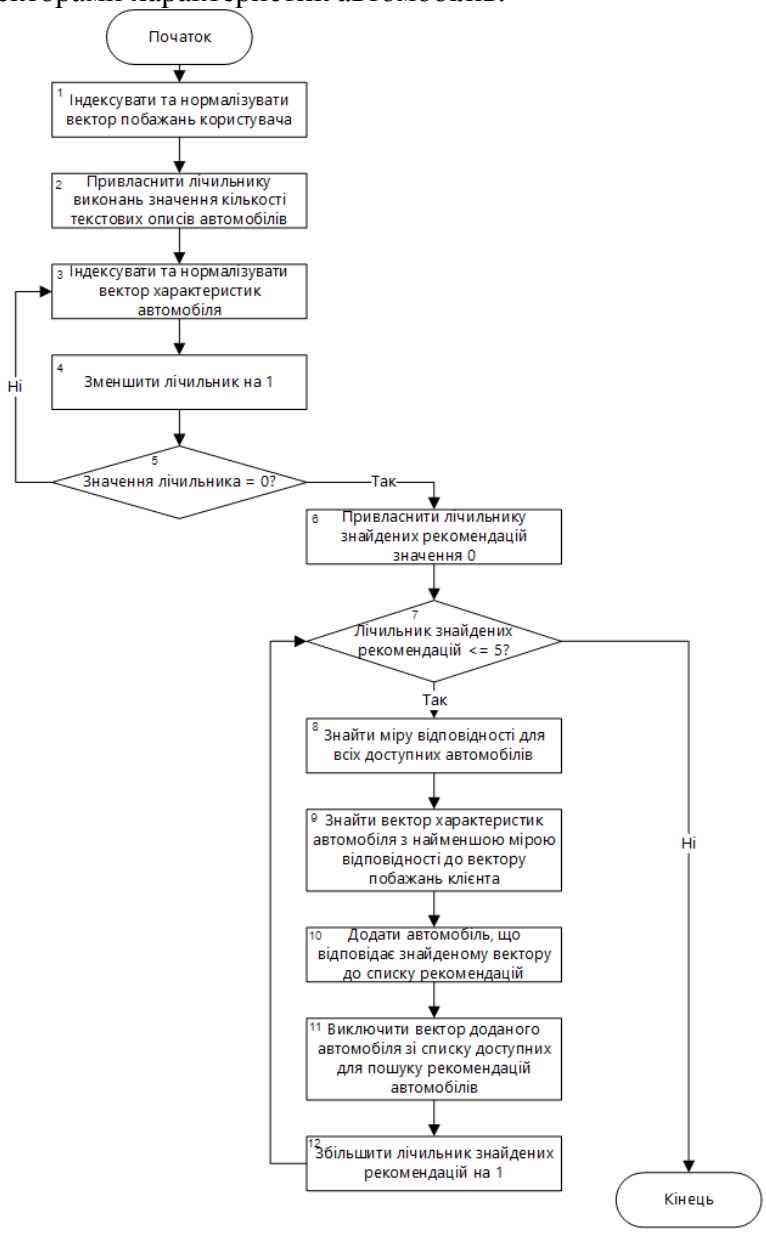


Рисунок 1. Блок-схема роботи алгоритму

Список літератури

1. Ковалевський Л. Г. Світовий автомобільний ринок: сучасний стан, особливості та перспективи розвитку / Л. Г. Ковалевський, Н. Ю. Коровайченко // Зовнішня торгівля: економіка, фінанси, право. - 2015. - № 5-6. - С. 60–67. - Режим доступу: http://nbuv.gov.ua/UJRN/uazt_2015_5-6_8.
2. Статистика автопродаж [Електронний ресурс] – Режим доступу до ресурсу: https://proautomoto.com/category/202-2020.
3. Щербань В. С., Гайдейчук Ю. А. Рекомендаційна система вибору відеофільмів; GoogleScholar. – 2016. – 4 с., Електронний ресурс: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10841/573.pdf?sequence=3
4. Щербак Д. В. Система рекомендації навчальних матеріалів / Д. В. Щербак, О. П. Сирота // Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія : Технічні науки. - 2018. - Т. 29(68), № 6(2). - С. 26-29. - Режим доступу: http://nbuv.gov.ua/UJRN/sntuts_2018_29(68)_6(2)__8.
5. MeleshkoE.V. Дослідження методів побудови рекомендаційних систем в мережі інтернет / MeleshkoE.V., S.G. Semenov, V.D. Khokh // Системи управління, навігації та зв’язку. Збірник наукових праць. – Полтава: ПНТУ, 2018. – Т. 1 (47). – С. 131-136. – doi:https://doi.org/10.26906/SUNZ.2018.1.131.

УДК 004.93

НОВІЧЕНКО Н.В.

ПОРІВНЯННЯ АЛГОРИТМІВ КЛАСИФІКАЦІЇ ЗООБРАЖЕНЬ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ВИЗНАЧЕННЯ АРХУТЕКТУРНОГО СТИЛЮ БУДІВЕЛЬ

Класифікація зображень використовується в багатьох галузях, таких як медицина, освіта, безпека, результати класифікації є основою для багатьох екологічних та соціально-економічних застосувань. Правильне вирішення задачіможе мати життєво важливе значення якщо використовується, наприклад, в медицині. Саме тому правильний вибір алгоритму, підходу та вдосконалення існуючих методів є надзвичайно важливою задачею. Інженери докладають багато зусиль у розробці вдосконалених підходів до класифікації та методів для підвищення точності класифікації. В статті наведено огляд існуючих алгоритмів розпізнавання(класифікації) зображень та їх порівняння на прикладі задачі класифікації архітектурних стилів будівель.

МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, АРХІТЕКТУРНІ СТИЛІ

Image classification is a widespread task, emerging in medicine, education, security, and many other fields. Image classification results are the basis for many environmental and socioeconomic applications. Proper problem solving can be vital if it is used, for example, in medicine. That is why the right choice of algorithm, approach, and improvement of existing methods is an extremely important task. Engineers put a lot of effort into the development of advanced approaches of classification and improvement of classification accuracy. The article presents the review of existing image classification algorithms and their comparison in solving the task of classification of architectural styles of building.

MACHINE LEARNING, NEURAL NETWORKS, IMAGE CLASSIFICATION, ARCHITECTURAL STYLES

1. Вступ

Задача класифікації полягає у побудові алгоритму що здатний класифікувати безліч об'єктів (ситуацій)деяким чином на класи.У математичній статистиці завдання класифікації називаються також завданнями дискримінантного аналізу.Існує безліч підходів до вирішення поставленної задачі, кожен з яких має свої особливості і мають різну ефективність залежно від особливостей задачі та вихідних даних.

Задача визначення архітектурного стилю будівлі полягає у присвоєнні номеру або найменування класу, що видається алгоритмом класифікації в результаті його застосування до даного конкретного зображення. Для визначення классу зображення будівлі має бути розбите на певні деталі, комбінація яких визначає архітектурний стиль. Кожен стиль відрізняється власними стійкими художніми формами, описує похідну епоху і визначає культурне обличчя епохи та країни у певний період. Наприклад, раціоналізм – напрямок в архітектурі 20 ст., намагається не тільки відобразити в архітектурі ті чи інші життєві процеси, а й активно впливати на ці процеси «винайти» для них нові форми. Характеризується цей стиль естетизацією сучасної техніки (що схоже на техніцизм), пошуки виразності простих геометричних форм, відмова від декору, інтерес до пропорціям, до кольору, до синтезу архітектури з іншими мистецтвами. Зовнішні ознаки – геометрична простота форм, асиметрія, мінімалізм у декору, використання металу, залізобетону та скла. Раціоналізм – спрощеність і стандарт. [1]

Задача класифікації зображеннь за архітектурним стилем ускладнюється тим, що велика кількість архітектурних стилів має схожість основних рис, а особливістю цієї задачі є можливість існування територіальних особливостей у кожного

архітектурного стилю. додаткові вимоги до методів класифікації зображень накладає необхідність мати можливість визначати стиль для зображень з різних ракурсів та різної якості.

2. Існуючі рішення задачі класифікації зображень будівель

Проблема класифікації архітектурних стилів будівель вже була розглянута раніше, адже з розвитком технологій комп'ютерного зору та постійно зростаючу кількість цифрової документації з'явилася необхідність у автоматизації процесу аналізу інформації. Проаналізувавши існуючі роботи можно виділити наступні підходи до розробки алгоритму класифікації:

- алгоритм, заснований на згорткових нейронних мережах (CNNs). Такий підхід можно побачити у роботі Кріс Песто, присвяченій вирішенню питання класифікації будинків Америки. Алгоритм спроможній розрізняти стилі, притаманні для архітектури американських будинків. Вибірка для навчання засновувалась на 5 поширених архітектурних стилів у США [2];
- підхід без використання машинного навчання. Наприклад, алгоритм Гаяне Шалунц, Юлл Хашимуса і Роберт Саблатинга представлений у роботі «Класифікація архітектурного стилю фасаду будівлі» використовує підхід на основі градієнтних напрямків з використанням SVM на 4 класи для класифікації архітектурного стилю з особливим фокусом на «зворотному процесуальному моделюванні» – використовуючи зображення для створення генеративної процедурної моделі для 3D-графіки [3];
- використання латентних змінних моделі - «Architectural Style Classification using Multinomial Latent Logistic Regression»[4];
- визначення стилю за вікнами будівлі - «Architectural Style Classification of Building Facade Windows»[5];
- алгоритм, заснований на різних нейронних мережах, що виділяють окремі елементи будівлі, за комбінацією яких визначається архітектурний стиль.«Класифікація зображень архітектурної спадщини методами глибокого навчання» виконана Хосе Ламасом, Педро М. Леронесом та іншими, основною метою поставили оцінку застосовності різних методів класифікації зображень архітектурної спадщини. В результаті даної роботи були отримані різні навчені нейронні сітки, спроможні розрізняти різні частини будівель (колони, горгульї, дзвіниці, тощо) [6].

Тобто, можна виділитидва кардинально різні підходи, що можуть бути використані до розробки алгоритму класифікації будівель:алгоритми, засновані на нейронних мережах та алгоритми без використання машинного навчання.

3. Порівняння алгоритмів класифікації зображень

Розглянутіалгоритмиу роботах [3]-[6]при аналізу архітектурного стилю будівлі не можуть враховать архітектурні елементи будівлі та форму будівлі. Для вирішення цієї проблеми найбільш ефектвним є підхід з використаннямdeep learning models, що на сьогодняшній день дають найточніші результати при роботі з зображеннями. До того, як deep learning models набули популярності, для класифікації зображень використовували, наприклад, наступні методи: виділити певні «особливості» на зображеннях – певні об'єкти і саме на цих об'єктах виконувалася класифікація за допомогою таких алгоритмів, як SVM. Деякі алгоритми працювали на основі значень рівнів пікселів, як виділених об'єктах. Deep learning models можна розглядати як автоматичні екстрактори об'єктів із зображення за допомогою своєї структури. Порівняно з SVM алгоритмом, deep learning models використовує інформацію про сусідні пікселі, що дозволяє більш ефективно спрощувати зображення, чого не дозволяє зробити SVM.

Розглянемо найбільш успішні deeplearningmodels – глибинні нейронні мережі (DNN), та згорткові нейронні мережі (CNN).

DNN використовують повністю звязану архітектуру, як на рисунку 1, де кожен нейрон в одному шарі з'єднується з усіма нейронами в наступному шарі.

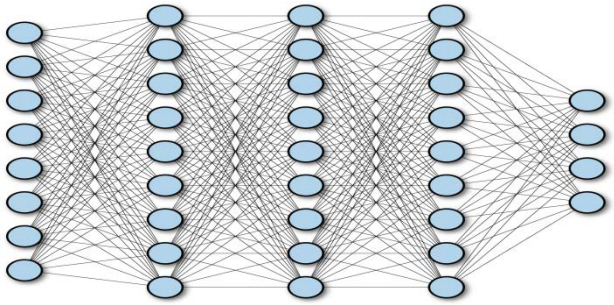


Рис.1 – багатошарова повністю звязана архітектура

Така архітектура неефективна, коли йдеться про обробку даних зображень, тому що для кольорового зображення із сотнями пікселів традиційна нейронна мережа генерує мільйони параметрів, що може призвести до перенавчання дуже швидко, модель буде виконувати дуже велику кількість обчислюнь, можливо, буде важко інтерпретувати результати, налагоджувати та налаштовувати модель для покращення її продуктивності[7].

Згорткові нейронні мережі (CNN) – це особлива архітектура штучних нейронних мереж, що заснована на механізмах зорової кори, саме тому ця архітектура є найбільш популярною для вирішення задачі класифікації зображень. Алгоритм CNN включає в себе 2 основних процеса: згортка та обєднання. Зображення передається через ряд згорткових, нелінійних об'єднуючих шарів (шарів спрощення) і повністю пов'язаних шарів, а потім генерує вихід. На шар згортки завантажується зображення (матриця з пікселів розміру n × n). Далі вибирається матрицю, що є частиною загальної матриці, яку називають фільтром (або нейроном, або ядром) за наступним принципом:

$$x_{ij}^{(l)} = \sigma(\mathrm{b} + \sum_{r=0}^{n}\sum_{c=0}^{n} W_{r,c}x_{i+r,j+c}^{(l-1)})$$

Ця операція повторюється для кожного фільтру, що отримується переміщенням вбік (або вниз) 1 одиницю, виконуючи подібну операцію. Після проходження фільтра по всіх позиціях отримують матрицю, але меншу, ніж вхідну матрицю.

Нейронна мережа складається з декількох згорткових мереж, поєднаних з обєднуючими шарами. Обєднуючі шари завжди йдуть після згорткових та мають функцію активації, що «виділяє» кордони на зображенні і робить матрицю більш «контрастною».

Ця операція аналогічна то процесу виділення людиною кордонів та меж предметів на зображенні. Обєднуючий шар дозволяє коригувати розміри матриці пікселів – зменшувати розмірність стовбців та рядків. Після прогону матриці на вище описаних шарах вона готова до аналізу нейронною мережою, тому що зображення достатньо зменшене в розмірах. Матриця потрапляє на «fully connected layer», де відбувається проекція матриці на вектор вхідного шару [8][9].

Саме така структура дозволяє працювати з зображеннями різних розмірів та якості.

4. Практичне застосування згорткових нейронних мереж (CNN) для класифікації зображень

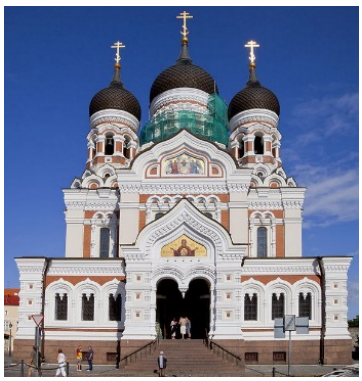
Для вирішення задачі класифікації було обрано архітектуру згорткових нейронних мереж – ResNet. CNN «витягають» низько-, середньо- та високорівневі ознаки зображення наскрізним багатошаровим способом та при збільшення кількості шарів можна досягти визначення більшої кількості «рівнів» ознак. Більшість архітектур CNN (ImageNet) дають покрашення результатів при збільшені кількості шарів, проте при досягненні певного значення результати різко погіршуються. Для уникнення цієї проблеми при необхідності збільшення шарів нейронної мережі, було використанно архітектуру ResNet, яка за рахунок глибокої «залишкової» структури навчання не дає гірші результати при збільшенні кількості шарів.

Результати класифікації зображень будівель з використанням створеного алгоритму

наведені на рисунках 2 – 4 у вигляді пар архітектурний стиль – ймовірність.

chicagoSchool : 72.9
artDeco : 11.3
artNouveau : 10.8

Рис.2 – Результати класифікації будівлі архітектурного стилю Chicago School



russianRevival : 68.5
baroque : 15.3
colonial : 11.4

Рис. 3 – Результати класифікації будівлі архітектурного стилю Russian Revival

queenAnne : 71.6
tudorRevival : 7.5
artNouveau : 4.1

Рис.4 – Результати класифікації будівлі архітектурного стилю Queen Anne

Висновки

Було розглянуто алгоритми вирішення задачі класифікації зображеннь будівлі за архітектурним стилем з використанням алгоритмів, заснований на нейронних мережах(DNN, CNN) табез використання машинного навчання (SVM), за допомогою латентних змінних моделі. Результати порівняння показали, що згорткові нейронні мережі найбільш підходять для вирішення задачі класифікації зображень. Наведено результати класифікації за допомогою згорткових нейронних мереж. Для класифікації, що вимагає аналізу данних, виділення основних «ознак» серед данних доцільно використовувати саме згорткові нейронні мережі через їх особливу архітектуру, що включає процеси згортки та обєднання. Саме така структура дозволяє працювати з зображеннями будівель з різних ракурсів, різних розмірів та якості. Для виключення погіршення результатів при збільшені глибини нейронної мережі доцільно використовувати архітектуру CNN – ResNet.

Перелік посилань

1. Земляна і. О. Найбільш значимі архітектурні стилі [електронний ресурс] / Ірина Ілександрівна Земляна // класна оцінка. – 2011. – Режим доступу до ресурсу: http://klasnaocinka.com.ua/ru/article/naibolee-znachimie-arkhitekturnie-stili.html.
2. Chris P. Classifying U.S. Houses by Architectural Style Using Convolutional Neural Networks / Pesto Chris. – Stanford: Stanford University. – 9 с.
3. Shalunts, Gayane, Yll Haxhimusa, and Robert Sablatnig. "Architectural style classification of building facade windows." International Symposium on Visual Computing. Springer Berlin Heidelberg, 2011.
4. Z. Xu, D. Tao, Y. Zhang, J. Wu, A. Tsoi, "Architectural Style Classification using Multinomial Latent Logistic Regression", European Conference on Computer Vision (ECCV2014), 2014.
5. Shalunts, Gayane, Yll Haxhimusa, and Robert Sablatnig. "Architectural style classification of building facade windows." International Symposium on Visual Computing. Springer Berlin Heidelberg, 2011.

6. Classification of Architectural Heritage Images Using Deep Learning Techniques Jose Llamas 1,* ID , Pedro M. Lerones 1 , Roberto Medina 1, Eduardo Zalama 2 and Jaime Gómez-García-Bermejo 2 ID https://pdfs.semanticscholar.org/1a1c/4e75c74b715fcc0903a044c4f7aa3d3bbf1c.pdf.
7. Shahar G.Neural Networks for Image Recognition: Methods, Best Practices, Applications [Електронний ресурс] / Shahar G // MissingLink.– Режим доступу до ресурсу: https://missinglink.ai/guides/computer-vision/neural-networks-image-recognition-methods-best-practices-applications/.
8. Sorokina K. Image Classification with Convolutional Neural Networks [Електронний ресурс] / Ksenia Sorokina //. – 2017. – Режим доступу до ресурсу: https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8.
9. Chen W. Convolutional Neural Network for Image Classification / W. Chen, X. Yang.. – (Johns Hopkins University).

УДК 004.93

ХУДА А. О.
ХАЛУС О. А.

МЕТОД ПОРІВНЯННЯ АУДІОФАЙЛІВ УСИСТЕМІ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ СЛІВ

У даній статті розглядається задача порівняння двох аудіофайлів для вирішення проблеми виявлення помилок у вимові людини при вивченні іноземних слів. Порівняння аудіофайлів відбувається в декілька етапів, які розглядаються у даній статті. Ціллю являється створення моделі, що може порівняти два аудіофайли та надати результати порівняння.
КЛЮЧОВІ СЛОВА: ДИНАМІЧНА ТРАНСФОРМАЦІЯ ЧАСОВОЇ ШКАЛИ, ДИСКРЕТНЕ ПЕРЕТВОРЕННЯ ФУР'Є, ШВИДКЕ ПЕРЕТВОРЕННЯ ФУР'Є, АУДІОСИГНАЛИ

This article considers an objective of comparing two audio files to solve the problem of detecting human pronunciation errors in the process of learning foreign words. The comparison of audio files proceed in several stages, which are considered in this article. The goal is to create a model that can compare two audio files and provide comparison results.
KEYWORDS: DYNAMIC TIME WRAPING, DISCRETE FOURIER TRANSFORMATION, FAST FOURIER TRANSFORMATION, AUDIOSIGNALS

1. Вступ

Сьогодні знання іноземних мов, а тим більше англійської, є неохідним. Якщо ж мову можна вивчити за допомогою різноманітних онлайн курсів, то ситуація з вимовую є складнішою. Є декілька варіантів тренування вимови. Перший варіант – слухати та повторювати. Цей варіант не є дуже ефективним, оскільки людині складно почути свою вимову зі сторони, а тим більше оцінити її. Другий спосіб – курси іноземних мов. Цей варіант ефективний, але доволі дорогий. Третій підхід – використання спеціалізованихзастосувань для спілкування з носіями мови для її вивчення. Третій варіант є дуже ефективним, але має декілька недоліків. По перше, складно знайти людину яка дійсно зацікавлена у тому, щоб допомогти з постановкою вимови, оскільки такі застосунки, в основному, призначені для того, щоб люди просто спілкувалися на різноманітні теми і не замислювались над постановкою вимови. По друге, є люди, яким некомфортно спілкуватись з незнайомими людьми і це є значною перешкодою для вивчення мови за домогою таких застосунків.

Таким чином вирішення задачі постановки вимови за допомогою інтелектуальної системи, яка має можливість аналізувати вимову користувача є дуже актуальною в наш час.

У статті представлено алгоритм порівняння двох аудіофайлів, один із яких

зчитаний з мікрофону користувача, а другий – згенерований системою за допомогою методу Text-to-Speech. А також, представлення результатів користувачу.

2. Загальний алгоритм порівняння аудіофайлів

Порівняння аудіофайлів є складним процесом і відбувається у декілька етапів за наступним алгоритмом.

КРОК1. Завантажити аудіофайли.

КРОК 2. Обрізати два сигнали так, щоб їх тривалість була однаковою

КРОК 3. Застосувати алгоритм швидкого перетворення Фур'є.

КРОК 4. Порахувати середньоквадратичну помилку. Два ідентичних сигнали очевидно будуть мати середньоквадратичну помилку рівну нулю.

КРОК 5. Відобразити помилку у випадку, коли середньоквадратична помилка більша за заданий поріг.

3. Синхронізація аудіофайлів за допомогою алгоритмудинамічної трансформації часової шкали

Задача, яку необхідно вирішити полягає в тому, щоб порівняти два аудіофайли різної довжини. Маємо два сигнали *x*та *y,* які можуть мати однакове звучання, але мають різну тривалість. Навіть, якщо виразити два аудіосигнали за допомогою однакової функції, підсумування їх попарних відстаней є неможливим, оскільки сигнали мають різну довжину. Ця проблема називається синхронізація аудіофайлів.

Для вирішення цієї проблеми застосовується алгоритм динамічної трансформації часової шкали (DTW– DynamicTimeWraping). Даний алгоритм застосовується для вирівнювання двох послідовностей подібного змісту, але, можливо, різної довжини.

Вимірювання відстані між двома часовими рядами потрібно для того, щоб визначити їх подібність і класифікацію. Таким ефективним виміром є евклідова метрика. Для двох часових послідовностей це просто сума квадратів відстаней від кожної *n*-ої точки однієї послідовності до n-ої точки іншої. Однак, використання евклідової відстані має істотний недолік: якщо два тимчасових ряди однакові, але один з них незначно зміщений у часі (уздовж осі часу), то евклідова метрика може порахувати, що ряди відрізняються один від одного. DTW-алгоритм був введений для того, щоб подолати цей недолік і надати наочне вимірювання відстані між рядами, не звертаючи увагу як на глобальні, так і на локальні зрушення на часовій шкалі.[1]

Розглянемо два часових ряда – *Q*довжини *n*та *C*довжини *m.*

$$Q = q_1, q_2, \dots, q_i, \dots, q_n$$
$$C = c_1, c_2, \dots, c_j, \dots, c_m$$

Перший етап алгоритму заключається в наступному. Необхідно побудувати матрицю *d*порядку $n \times m$(матрицю відстаней), в якій елемент d_{ij}є відстанню $d(q_i, c_j)$між двома точками q_i та c_j. Зазвичай використовується евклідова відстань: $d(q_i, c_j) = (q_i - c_j)^2$або $d(q_i, c_j) = |q_i - c_j|$. Кожний елемент *(i, j)* матриці відповідає вирівнюванню між точками q_i і c_j.

На другому етапі будуємо матрицю трансформармації, кожний елемент якої обчислюється виходячи з наступного співвідношення:

$$D_{i\,j} = d_{i\,j} + min(D_{i-1\,j}, D_{i-1\,j-1}, D_{i\,j-1})$$

Після заповнення матриці трансформаціїї, переходимо до заключного етапу, який полягає в тому, щоб побудувати деякий оптимальний шлях трансформації і DWTвідстань (вартість шляху).

Шлях трансформації *W*– це набір суміжних елементів матриці, який встановлює відповідність між *Q*та *C*. Він являє собою шлях, який мінімізує загальну відстань між *Q*та *C*. *k*-ий елемент шляху *W*визначається як:

$$w_k = (i, j)_k, \; d(w_k) = d(q_i, c_j) = (q_i - c_j)^2$$

Таким чином:

$$W = w_1, w_2, \dots, w_k, \dots, w_K; \quad max(m, n) \leq K < m + n,$$

де *K* – довжина шляху.

Шлях трансформації повинен відповідати таким обмежуючим умовам:

- граничні умови: початок шляху $w_1 = (1, 1)$, його кінець $w_K = (m, n)$. Це обмеження гарантує, що шлях

трансформації містить всі точки обох часових рядів.

- Безперервність (умова на довжину кроку): будь-які два суміжних елемента шляхуW, $w_k = (w_i, w_j)$і $w_{k+1} = (w_{i+1}, w_{j+1})$, задовільняють наступні нерівності: $w_i - w_{i+1} \leq 1, w_j - w_{j-1} \leq 1$. Це обмеження гарантує, що шлях трансформації пересувається на один крок за один раз.Тобто обидва індекси iта j можуть збільшитися лише на 1 на кожному кроці шляху.
- Монотонність: будь-які два суміжних елемента шляхуW, $w_k = (w_i, w_j)$і $w_{k-1} = (w_{i-1}, w_{j-1})$,задовільняють наступні нерівності:$w_i - w_{i-1} \geq 0, w_j - w_{j-1} \geq 0$. Це обмеження гарантує, що шлях трансформації не повертатиметься назад до пройденої точки.Тобто обидва індексиiтаj або залишаються незмінними, або збільшуються, але ніколи не зменшуються.

Хоча існує велика кількість шляхів трансформації, які задовільняютьусім вищевказаним умовам, проте необхідний тільки той шлях, який мінімізує DTW відстань (вартість шляху).

DTW відстань (вартість шляху) між двома послідовностями розраховується на основі оптимального шляху трансформації за допомогою формули:

$$DTW(Q,C) = min\left\{\frac{\sum_{k=1}^{K} d(w_k)}{K}\right\}$$

K в знаменнику використовується для врахування того, що шляхи трансформації можуть бути різної довжини.

Просторова і тимчасова складність алгоритму – квадратична*O (nm)*, так як DTW алгоритм повинен вивчити кожну клітинку матриці трансформації.

4. Перетворення Фур'є для обробки аудіосигналів

Як відомо, звуковий сигнал у комп'ютері може бути представлений у вигляді деякого набору відліків його амплітуд, що створюються через певні проміжки часу (періоди дискретизації) та представлені деякою кількістю двійкових розрядів (розрядність виборки). Такепредставлення зручне для зберігання звукового сигналу та йогозворотнього перетворення в безперервний сигнал. Однак, деякі операції обробки звукового сигналу в такому представленні не завжди є зручними. Це пов'язано з тим, що реальний звуковий сигнал складається з частот з визначеною амплітудою та фазою. Таким чином, застосування таких операцій обробки звукового сигналу як "фільтр нижніх частот" або "фільтр верхніх частот" вимагає перетворень представлень звукового сигналу у вигляді відлікув його частотному спектрі. Після цього перетворення звукового сигналу буде представлено у вигляді коефіцієнтів, що відповідають амплітудам і фазам частот, якіі складають цей сигнал. Тепер, наприклад, операція "фільтр нижніх частот", що вирізає із сигналів усі частоти, які вище заданої, може просто онулити, відповідні частотам, коефіцієнти,які необхідно вирізати. Насправді, область застосування такого перетворення значно ширша: обробка растрових зображень, телекомунікації, дослідження та вимірювання сигналів, радіолокація та ін. Прикладом застосування перетворення може бути передача даних у цифровій формі по аналоговим лініями телефонної мережі (модем). Для передачі даних у цифровій формі вони починають перетворюватися в певний набір частот і передаються по лінії передачі, а потім, на приймаючій стороні, виконується фундаментальне перетворення та відновлюються вихідні дані. Далі розглянуті фундаментальні поняття, які необхідні для розуміння роботи перетворення Фур'є. [3]

Рядом Фур'є називається нескінченна математична послідовність, яка складається з коефіцієнтів при функціях синуса та косинуса вигляду:

$$\frac{a_0}{2} + \sum_{m=1}^{\infty}\left(a_m \cos\frac{m\pi x}{l} + b_m \sin\frac{m\pi x}{l}\right)$$

Доведено, що якщо деяка періодична функція з періодом $2l$ на інтервалі $[-l, l]$задовільняє умови першого роду (умови Діріхле), тобто безперервна і має скінченну кількість екстремумів і точок розриву I роду, то ця функція може представлятися у вигляді суми ряду Фур'є.

Для визначення коефіцієнтів ряду Фур'є використовуються наступні формули:

$$a_m = \frac{1}{l}\int_{-l}^{l} f(x)\cos\frac{m\pi x}{l}\,dx,$$

$$a_m = \frac{1}{l}\int_{-l}^{l} f(x)\sin\frac{m\pi x}{l}\,dx$$

Якщо функція, що розкладається, парна, тобто $f(-x) = f(x)$[2], то ряд Фур'є складається лише з косинусів. Це означає, що усі коефіцієнти при синусах рівні нулю. Якщо ж функція є непарною, тобто $f(-x) = -f(x)$ [2], тоді ряд Фур'є складається лише з синусів. Це означає, що усі коефіцієнти при косинусах рівні нулю. В загальному випадку, коефіцієнти при синусах і косинусах не рівні нулю. Таким чином, будь яку періодичну функцію, яка задовільняє умови першого роду, можна розкласти в ряд Фур'є, представивши дану функцію у вигляді суми косинусів і синусів.

Уявімо, що період повторення досліджуваної функції – нескінченність. Тобто функція не є періодичною. В такому випадку, має місце інтегральна формула Фур'є, яка отримується шляхом граничного переходу з ряду Фур'є періодичної функції з періодом 2*l:*

$$f(x) = \frac{1}{\pi}\int_{0}^{+\infty} dz \int_{-\infty}^{+\infty} f(u)\cos z(u-x)du$$

З цією формулою пов'язані інтегральні перетворення Фур'є:

$$F(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} e^{izx} f(x)dx$$

$$f(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty} e^{-izx} F(z)dz$$

З цих формул можна винести синус-перетворення Фур'є для непарних функцій:

$$f_s(z) = \sqrt{\frac{2}{\pi}}\int_{0}^{+\infty} f(x)\sin zx\,dx$$

$$f(x) = \sqrt{\frac{2}{\pi}}\int_{0}^{+\infty} f_s(z)\sin zx\,dx$$

Тепер винесемо косинус-перетворення Фур'є для парних функцій:

$$f_c(z) = \sqrt{\frac{2}{\pi}}\int_{0}^{+\infty} f(x)\cos zx\,dx$$

$$f(x) = \sqrt{\frac{2}{\pi}}\int_{0}^{+\infty} f_c(z)\sin zx\,dx$$

Таким чином безперервне перетворення Фур'є дозволяє представити неперіодичну функцію у вигляді інтеграла функції, яка представляє у кожній своїй точці коефіцієнт ряду Фур'є для неперіодичної функції.

В дискретному перетворені Фур'є (DFT – DiscreteFourierTransform) досліджувана функція є періодичною, має кінцевий період повторення та є дискретною. Для визначення амплітуд та фаз частотних складових сигналу в DFT використовується співвідношення з базисними функціями синуса та косинуса. Спектр частот в DFT визначається з амплітуд синусів і косинусів з частотами повторення у досліджуваній виборці від 0 до *N*/2, де *N*– кількість елементів вибірки. Перетворення Фур'є розкладає дискретизований сигнал з *N*вибірок на *N*/2+1 синусних та *N*/2+1 косинусних складових. Це призводить до наступних формул DFT:

$$ReX[k] = \sum_{i=0}^{N-1} x[i]\cos\left(\frac{2\pi ki}{N}\right)$$

$$ImX[k] = -\sum_{i=0}^{N-1} x[i]\sin\left(\frac{2\pi ki}{N}\right)$$

Ці формули описують пряме перетворення Фур'є. $ReX[k]$ – масив, у якому міститься значення косинусоідальних складових, відповідно, $ImX[k]$ – масив синусоідальних складових.

По наведеним вище формулам відбувається розкладання досліджуваного сигналу в його спектр.

Після розкладу сигналу на синусоідальні та косинусоідальні складові можна отримати значенння амплітуд та фаз частотних складових сигналу. Для переводу пари відповідних коефіцієнтів при синусі і косинусі в амплітуду і фазу частотної складової використаємо формули:

$$MagX[k] = (ReX[k]^2 + ImX[k]^2)^{1/2}$$

$$PhaseX[k] = arctan\left(\frac{ImX[k]}{ReX[k]}\right)$$

Отримані значення амплітуди і фази називають полярним представленням. Значення коефіцієнтів при синусах і косинусах характеризують прямокутне представлення. Значення фази сигналу несе у собі інформацію про форму і зміни сигналу.

Швидке перетворення Фур'є (FFT – FastFourierTransform)– це алгоритм, який часто використовується для цифрової обробки сигналів при перетворенні дискретних даних з часового у частотний діапазон. Швидке перетворення Фур'є є однією їз реалізації перетворення Фур'є. Алгоритм FFTобчислює дискретне перетворення Фур'є (DFT – DiscreteFourierTransform) або зворотнє дискретне перетворення (IDFT – InverseDiscreteFourierTransform). Оскільки обчислення DFT є надто повільним, часто використовуютьFFT, якийшвидко обчислює такі перетворення, розкладаючи матрицю DFTна добуток розсіяних (переважно нульових) факторів. Таким чином алгоритму FFTвдається зменшити складність DFTз $O(n^2)$до $O(n\log n)$, де n– розмір даних. Різниця в швидкості може бути величезною, особливо для великих наборів даних, де N може складати тисячі або мільйони.

5. Середньоквадратична помилка

Після виконання перетворення Фур'є отримано набір характерних частот двох аудіосигналів. Для визначення відносної різниці між ними використовується метод визначення середньоквадратичної помилки. Середньоквадратична помилка визначається наступною формулою:

$$E = \frac{1}{n}\sum_{i=1}^{n}(MagA_i - MagB_i)^2,$$

де $MagA_i$ – це амплітуда частоти *i* в сигналі *A*, а $MagB_i$ – це амплітуда частоти *i* в сигналі *B*.

При визначені середньоквадратичної помилки, різницею в фазових складових нехтується, оскільки вони не несуть корисної інформації.

Висновок

У даній статті було розглянуто метод порівняння двох аудіофайлів, що може бути використано з метою визначення правильності вимови іншомовних слів. Даний метод відбувається у декілька етапів, найважливіші з яких були розглянуті у статті. Було розглянуто алгоритм динамічної трансформації часової шкали для синхронізаціїї аудіофайлів, дискретне перетворення Фур'є та швидке перетворення Фур'є для представлення аудіосигналів у дискретному вигляді. Для відображення результату порівняння двох аудіофайлів було обрано середньоквадратичну помилку.

Список літератури

1. Müller M. Fundamentals of Music Processing — Audio, Analysis, Algorithms, Applications. / Meinard Müller., 2015. – 480 с.
2. В. П. Денисюк, В. К. Репета. Вища математика – Київ: НАУ, 2013. – 145 с.
3. Применение преобразования Фурье в цифровой обработке звука [Електронний ресурс] – Режим доступу до ресурсу: http://shackmaster.narod.ru/fourier.htm.

УДК004.93

БУРЛАЧЕНКО Є.О.
ХАЛУС О.А.

ПРАКТИЧНЕ ЗАСТОСУВАННЯ CLOUDNATURALLANGUAGE ДЛЯ NLP

У даній статі розглянуто застосування функціоналу сервісу CloudNaturalLanguageвід компанії Google. Демонструєтьсяпростота використання та детальний об’єм наданої ним інформації. Наведено приклади використання різних методів та можливий аналіз на прикладі японської мови.

CLOUD NATURAL LANGUAGE, GOOGLE CLOUD PLATFORM, NLP, АНАЛІЗ ТЕКСТУ

The subject of this article is an application of Google’s Cloud Natural Language service. It demonstrates how simple it is to use and how much useful information it provides. Provided examples of different method usage based on Japanese language.

CLOUD NATURAL LANGUAGE, GOOGLE CLOUD PLATFORM, NLP, TEXT ANALYSIS

1.Вступ

NLP – Natural language processing – напрямок штучного інтелекту та математичної лінгвістики, що вивчає вирішення проблем компьютерного аналізу та синтезу природніх мов.

Широко використовується для онлайн ботів-консультантів, допомагаючи їм створити враження розмови з живою людиною, при автоматизованому аналізі коментарів, листів та інших форм відгуку для виявлення категорій проблем та їх важливості, у застосунках для вивчення мов для автоматизації перевірки складних завдань, застосунках для перекладу тощо.

Cloud Natural Language – сервіс у складі Google Cloud Platform, що надає методи для роботи з NLP, такі як переклад тексту, сутнісний аналіз, синтаксичний, семантичний аналіз, відокремлення токенів тощо. Також надає можливість під’єднати власну нейронну мережу для обробки чи генерації тексту.

Вирішення задач NLP потребує багато часу та ресурсів як апаратних, так і програмних, тому багатьом компаніям буде вигідніше та зручніше використати Cloud Natural Language як основу для своїх власних розробок. В наступних розділах буде розглянуто різні методи, що надає CloudNaturalLanguage, з прикладами використання.

2.Постановка задачі

На вхід кожного з видів аналізу подається текст, що потрібно проаналізувати, закодований за допомогою Protobuf, згідно інтерфейсу, що описано в документації Cloud Natural Langugeдля потрібного виду аналізу. Також у цьому інтерфейсі можливо вказати мову тексту, якщо цього не зробити вона буде визначена автоматично.На виході кожного з аналізів отримується структура, у якій знаходиться відповідна до аналізу інформація про текст, також у кодуванні Protobuf. Наприклад, на виході з класифікації тексту отримується масив структур, у кожній із яких міститься клас тексту та впевненість, з якою класифікатор відносить текст до цієї категорії.

3. Сутнісний аналіз

Сутнісний аналіз тексту дозволяє відокремити з тексту усі іменники (імена, власні та загальні назви тощо), та для кожного з них виявити тип (особа, організація, локація, продукт і т.п.), помітність (наскільки важливим є дана сутність у тексті), а також наскільки позитивним чи негативним є ставлення до сутності у тексті.

Розглянемо роботу сутнісного аналізу на прикладі наступного речення -
天才小島秀夫先生は横浜へ行きました。

(ГенійКодзіма Хідеопоїхав у Йокогаму)

```
[
  {
    "entities":[
      {
        "mentions":[
          {
            "text":{
              "content":"小島秀夫",
              "beginOffset":-1
            },
            "type":"PROPER",
            "sentiment":null
          },
          { ⊞ }
        ],
        "metadata":{
          "wikipedia_url":"https://en.wikipedia.org/wiki/Hideo_Kojima",
          "mid":"/m/01xn37"
        },
        "name":"小島秀夫",
        "type":"PERSON",
        "salience":0.6769457459449768,
        "sentiment":null
      },
      {
        "mentions":[ ⊞ ],
        "metadata":{
          "mid":"/m/0kstw",
          "wikipedia_url":"https://en.wikipedia.org/wiki/Yokohama"
        },
        "name":"横浜",
        "type":"LOCATION",
        "salience":0.3230542540550232,
        "sentiment":null
      }
    ],
    "language":"ja"
  },
  null,
  null
]
```


Рис. 1 – Результат сутнісного аналізу у форматі JSON

Як можна побачити на рисунку, сутнісний аналіз успішно виявив сутності “Кодзіма Хідео” та “Йокогама”, визначивши їх тип та навіть метадані (сторінка у Вікіпедіїта ідентифікатор у Freebase) У реальному світі даний аналіз застосовуєтьсяяк самостійно, так і у вигляді одного з етапів більш складного аналізу. Для сортування листів на електронній пошті, звернень до технічної підтримки, аналіз резюме, кластеризація відгуків на веб-сайті для виявлення найбільш популярних, проблемних продуктів і т.п.

4. Синтаксичний аналіз

Синтаксичний аналіз дозволяє розділити текст на речення та токени – окремі синтаксичні одиниці (слова, частки, розділові знаки тощо), для кожного з токенів знайти властивості , такі як частина речення, дерево залежностей, лемма, час (доконаний, прогресивний, недоконаний), відмінок, стать, настрій, кількість (однина, множина), особа (перша, друга), пасивна чи активна форма, а також інші, в залежності від наявності тих чи інших властивостей у мові. Розглянемо роботу синтаксичного аналізу на прикладі наступного речення -

東京スカイツリーは2012年2月29日に建てました。(Tokyo Sky Tree було побудовано 29 лютого 2012 року).

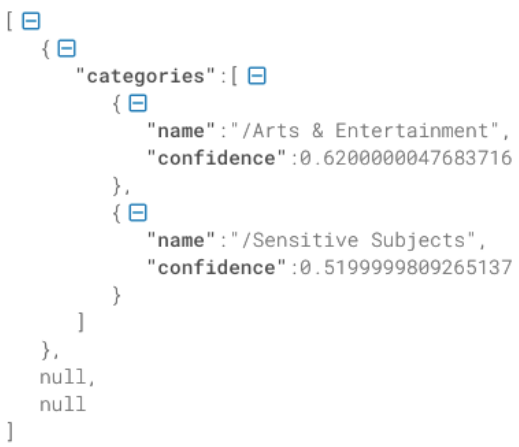
```
[
  {
    "sentences":[
      {
        "text":{
          "content":"東京スカイツリーは2012年2月29日に建てました",
          "beginOffset":-1
        },
        "sentiment":null
      }
    ],
    "tokens":[
      {
        "text":{
          "content":"東京",
          "beginOffset":-1
        },
        "partOfSpeech":{
          "tag":"NOUN",
          "aspect":"ASPECT_UNKNOWN",
          "case":"CASE_UNKNOWN",
          "form":"FORM_UNKNOWN",
          "gender":"GENDER_UNKNOWN",
          "mood":"MOOD_UNKNOWN",
          "number":"NUMBER_UNKNOWN",
          "person":"PERSON_UNKNOWN",
          "proper":"PROPER",
          "reciprocity":"RECIPROCITY_UNKNOWN",
          "tense":"TENSE_UNKNOWN",
          "voice":"VOICE_UNKNOWN"
        },
        "dependencyEdge":{
          "headTokenIndex":2,
          "label":"NN"
        },
        "lemma":"東京"
      },
      {
        "text":{
          "content":"スカイ",
          "beginOffset":-1
        },
        "partOfSpeech":{
          "tag":"NOUN",
          "aspect":"ASPECT_UNKNOWN",
          "case":"CASE_UNKNOWN",
          "form":"FORM_UNKNOWN",
          "gender":"GENDER_UNKNOWN",
          "mood":"MOOD_UNKNOWN",
          "number":"NUMBER_UNKNOWN",
          "person":"PERSON_UNKNOWN",
          "proper":"PROPER",
```


Рис. 2 - Результат синтаксичного аналізу у форматі JSON

На рисунку можна побачити що речення було успішно розбито на токени та для кожного з них було наведено додаткову інформацію по кожній категорії. Даний аналіз має аналогічне до сутнісного застосування, але на відміну від нього більш спрямоване на розуміння суті речення – онлайн боти можуть використовувати даний аналіз для виявлення команд та їх окремих параметрів без необхідності для користувача використовувати строго задану форму речення, створюючи ілюзію розмови зі справжньою людиною та полегшуючи взаємодію з ботом.

5. Класифікація тексту

Класифікація тексту дозволяє віднести його до певної категорії (бізнес, наукова стаття, роман, вірш тощо). Розглянемо результат класфікації на прикладі відривку з“Різдв’яної пісні”Чарлза Диккенса.



```
[
  {
    "categories":[
      {
        "name":"/Arts & Entertainment",
        "confidence":0.6200000047683716
      },
      {
        "name":"/Sensitive Subjects",
        "confidence":0.5199999809265137
      }
    ]
  },
  null,
  null
]
```


Рис.3 - Результат класифікації у форматі JSON

Як можна побачити, класифікатор відніс відривок до категорії “мистецтво”, але з доволі низькою впевненістю. Це пов'язане з необхідність надавати якомога більший об'єм тексту для найбільш точної класифікації.

Класифікацію тексту використовують для управління вмістом, контекстуального пошуку, аналізу відгуків та схожих сферах.

Висновки

Застосування NLPвирішує багато проблем, пов’язаних з текстом та допомагає, може підвищити прибуток компаній за рахунок автоматизації пошуку та генерації тексту. Використання готових рішень як CloudNaturalLanguageдопомагає зменшити витрати на розробку та підтримку ПЗ, що робить цей сервіс особливо привабливим для малих компаній чи стартапів.

Джерела

1.Cloud Natural Language Api Documentation [Електронний ресурс] // GoogleInc.. – 2020. – Режим доступу до ресурсу: https://cloud.google.com/natural-language.

УДК004.021

БОГДАНЕНКО М. О.
ПАВЛОВ О.А.
ЖДАНОВА О.Г.

СИСТЕМА ПІДТРИМКИ ДОСЛІДЖЕНЬ ОПТИМІЗАЦІЙНИХ ЗАДАЧ В УМОВАХ НЕВИЗНАЧЕНОСТІ

Робота присвячена системі підтримки досліджень оптимізаційних задач в умовах невизначеності. Розглянуто області застосування задач лінійного програмування в умовах невизначеності, приведено математичну модель цієї задачі та метод знаходження компромісних розв’язків, розглянуто структуру системи підтримки досліджень оптимізаційних задач.

СИСТЕМА ПІДТРИМКИ ДОСЛІДЖЕНЬ, ЗАДАЧА ЛІНІЙНОГО ПРОГРАМУВАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ, КОМПРОМІСНИЙ РОЗВ’ЯЗОК

The work is devoted to thes ystem of support of researches of optimization problems in the conditions of uncertainty. The are asofapplication of linear programming problems in condition sofuncertainty are considered, the mathematical model of this problem and the method of finding compromise solutions are given, thes tructure of the support system for research of optimization problems is considered.

RESEARCH SUPPORT SYSTEM, LINEAR PROGRAMMING PROBLEM UNDER UNCERTAINTY, COMPROMISE SOLUTION

1. Вступ

Прийняттярішень в умовах невизначеності - невід'ємна частина нашого повсякденного життя. Деякі приклади таких задач: вибір маршруту і транспортного засобу для перевезення певної кількості вантажу, визначення порядку виконання певної множини робіт, планування групою виконавців виконання деякого проекту. Усі ці задачі об'єднує наступне: у будь-який момент часу виконання цих робіт може виникнути випадковий фактор, який непередбачуваним чином вплине на результат. Наприклад, по дорозі може виникнути затор, при виконанні роботи може вийти з ладу обладнання, а при виконанні проекту - захворіти один ізвиконавців.

При вирішенні такого роду задач зазвичай прийняття рішення покладається на керівника який прогнозує можливі події, їх наслідки, та, використовуючи власний досвід, передбачає результат.

Чому прийняття рішень важливе? У першу чергу, від прийнятого рішення залежить кінцевий результат, а отже, його позитивні та негативні наслідки. Невірне рішення призводить до того, що негативні наслідки переважають над позитивними, і у підсумку витрати більші за здобутки. Такі абстрактні поняття як «здобутки» та «втрати» використано не даремно, адже описується деяка абстрактна задача. Залежно від контексту, це можуть бути: час, гроші, ресурси, енергія та будь-яка інша форма, яку можуть прийняти «здобутки» і «втрати». Ці самі витрати, як було сказано, напряму залежать від прийняття рішення, а рішення, у свою чергу, приймаються відповідальною особою. Нажаль, гарантувати якість рішення, яке було обґрунтоване суб'єктивними оцінками керівника неможливо, єдиний гарантований варіант – це прийняття рішень методами теорії рішень, адаптованими під конкретну цільову область.

Отже, прийняття рішень на основі власних оцінок – не варіант. Це цілком логічний висновок, проте через малу обізнаність людей у теорії рішень, та незручність виконання моделювання, прийняття рішень на основі власних оцінок є найрозповсюдженішим у бізнесі.

Ця робота присвячена розгляду системи підтримки дослідження оптимізаційних задач в умовах невизначеності, яка дозволить виконувати дослідження впливу зміни параметрів задач в умовах невизначеності на шуканийрозв'язок. В першій версії системи поки передбачається дослідження задачі лінійного програмування в умовах невизначеності. Метою створення системи єдосліджень алгоритмів розв'язання задачі та впливу вхідних даних (умовно-постійних параметрів задачі) на результати вирішення оптимізаційної задачі в умовах невизначеності.

2. Опис системи підтримки дослідження оптимізаційних задач в умовах невизначеності

Система являє собою комплекс программ та бібліотек, призначених для вирішення оптимізаційних задач в умовах невизначеності.

У основі поточної версії лежить використання бібліотек GoogleOR-Tools, які використовують для вирішення різного роду оптимізаційних задач, зокрема, методрозв'язання задач лінійного програмування.

На рис. 1 показано схему структури системи. Як можна побачити із рисунка, система модна поділити на певні модулі: модуль імпорту даних у вирішувач, модульпланування експериментів, модуль контролю розв'язання (він же вирішувач), модуль розв'язання індивідуальних задач, модуль аналізу, класифікатор задач, збереження результатів.

Модуль імпорту перетворює вхідні дані у форму, у якій вирішувач зможе їх обробити.

Модуль планування експериментів здійснює розподіл потоків даних та генерацію даних за встановленими користувачем параметрами.

Модуль контролю розв'язання здійснює контроль над процесом розв'язання задачі, збирає дані із розпаралелених процесів обчислення. Залежно від вхідних даних він націлений на розв'язання задачі із згенерованими даними, чи розв'язання індивідуальної задачі лінійного програмування в умовах невизначеності.

Модуль аналізу здійснює аналіз результатів, які було отримано в кінці вирішення.

Модуль класифікатор задач здійснює класифікацію задач, які було вирішено, є частиною аналізатора.
Модуль збереження результатів виконує збереження отриманих результатів розв'язання.

Рис. 1 Схема структури системи

3. Задачалінійного програмування в умовах невизначеності

Модель (детермінованої) задачі лінійного програмування:

$$c^T x,$$
$$Ax = b,$$
$$x \geq 0,$$

де x – вектор розв'язку, A – матриця коефіцієнтів обмежень, b–вектор обмежень, c–вектор витрат чи прибутків.

Як відомо, лінійне програмування широко застосовується для моделювання виробництва із продажами. Отже, лінійне програмування в умовах невизначеності – це логічне продовження лінійного програмування із врахуванням варіативності майбутнього.

Як було сказано, задача в умовах невизначеності містить невизначені значення. У випадку, коли не можна визначитиодин вектор коефіцієнтів цільової функції, складається декілька можливих, що описують різні варіанти перебігу подій.У житті це може виглядати так: існує певне виробництво, маються ресурси, мається попит на товар, що виробляється. У класичному лінійному програмуванні розглядається один найбільш вірогідний варіант ринкових цін на товар. У задачі ж лінійного програмуваннів умовах невизначеності встановлюються декілька варіантів майбутнього.

Якщо прийняти, що у нас існує Rваріантів можливого майбутнього, то задача приймає наступний вигляд:

$$f = \sum_{j=1}^{n} c_j x_j \to \min(\max) \quad (1)$$

$$c = \begin{cases} c^1 \\ \text{або} \\ c^2 \\ \text{або} \\ \dots \\ c^R \end{cases} \quad (2)$$

$$Ax = b, \quad (3)$$
$$x \geq 0. \quad (4)$$

Без втрати загальності домовимось, що напрямком оптимізації є мінімізація цільової функції. В ідеальному випадку існує абсолютний розв'язок, який є оптимальним за будь-яким з Rкритеріїв

$$f^r = \sum_{j=1}^{n} c_j^r x_j, \; r = 1,\dots,R,$$

але на практиці це малоймовірно.

Надалі, будемо називати *частковою* задачу

$$f^r = \sum_{j=1}^{n} c_j^r x_j \to \min,$$
$$Ax = b,$$
$$x \geq 0.$$

Якщо прийняти рішення реалізувати оптимальний розв'язок однієї з часткових задач, то скоріше за всена момент реалізації цей розв'язок не буде найкращим.

Для задач оптимізації в умовах невизначеності в роботах [1,2] для певного класу задач в умовах невизначеності запропонованатеорія, що базується на знаходженні «компромісних розв'язків», наводиться ряд критеріїв, за якими можуть оцінюватись компромісні розв'язки, та запропонованіметоди їх розв'язання, В роботах[3, 4] продемонстроване застосування розробленої теорії на прикладі

транспортної задачі лінійного програмування.

Згідно [1-4] введемо наступні позначення для ЗЛП в умовах невизначеності:

$$f_{opt}^{r} = \sum_{i=1}^{n} c_i^r x_i^r,$$

де f_{opt}^{r} – оптимальне значення r-ї часткової задачі, для якої x^r–оптимальний розв'язок.

Нехай x–деякийдопустимий розв'язок (розв'язок, який задовольняє обмеження (3)-(4)). Введемо для нього наступні величини

$$\Delta_r = \sum_{i=1}^{n} c_i^r x_r - f_{opt}^{r},$$

$$y_r = \max\{0; \Delta_r - l_r\}, r = 1, \dots, R.$$

3.1. Критерії якості компромісних розв'язків

В постановці задачі лінійного програмування в умовах невизначеності, що аналізуєтьсяв цій роботі, передбачається, що експертами встановлено набір величин $l_i > 0, i = \overline{1,R}$, які визначають верхні значення поступок для кожного з часткових критеріїв.

Критерій 1

Знайти такий допустимий розв'язок для якого виконується

$$\Delta_i \le l_i, l_i > 0, i = \overline{1,R} \qquad (5)$$

Критерій 2

Знайти такий допустимий розв'язок для якого виконується

$$\sum_{r=1}^{R} \alpha_r y_r \to min \qquad (6)$$

y_r тут – це величина перевищення Δ_r над заданим експертом обмеженням l_r.

Згідно [2] для знаходження компромісних розв'язків, що задовольняють критеріям (5) та (6) достатньо розв'язати наступну задачу

$$\sum_{r=1}^{R} \alpha_r y_r \to min \qquad (7)$$

$$\sum_{i=1}^{n} c_i^r x_i - \sum_{i=1}^{n} c_i^r x_i^r - y_r \le l_r, r = \overline{1,R} \qquad (8)$$

$$Ax = b, \qquad (9)$$

$$x_i \ge 0, i = \overline{1,n} \qquad (10)$$

$$y_r \ge 0, r = \overline{1,R} \qquad (11)$$

4. Опис процесу роботи системи підтримки дослідження оптимізаційних задач в умовах невизначеності

Весь процес роботи програми тісно пов'язаний із методикою оптимізації задач в умовах невизначеності.

Перший етап: процес роботи системи розпочинається із вибору користувачем файлу задачі та зчитування даних із нього. У файлі мають бути вказані дані для задачі, та вказано, параметри генератора, якщо він має застосовуватись..

Другий етап: наступний крок після зчитування – перетворення даних у задачі, які можуть бути розв'язані. Залежно від параметрів файлу, на цьому етапі або здійснюється перетворення, або відбувається генерація даних та їх перетворення.

Третій етап: після перетворення настає час розв'язання задач лінійного програмування, які є умовами нашої задачі в умовах невизначеності. Ці задачі лінійного програмування або описані у файлі вхідних даних, або згенеровані на попередньому кроці. Оптимізація цих задач лінійного програмування відбувається у паралельному режимі, що здійснено для економії часу та оптимізації робочого часу програми. На цьому етапі також відбувається вибраковка тих задач лінійного програмування, які не мають розв'язку.

Четвертий етап: після закінчення попереднього кроку, що відбувається одночасно із закінченням оптимізації усіх частковихзадач лінійного програмування, настає час формування задачі лінійного програмування(7)-(11), яка і дасть нам шукане компромісне рішення.

П'ятий етап: вирішення задачі лінійного програмування (7)-(11) та створення звіту. У звіті вказуються результати вирішення усіх частковихзадач лінійного програмування, задачі лінійного програмування (7)-(11).

Висновок

У цій публікації було розглянуто систему підтримки досліджень оптимізаційних задач в умовах невизначеності, окреслено область застосування цієї системи на прикладі області застосування задач в умовах невизначеності. Для задачі лінійного програмування в умовах невизначеності було наведена математичну постановкуНа основі методу, яким вирішується задача лінійного програмування в умовах невизначеності було створено алгоритм, який ліг у основу системи підтримки дослідження оптимізаційних задач в умовах невизначеності. Робота із цією системою дозволить провести дослідження, які можуть дозволити визначити нові властивості задачі та встановити вплив зміни вхідних даних на кінцевий результат.

Список літератури

1. *Pavlov A.A.* Optimization for one class of combinatorial problems under uncertainty. Адаптивні системи автоматичного управління. 2019. 1. № 34. С. 81–89. doi: 10.20535/1560-8956.1.2019.178233.
2. *Pavlov A.A.* Combinatorial optimization under uncertainty and formalmodels of expertes timation. Вісник Національного технічного університету «ХПІ». 2019. № 1. С. 3–7. doi: 10.20998/2079-0023.2019.01.01.
3. Alexander A. Pavlov, Elena G. Zhdanova The Transportation Problemunder Uncertainty // Journal of Automation and Information Sciences – 2020. – № 4 (52)– pp. 1-13. (**Scopus**)
4. Pavlov A. Zhdanova E. Finding a compromise solution to the transpor tationproblemunderuncertainty // Adaptivesystemsofautomaticcontrol. . – 2020. – № 1 (36)– С.60-72. (Фаховевидання)

УДК 004.94

ZHYDKOV V.

APP BUILDING SUPPORT SYSTEM AS PART OF THE IT-ENTERPRISE ERP SYSTEM

The subject of the article is one of the approaches to developing and maintaining multi-platform applications, namely such a system that will easily and quickly develop and maintain software. As part of the ERP-system IT-Enterprise, the option of implementing a multi-platform application using WEB-calculations is considered.
KEYWORDS:WEB-CALCULATIONS, DATABASE, DATABASE MANAGEMENT SYSTEMS, INTERFACES, USERS, SQL, INFORMATION BASE, INFORMATION MANAGEMENT SYSTEMS, SOFTWARE, DATABASE MANAGEMENT SYSTEMS, TECHNICAL FACILITIES

В даній роботі розглянуто один із підходів розробки та підтримки багатоплатформенних застосувань, а саме така система за допомогою якої буде легко та швидко розробляти і підтримувати програмне забезпечення. У складі ERP-системи IT-Enterprise розглянуто варіант реалізації багатоплатформенного застосування за допомогою WEB-розрахунків.
КЛЮЧОВІ СЛОВА:ВЕБ-РОЗРАХУНКИ, БАЗА ДАНИХ, СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ, ІНТЕРФЕЙС, КОРИСТУВАЧ, SQL, ІНФОРМАЦІЙНА БАЗА, ІНФОРМАЦІЙНА УПРАВЛЯЮЧА СИСТЕМА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ, КОМПЛЕКС ТЕХНІЧНИХ ЗАСОБІВ

1. Introduction

More and more applications are moving to multi-platform support in order to optimize the performance of their software, make it more versatile and easier to use. To implement the transition of software from one platform to another, it is necessary not only to optimize it for another platform, but to implement it in another programming language. To translate from one programming language to another can cause a number of problems, both with the

implementation itself, and with the limitations of the device that I use this software.

As multi-platform support requires a lot of time, it will be useful to develop a system that will greatly simplify and speed up the development and maintenance of these applications.

In the ERP-system[1] IT-Enterprise[2] there is a solution to the problem. With the help of WEB-calculations it is possible to easily specify the execution of the necessary actions and specify the input and output data, configure access to the execution ofthe developed WEB-calculations and with the help of Python configure the execution of the necessary tasks.

2. Web-services

The IT-Enterprise system provides a set of different tools for software data exchange with other software products.

One of these tools is Web-services[3] - the ability to publish a call to IT-Enterprise calculations to obtain data from IT-Enterprise, enter data into IT-Enterprise, receive analytical reports and summaries, etc.

Usually Web-services are used for integration with other software products, with their help third-party software has the ability to access data, and access to data can be provided remotely (via the Internet), and only authorized.

The list of available calculations, published in the form of web-services, is determined by the administrator of the IT-Enterprise system.

External programs can be both Web-applications, Windows-applications, and applications for mobile platforms - native-applications for Android, iOS, Windows Phone, etc.

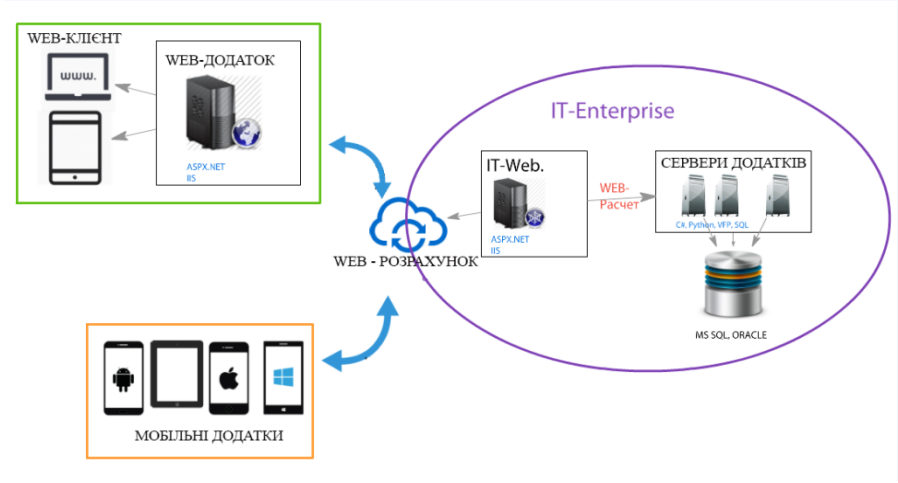


Figure 1. Scheme of interaction of IT-Enterprise with external applications

3. Setting up mobile WEB-calculation

To start working with a mobile application, its settings must include a URL link to the IT-Enterprise web services (the Internet address where the web service is published). This link can be rigidly "stitched" in the application or the application can have settings to be able to change this link.

By default, IT-Enterprise mobile applications are configured for IT-Enterprise demo project web services. Therefore, after installing the program, you must specify in the settings links to web-services deployed on the web-server and connect to it.

An example of a full link is: https://myCompany.com/ws/webservice.asmx.You can also specify an abbreviated version of the link: myCompany.com (data transfer protocol and /ws/webservice.asmx will be added to the application automatically).

Note: Starting with iOS 9.0 and higher, data exchange with web services for mobile applications running iOS must be performed over the HTTPS network protocol with traffic encryption over the cryptographic protocol TLS 1.2 (or higher).

4. Development of web-calculations

Development of web-calculations is performed in the function "Web-calculations" (Administration - Code Libraries - Web-calculations).

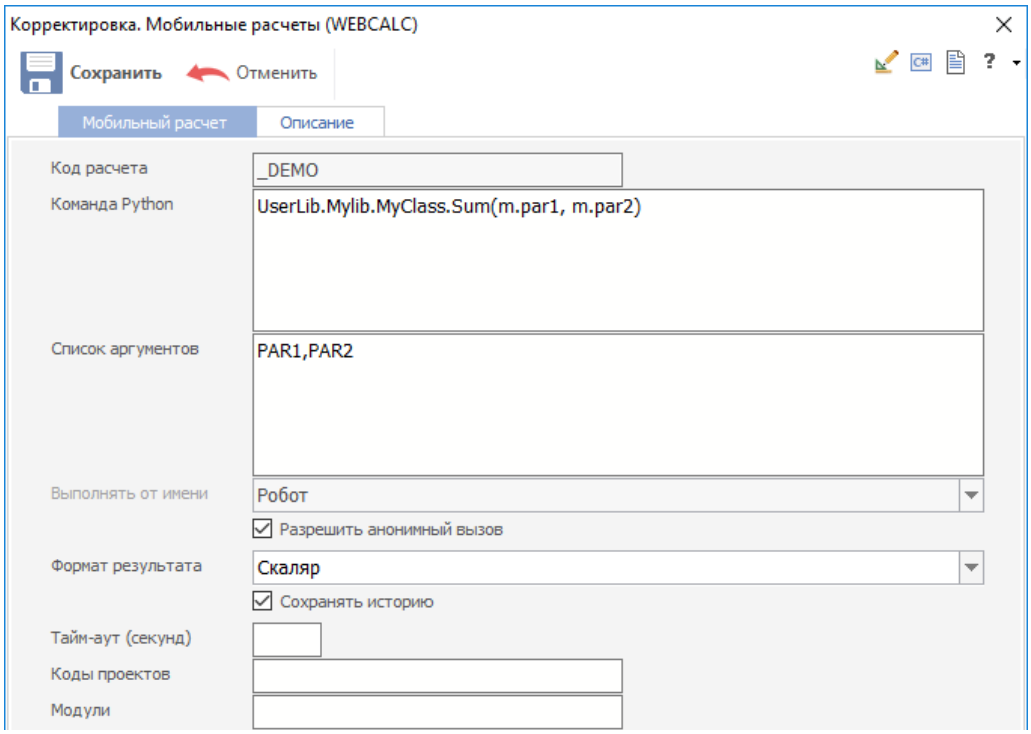


Figure 2. Web-calculation adjustment screen

Calculation code - a unique identifier of the calculation;

Iron python[4] command - a program in Python, which will be executed when calling a web-calculation.

Python is a flexible universal mechanism that allows you to use any IT-Enterprise functionality, such as:

- system functions of IT-Enterprise C # technology (for a full list of features, see in the API library);
- custom C # code libraries (for more information about custom code libraries, see in the document "Programmer's Guide. Development on the IT10 platform »);
- VFP programs (system and created by the customer's team) (for more information on the Python language and its application in IT-Enterprise, see in the documentation "Programmer's Guide. Development on the IT10 platform ").

Argument list - a list of parameter names accepted by the web-calculation. The values of the specified parameters are transferred from the parameters of the web-calculation call to the Python command.

Allow anonymous call - indicates whether it is possible to perform the calculation for users who have not passed the authentication procedure and have not received a temporary "ticket".

Perform on behalf of - this attribute indicates under which IT-Enterprise account you need to perform the calculation:

- under the account work (without impersonalization);
- under the account of the user who called the calculation;
- under the account of the user who called the calculation, including the SQL connection.

The use of impersonalization takes some time, which reduces the response rate of the system. Recommendations for the use of impersonalization:.

- if the calculation result is the same for all users, then do not use impersonalization;
- if the calculation behaves differently for different users, then use impersonalization.

Result format - indicates in what form the calculation returns its result to the calling party. Options:

- scalar: a simple value in the form of a string;
- collection of objects: list of values (lines, numbers, dates);

- dictionary of objects: list of values "key" - "value" (Dictionary <string, object>);
- table name in local SQL;
- set of tables in local SQL (names through a comma);
- arbitrary object: an instance of any public class, an anonymous object.

Formats of the value of the returned web-calculations:

- description: text description of the calculation. It must state:
- purpose of calculation;
- description of input parameters (their purpose, types, features of use);
- description of the returned result.

1. Usageformulti-platform

With such a wide range of web-calculation settings, it is possible to quickly configure the execution of any process, method or algorithm and easily use it with a web-service that calls the corresponding web-calculation and executes it on the server part. By describing the main functions in the main environment, there is no need to intervene in another environment, by calling the appropriate web-calculation, which in turn makes the application service, which reduces the requirements for the technical means.

Conclusion

This article offers an option to simplify the creation of multi-platform applications using WEB-calculations as part of the ERP-system IT-Enterprise. The possibility of setting up WEB-calculations was described in detail andwas described the principle of their operation and application in software development was described.

References

1. Odd JøranSagegg, ErlendAlfnes, ERP Systems for Manufacturing Supply Chains, 2020 pp. 11-46.
2. IT-Enterprise system[Electronic resource] //Access mode: https://www.it.ua/about-company.
3. Ethan Cerami, Web Services Essentials, O'Reilly Media, Inc., 2002 pp. 3-13.
4. John Paul Mueller, ProfessionalIronPythonDesign and Develop IronPython Techniques, Wrox Programmer to Programmer™, 2010 pp. 3-45

УДК 519.876.5

ВИХЛЯЄВА А.О.
ПОПЕНКО В.Д.

АНАЛІЗ МОДЕЛЕЙ РАНЖУВАННЯ ВЕРШИН У ГРАФАХ МЕРЕЖ РІЗНОГО ПРИЗНАЧЕННЯ

У роботі розглядається порівняльний аналіз методів обчислення центральності графу в мережах різного призначення. Центральність - одне з основних понять в аналізі соціальних мереж. Початкове визначення взаємозв'язку вузла в графіку базується на частці кількості найкоротших шляхів між будь-якими двома вузлами, на яких лежить даний вузол, до загальної кількості найкоротших шляхів, що з'єднують ці вузли. Цей метод має квадратичну складність і не враховує непрямі шляхи. Пропонується порівняти класичну центральність, централізацію потоку бета-струму, засновану на законі Кірхгофа для електричних ланцюгів, а також PageRank-центральність. Представлені результати чисельних експериментів для київського метрополітену.

Ключові слова: соціальна мережа, PageRank, центральність, електрична центральність.

The paper considers a comparative analysis of methods for calculating the centrality of the graph in networks for different purposes. Centrality is one of the main concepts in the analysis of social networks. The initial determination of the relationship of a node in the graph is based on the fraction of the number of shortest paths between any two nodes on which the node lies, to the total number of shortest paths connecting these nodes. This method has a quadratic complexity and does

not take into account indirect paths. It is proposed to compare classical centrality, beta-current centralization based on Kirchhoff's law for electrical circuits, and PageRank centrality. The results of numerical experiments for the Kyiv metro are presented.
Key words: social network, PageRank, centrality, electric centrality.

1. Вступ

Центральність є важливим інструментом мережевого аналізу [1]. У соціальних, біологічних, комунікаційних та транспортних мережах важливо мати відносну структурну визначеність вузлів або посилань для ідентифікації ключових елементів мережі. Структура мережі представлена графом, тому далі ми будемо говорити про вершини та ребра.

Два найбільш часто використовуваних показника - вершина між вершинами і центральна близькість вершин.

Вони базуються на припущенні, що інформація (що підлягає передачі) передається найкоротшими шляхами. Хоча центральність вимірює ступінь, до якої вершина знаходиться між парами інших вершин, тобто на найкоротших шляхах, що їх з'єднують, близькість є оберненою до середньої відстані відносно інших вершин.

Цей показник кількісно визначає частоту або кількість разів, коли вузол діє як міст по найкоротшому шляху між двома іншими вузлами. Інформація про посередницькі вузли представляє великий інтерес для багатьох практичних застосувань. Наприклад, у телекомунікаційних мережах центральність вимірює контроль, який здійснюється даним вузлом над інформаційним трафіком [2]. Однак у транспортних мережах ця центральність вимірює транспортний потік через вузли [3], тоді як у біологічних мережах це дозволяє ідентифікувати критичні гени [4]. Нарешті, у соціальних мережах він вимірює посередницьку здатність користувачів мережі. Але є багато мереж, в яких трафік не протікає за геодезичними шляхами [3,5]. Наприклад, у нас є пішохідні потоки в міських мережах, новини, чутки та повідомлення у всіх соціальних мережах, які зазвичай не передаються геодезичними шляхами, а поширюються по всій мережі до місця призначення.

2. Змістова постановка задачі

Аналіз соціальних мереж досліджує структуру взаємодії між соціальними сутностями. Цими соціальними сутностями можуть виступати люди, організації, міста, країни, веб-сторінки або наукові публікації. Пропонується розв’язати задачу методами ранжування вершин графу в транспортній мережі. В якості прикладу мережі траспортного характеру було обрано всім відомий київский метрополітен. Діють три лінії, експлуатаційна довжина яких становить 69,648 км, 52 станції зі трьома підземними вузлами пересадки в центрі міста.

Щоденний пасажиропотік на момент 2019 року складає 1,91 млн. Найбільш завантаженими станціями на 2011 рік є «Вокзальна» (68,3 тис. осіб/добу), «Лісова» (66,9тис.осіб/добу), «Почайна» (53,3 тис. осіб/добу), «Лівобережна» (52,6 тис. осіб/добу), «Дарниця» (51,8 тис. осіб/добу) та «Мінська» (50,8 тис. осіб/добу). Найменш завантаженими станціями є «Дніпро» (3,0 тис. осіб/добу), «Червоний хутір» (5,0 тис. осіб/добу), «Славутич» (6,3 тис. осіб/добу), «Гідропарк» (6,9 тис. осіб/добу) та «Вирлиця» (7,4 тис. осіб/добу)[97].

У 2018 році найбільш завантаженою була станція «Лісова». За 2019 рік найбільш завантаженою стала станція «Академмістечко», якою скористалося більше 21,3 млн пасажирів, найменший пасажиропотік зафіксовано на станції «Дніпро», яка прийняла 938,6 тисяч пасажирів. Загалом Святошинсько-Броварська лінія з 18 станцій перевезла 203 млн 176 тис. пасажирів. Другою за завантаженістю стала Оболонсько-Теремківська лінія, якою скористалося 173,2 млн пасажирів. Максимальний пасажиропотік на цій лінії зафіксовано на станції «Мінська» (понад 17 млн осіб/рік), мінімальний — «Поштова площа» (4,9 млн пасажирів). Сирецько-Печерською лінією за 2019 рік скористалися майже 118,8 млн пасажирів. Найбільші пасажиропотоки на зеленій лінії зафіксовано на станціях

«Лук'янівська» (14 млн осіб/рік) та «Позняки» (14,8 млн осіб/рік), найменше на цій лінії користувалися станцією «Червоний хутір» (1,8 млн осіб/рік). Всього за 2019 рік «Київський метрополітен» перевіз 495 млн 300 тис. пасажирів, що на 700 тисяч менше, ніж 2018 року.

Побудуємо граф на основі київського метрополітену та визначимо за допомогою методів обчислення класичної центральності, електричної центральності та PageRank найбільш навантажені станції метро та порівняємо роботу вищезгаданих алгоритмів. Визначимо найкращий метод центральності, який найбільше корелює з пасажиропотоком. На рисунку 1 зображено гілки київского метрополітену та рух швидкісного трамваю.

Рис. 1. Схема маршрутів київского метро та схема руху швидкісних трамваїв.

3. Математична постанова задачі

Дано зважений орієнтований граф $G = (V, E)$, де $V = \{1, 2, \dots, n\}$ – множина вершин, $E = \{(i, j)\}$ – множина ребер. Шлях або прогулянка визначається як послідовніть вузлів $\wp = (i_{0,\dots,} i_T)$, де $T > 0$ та $(i_{\tau,}\, i_{\tau+1}) \in E$ для усіх $\tau = 0, \dots., T - 1$. Шлях поглинається, якщо останній вузол шляху з'являється на ньому лише один раз. Набір усіх поглинаючих шляхів позначається починаючи від вузла *s* і закінчується у вузлі *t*, тобто *s*-*t*-шляхів або *s*-*t*-прогулянок, через $\mathcal{P}_{st}$.

Ваги на ребрах, $a_{i,j}$, $(i,\ j) \in$ E, відображають подібність або міцність зв'язку між сусідніми вузлами та утворюють матрицю суміжності **A**.

Вагу ребер визначають ймовірності переходу випадкового блукання $p_{ij}^{ref} = a_{ij} / \sum_i a_{ij}$. Ймовірності переходів утворюють еталонну матрицю ймовірностей переходу $\boldsymbol{P}^{ref}$, яка може бути обрахована як $\boldsymbol{P}^{ref} = \boldsymbol{D}^{-1}\boldsymbol{A}$, де $\boldsymbol{D}$ – діагональна матриця, що містить суми рядків матриці **A**.

На додаток до ваг, ребрам також присвоюються витрати c_{ij}, які, на відміну від ваг, можуть розглядатися також як несхожість або відстань сусідніх вузлів. Важливість шляху $\wp$ визначається як $\check{c}(\wp) = \sum_{(i,j)\in\wp} c_{ij}$. Відповідно, термін найкоротший шлях ми використовуємо для позначення шляху між двома вузлами з найменшими витратами за всі шляхи між вузлами. Позначимо множину найкоротших шляхів від вузла *s* до вузла *t* через $\mathcal{P}_{st}^*$, загальна сума таких шляхів складає $|\mathcal{P}_{st}^*|$ і вартість найкоротшого шляху від *s* до *t* позначається як $\widetilde{c_{st}^*}$. Спрямований граф, що складається лише з вузлів і спрямованих ребер, що належать до найкоротших шляхів від s до t, називається графом спрямованих найкоротших шляхів від *s* до *t*.

У багатьох ситуаціях витрати ребер та вагові значення ребер можуть бути визначені один на одному, наприклад, як взаємні величини $c_{ij} = 1/a_{ij}$, що відповідає інтерпретації витрат як опір та вага - як провідності в електричному колі. Ця конвенція також зазвичай використовується при обчисленні міри центральності у зважених мережах.

Відповідно, ймовірності переходу і, таким чином, неупереджена випадкова прогулянка можуть бути незалежними від витрат. Це означає, що витрати ребер визначають інтерпретацію найкоротших шляхів, тобто низькотемпературну поведінку системи, тоді як ваги ребер визначають інтерпретацію випадкової прогулянки, тобто поведінку високих температур. Таким чином, взаємозв'язок між вагами та витратами подібний до компромісу між пошуком, що базується на місцевих можливих переміщеннях, та

експлуатацією, що базується на довгострокових бажаних переміщеннях.

4. Опис методів розв'язання.

Класична центральність.

Центральність вершини дорівнює числу геодезичних шляхів між всіма вершинами, які проходять через задану вершину. Центральність вершини – це важлива міра, яа відображає те, наскільки вершина приймає участь в процесі розповсюдження інформації між іншими вершинами в графі. Формальне визначення міри центральності:

$$c_{\text{в}}(v)=\frac{1}{n_B}\sum_{s,t\in V}\frac{\sigma_{s,t}(v)}{\sigma_{s,t}}, \qquad (1)$$

де $\sigma_{s,t}$ – число геодезичних шляхів між вершинами *s* і *t*, $\sigma_{s,t}(v)$ – число геодезичних шляхів між вершинами *s* і *t*, які проходять через вершину *v*.

Коефіцієнт нормування n_B дорівнює $n_B=(n-1)(n-2)$, якщо вершина *v* не може бути початковою *s* або кінцевою *t* вершинами, та $n_B=n(n-1)$ інакше.

Обчислення міри центральності передбачає обчислення усіх геодезичних шляхів між всіма парами вершин у графі, котра потребує часу рівне $O(n^3)$ за допомогою алгоритму Флойда-Уоршелла.

Pagerank.

Pagerank [6] - широко відомий інструмент для ранжування веб-сторінок. Якщо всесвітня павутина розглядається як граф, то pagerank сторінки визначається як розподіл, який задовольняє такому лінійному рівнянню (2):

$$\mathbf{v}=(1-\alpha)P^T\mathbf{v}+\frac{\alpha}{n}\mathbf{1}, \qquad (2)$$

α - ймовірність перезапустити випадкове блукання на заданому кроці. Зауважемо, що коефіцієнт перезавантаження α відіграє дуже важливу роль. Теорема Перрона-Фробеніуса вказує, що стохастична матриця *P* має унікальний головний власний вектор, якщо вона незвідна і аперіодична. Випадковий перезапуск гарантує, що, за цією моделлю: 1) всі вузли доступні від усіх інших вузлів, і 2) ланцюг Маркова є аперіодичним. Також ймовірність перезапуску робить другу за величиною власну величину верхньою межею α [7]. Простіше кажучи, сторінки, на яких, швидше за все, з'явиться випадковий відвідувач, мають великі значення в стаціонарному розподілі ланцюга Маркова, який з ймовірністю α випадковим чином переходить відповідно до структури зв'язку в Інтернеті та з вірогідністю *1-α* телепортується відповідно до вектора телепортаційного розподілу **v**, де **v** зазвичай рівномірний розподіл на всі сторінки. Узагальнюючи, замінемо поняття "перехід відповідно до структури зв'язку мережі" на "перехід відповідно до стохастичної матриці P."

Проте, через те, що граф гіперпосилань реальної мережі Веб не завжди зв'язний та має тупикові вершини, у формулу випадкового блукання доводиться запровадити можливість випадкового «стрибка» на іншу вершину. Таким чином, формула обчислення розподілів знаходження **v`** матиме вигляд:

$$\mathbf{v`}=\alpha M\mathbf{v}+(1-\alpha)\frac{e}{n}, \qquad (3)$$

– α — константа, що зазвичай має значення в проміжку 0.8…0.9;

– *e* — одиничний вектор (розміром *n*, всі елементи якого дорівнюють 1);

– *n* — кількість індексованих сторінок, відповідно — кількість вершин графу;

– $\alpha M\mathbf{v}$ — моделює випадок, коли з ймовірністю α мандрівник вирішує обрати вихідне посилання з поточної сторінки;

– $(1-\alpha)\frac{e}{n}$ — вектор, кожен елемент якого дорівнює $(1-\alpha)/n$ і який моделює появу користувача на будь-якій сторінці з ймовірністю $(1-\alpha)$.

Персоналізований PageRank. PageRank надає демократичний погляд на важливість веб-сторінок. Однак враховуючи нерівномірну щільність ймовірності перезапуску ми можемо створити персоналізований розподіл відносної важливості вузлів. Основна ідея - почати випадкове блукання від вузла *i*; на будь-якому кроці руху переміщається випадковим чином до сусіда поточного вузла з ймовірністю *1 - α* і скидається на початковий вузол *i* з ймовірністю *α*. Стаціонарний розподіл цього процесу - це персоналізований Pagerank вузол *i*. В результаті стаціонарний розподіл локалізується навколо стартового вузла. Це

також називається «укорінений PageRank» [8]. Якщо ми узагальнимо стартовий вузол *i* до початкового розподілу *r*, вектор персоналізації буде заданий:

$$\mathbf{v} = (1-\alpha)P^T\mathbf{v} + \alpha r \qquad (4)$$

Для отримання персоналізації вузла *i*, розподіл перезавантаження встановлений на вектор, де елемент i^{th} встановлений на 1, а всі інші записи встановлені на 0. P^T *v* - це розподіл після одного кроку випадкового блукання з *v*. Вищезазначене визначення означає, що персоналізований PageRank може бути обчислений шляхом розв'язання великої лінійної системи, що включає матрицю переходу *P*.

Алгоритм PageRank обчислюється за такою формулою:

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right), \qquad (5)$$

де $PR(A)$- вага сторінки А(те, що ми шукаємо);

d - коефіцієнт затухання, який зазвичай встановлюють рівним 0,85;

PR(T1) – вага PadeRank сторінки, що посилається на сторінку А;

- *C(T1)* – кількість посилань із цієї сторінки;

- $\frac{PR(T_n)}{C(T_n)}$ – для кожної сторінки, яка вказує на сторінку А.

Електрична центральність.

Одним з недоліків класичної міри центральності є те, що в розрахунках враховуються лише геодезичні шляхи і не беруться до уваги шляхи, що можуть бути довшими геодезичних лише всього на один чи два кроки, хоча дані негеодезичних шляхів можуть мати важливу роль в описі і обгрунтуванні процесів, які протікають в комунікативних мережах.

В [7] граф розглядається як електричний ланцюг з ідеальними елементами, де кожне ребро має деяку пропускну здатність, а вершини графу є її вузлами. Для пошуку міри центральності в моделі електричного ланцюга використовують правила Кірхгофа. Для цього електричний ланцюг заземлюється в деякій вершині *t* і подається електричний струм в деякій вершині *s*. Струм подається в деякій єдиній вершині *s*, а також в єдиній вершині *t* мережа заземлюється. Мірою центральності вершини *v* за всіма можливими парами *s* і *t*.

Міра електричної центральності $s \in V$ в рамках моделі електричних ланцюгів була запропонована в [7] і визначається наступним чином:

$$CF_c(s) = \frac{n_C}{\sum_{t\neq s}(\varphi^{st}(s)-\varphi^{st}(t))}, \qquad (6)$$

де φ^{st} – абсолютний потенціал у вершині за умови, що джерело струму підключено до вершини s_1, а сток знаходитья у вершині *t*. Вираз $\varphi^{st}(s) - \varphi^{st}(t)$ відповідає чинному опору в електричному ланцюгу. Таким чином ефективний опір може служити деякою альтернативою для відстані між вершинами *s* та *t*.

5. Результати аналізу порівняння методів обчислення центральності графу в мережах різного призначення.

В таблиці 1 представлені результати обчислення класичної, електричної та PageRank центральності.

Табл. 1. Результати аналізу порівняння методів обчислення центральності графу в мережах різного призначення

Назва станції	**Класична центральність**	**PageRank - центральність**	**Електрична центральність**
Академмістечко	0.857	0.678493	0.874793
Житомирська	0.639	0.592942	0.638482
Святошин	0.536	0.678372	0.578438
Берестейська	0.347	0.573829	0.747389
Шулявська	0.449	0.375897	0.477825

Політехнічний інститут	0.679	0.783983	0.848390
Вокзальна	0.536	0.587874	0.383489
Університет	0.448	0.378927	0.448927
Театральна	0.328	0.649374	0.539849
Хрещатик	0.790	0.634899	0.573857
Арсенальна	0.621	0.846367	0.748930
Дніпро	0.312	0.433478	0.573828
Гідропарк	0.487	0.646434	0.574839
Лівобережна	0.624	0.573378	0.547378
Дарниця	0.547	0.643820	0.475284
Чернігівська	0.789	0.837478	0.683848
Лісова	0.812	0.874389	0.624583
Героїв Дніпра	0.637	0.638304	0.742949
Мінська	0.841	0.839020	0.630208
Оболонь	0.567	0.430348	0.639430
Почайна	0.662	0.737439	0.649583
Тараса Шевченка	0.432	0.650393	0.574727
Контрактова площа	0.389	0.483920	0.547897
Майдан Незалежності	0.758	0.784289	0.537482
Площа Льва Толстого	0.743	0.847349	0.634789
Олімпійська	0.539	0.463088	0.647372
Палац «Україна»	0.575	0.638478	0.746286
Либідська	0.468	0.573899	0.447899
Деміївська	0.732	0.843848	0.747892
Голосіївська	0.580	0.476349	0.477294
Васильківська	0.698	0.738434	0.648597
Виставковий центр	0.538	0.648382	0.743856
Іподром	0.553	0.538329	0.647378
Теремки	0.749	0.638483	0.567388
Сирець	0.438	0.573468	0.638481
Дорогожичі	0.350	0.473846	0.477930
Лук'янівська	0.678	0.738480	0.747331
Золоті ворота	0.493	0.547834	0.647399
Палац спорту	0.573	0.778376	0.647372
Кловська	0.478	0.467838	0.664838
Печерська	0.384	0.658938	0.475777
Дружби народів	0.763	0.647891	0.578368
Видубичі	0.732	0.894799	0.457583
Славутич	0.593	0.674892	0.557588
Осокорки	0.693	0.455993	0.538532
Позняки	0.458	0.663728	0.578738
Харківська	0.583	0.666372	0.573857
Вирлиця	0.535	0.647383	0.623589
Бориспільська	0.777	0.792848	0.457892
Червоний хутір	0.647	0.684029	0.745738

Висновки

У роботі розглядається порівняльний аналіз методів обчислення центральності графу в мережах різного призначення. Було порівняно класичну центральність, централізацію потоку бета-струму, засновану на законі Кірхгофа для електричних ланцюгів, а також PageRank-центральність. Представлені результати чисельних експериментів для київського метрополітену.

Перелік посилань

1. Stephen J. Hardiman and Liran Katzir. 2013. Estimating clustering coefficients and size of social networks via random walk. - 2013. - 539-550.
2. Erdos P. On Random Graphs. I / P.Erdos, A.Renyi. // Publicationes Mathematicae. – 1959.– №6.– С.290–297.
3. Guillaume J. Bipartite graphs as models of complex networks / J. Guillaume, M. Latapy. // CAAN. – 2004. – С. 215–221.
4. Newman M. Random graphs with arbitrary degree distributions and their applications / M. Newman, S. Strogatz, D. Watts. // Phys. Rev. E. – 2001. – №64.
5. P. Gundecha, H. Liu, Mining social media: a brief introduction, in: New Directions in Informatics, Optimization, Logistics, and Production. – 2012. - pp. 1–17.
6. Watts D. Networks, Dynamics, and the Small-World Phenomenon / D.J. Watts. // American Journal of Sociology. – 1999. – №105. – С. 493–527.
7. Barabasi A. Emergence of scaling in random networks / A. Barabasi, R. Albert. // Science. – 1999. – №286. – С. 509–512.
8. I. King, J. Li, K.T. Chan, A brief survey of computational approaches in social computing, in: 2009 International Joint Conference on Neural Networks, IEEE. – 2009.- pp. 1625-1632

УДК 004.021

САМАРА О.
ЖУРАКОВСЬКА О.С.

ІНФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА З ПІДБОРУ КОМПЛЕКТУЮЧИХ ДЛЯ ПК

У цьому дослідженні розглянуто задачу побудови класифікації для системи з підбору конфігурації ПК за комплектуючими по заданим критеріям. Наводиться опис існуючих методів, проаналізовано їх перегаги та недоліки. Було обрано метод дерев рішень та запропоновано спосіб його модифікації, з використанням на другому етапі генетичного алгоритму, який підвищить якість отриманих розв'язків.

КЛЮЧОВІ СЛОВА: ДЕРЕВО РІШЕНЬ, ГЕНЕТИЧНИЙ АЛГОРИТМ, ПРОБЛЕМА КЛАСИФІКАЦІЇ, ВИБІР КОМПЛЕКТУЮЧИХ ПК.

In this paper, the task is to build the classification for information retrieval system for selecting PC components according to the specified requirements. Guided by the description of existing methods, analyzed by the steps and shortcomings. The method of trees of solutions has been converted, and it has been pro- posed in the way of its modification, with changes based on a different stage of the genetic algorithm, which is to adjust the number of changes.

KEYWORDS: DECISION TREE, GENETIC ALGORITHM, RULE, CLASSIFICATION PROBLEM, SELECTION OF PC COMPONENTS.

Вступ

Щоб мінімізувати вплив недоліків на збірку ПК, створюються інструменти, які можуть допомогти людині в пошуку, пропонуючиті компоненти, які будуть максимально підходити під виконання

визначених ним завдань. Такі програмні засоби отримали назву "онлайн-конфігуратор". Це онлайн-конструктор, за допомогою якого можна самостійно підібрати комплектуючі для майбутнього комп'ютера або оновити вже існуючу систему. Для створення такого конфігуратора потрібно сформувати множину комплектуючих, ранжувати їх між собою, сформувати множину класів задач, з якими буде працювати комп'ютер в майбутньому, а також вирішити задачу класифікації [1].

Для вирішення задачі класифікації нами були розглянуті наступні методи:

- нейронні мережі;
- дерева прийняття рішень;
- метод найближчого сусіда;
- дискримінантний аналіз [2].

Огляд методів

З аналізу даних методів ми зробили висновок, що найбільш придатними методами є дерево прийняття рішень і байєсівська класифікація. Далі порівнюючи плюси і мінуси кожного з методів. Ми виявили, що основний недолік байєсівського класифікатора – відносно низька якість класифікації в більшості реальнихзадач. У класифікатора, заснованого на методі дерева прийняття рішень - проблема отримання оптимального дерева рішень з точки зору деяких аспекті в оптимальності, яку можна вирішити за допомогою генетичного алгоритму [3]. Пропонований нами алгоритм заснований на оптимізації методу дерев рішеньпід нашу систему.

Задача оптимізації методу дерева прийняття рішень

Ми знайшли алгоритм, якийдопоможе нам оптимізувати результат обраного нами методу - цегенетичний алгоритм.

Ідея алгоритму [4] полягає в тому, що ми спочатку використовуємо алгоритм дерева рішень для генерації правил класифікації комплектуючих, а потім відповідно атрибутам заданого правила, таким як сумісність, показник характеристик, рейтинг і різниця між ціною, ми можемо побудувати функцію генетичного алгоритму. Чим більше значення функції, тим більш оптимальним буде правило, яке і є оптимізацією обраного нами методу. Таким чином, ми можемо використовувати генетичний алгоритм для оптимізації результату дерева рішень в рамках нашої системи.

Змістовна постановка проблеми

Вхідні дані:

- U- множина комплектуючих;
- u_i- сумісність комплектуючих i множини U;
- c_i- ціна комплектуючих i множини U;
- c- очікувана ціна комплектуючих, яка була задана користувачем;
- r_i– рейтинг комплектуючих i множини U;
- h_i - показник характеристик комплектуючих i множиниU. I_i–функція показника правила (оптимізації).

Формула розрахункуфункціїпоказника правила:

$$Ii = \frac{ui(ri + hi)}{(2 - ui)(\Delta c + 1)} \quad ,$$

де $\Delta c = c - c1$

Процес виконання генетичного алгоритму

У генетичному алгоритмі процесс виглядає наступним чином:

Крок 1. Визначення атрибутів заданого правила: сумісність, показник характеристик, рейтинг і різніцяміж ціною.

Крок 2. Операції попередньої обробки: очищення даних, безперервна дискретизація атрибутів і обчислення коефіцієнтів передачі інформації для кожного атрибута функції.

Крок 3. Виконання кроків 4-6, поки не буде досягнуто потрібної кількості запропонованих конфігурацій ПК.

Крок 4. Оцінка придатності конфігурацій ПК шляхом обчислення цільової функції.

Крок 5. Вибір конфігурацій ПК.

Крок 6. Мутація[5], заміна неможливих конфігурацій ПК

Крок 7. Висновок рішення (найкращі варіанти конфігурацій ПК)

Висновок

У цій статті ми пропонуємо метод модифікованого рішення дерева на основі генетичного алгоритму. Він використовує можливість оптимізації генетичного алгоритму. Шляхом порівняння можна зробити висновок, що алгоритм, який ми запропонували в цій статті, був покращений за порівнянням із звичайним алгоритмом дерева рішень за точністю. За допомогою алгоритму було сгенерировано правило, яке дає більшу точність отриманого результату.

Посилання

1. Артіко Музаффар Егамбергановіч “Анализ эффективности применения методов классификации” pp. 5-6.
2. Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0.
3. Tobias Blickle and Lothar Thiele "A Comparison of Selection Schemes used in Genetic Algorithm", 1995, 2 Edition.
4. Д.И.Батищев, С.А.Исаев Оптимизация многоэкстремальных функций с помощью генетических алгоритмов./Межвузовский сборник научных трудов "Высокие технологии в технике, медицине и образовании", Воронеж, ВГТУ, 1997г, стр.4-17.
5. I. De Falcoa, A. Della Cioppab, E. Tarantino “Mutation-based genetic algorithm: performance evaluation”, SPAIM, National Research Council of Italy, Via Patacca 85, I-80056 Ercolano (NA), ItalybDepartment of Computer Science and Electrical Engineering, University of Salerno, Via Ponte Don Melillo 1, I-84084 Fisciano (SA), Italypp. 285–299.

УДК 519.816:004.9

АКІМОВ Д.Д.
ЖУРАКОВСЬКА О.С.

ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ ПРИ ФОРМУВАННІ АСОРТИМЕНТУ ПРОДУКЦІЇ ІНТЕРНЕТ МАГАЗИНУ

Робота присвячена підтримці прийняття рішень при формуванні асортименту продукції інтернет магазину. Задача що розглядається відноситься до задач динамічного програмування. Розглянуто підходи до розв’язання наведеного класузадач. Сформульовано постановку задачі. Наведено алгоритм розв’язання задачі.
ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ, АСОРТИМЕНТ ПРОДУКЦІЇ, ІНТЕРНЕТ-МАГАЗИН, ДИНАМІЧНЕ ПРОГРАМУВАННЯ

The work is devoted to decision support in the formation of the range of products of the online store. The problem under consideration refers to the problems of dynamic programming. Approaches to solving this class of problems are considered. The statement of the problem is formulated. An algorithm for solving the problem by this method is given.
NOMENCLATURE, ONLINE STORE, DECISION MAKING SUPPORT, DYNAMIC PROGRAMMING

1. Вступ

На сучасні умови економічного розвитку України впливають багато факторів, і тому торгові підприємства повинні пристосовуватись до сучасних реалій ведення бізнесу, до визначальної ринкової рушійної складової – конкурентного середовища. Останнє потребує створення адекватних до ринкових вимог правил, методів, механізмів та інструментів стратегічного планування і управління, особливо в умовах фінансової економічної кризи, яка охопила на сьогодні й Україну, та пандемії, що змусила змінити умови роботи більшості підприємств в країні.

Перед власниками торгових підприємств постала необхідність формування

асортименту продукції підприємства таким чином, щоб у керівника, або аналітика, були можливості для її подальшого аналізу, задля розуміння того, який товар користується попитом, який приносить найбільший прибуток, тощо.

Отже, метою даної роботи є підвищення ефективності процесу формування асортименту продукції інтернет магазину.

2. Використаний метод аналізу асортименту продукції

Для аналізу номенклатури використаємо метод динамічного програмування. Динамічне програмування в теорії управління і теорії обчислювальних систем - спосіб вирішення складних завдань шляхом розбиття їх на більш прості підзадачі. Він застосовується до завдань з оптимальною підструктурою, що виглядає як набір перекриваючих одне одну підзадач, складність яких трохи менше вихідної. В цьому випадку час обчислень, в порівнянні з стандартними методами, можна значно скоротити[1].

У випадку аналізу номенклатури, для досягнення мети – підвищення ефективності процесу формування асортименту продукції інтернет магазину, були виконані наступні задачі:

- визначення прибутку позицій номенклатури за певний період;
- прогнозування прибутку від покупки товару клієнтами у майбутньому періоді;
- закупівля товару згідно прогнозу;
- порівняння отриманого прибутку.

3. Постановка задачі

Задача, що буде вирішуватись, складається з двох етапів, що виконуються послідовно.

На першому етапі необхідно спрогнозувати кількість проданого товару по кожному виду продукції.

На другому етапі потрібно скласти оптимальний план закупівель продукції, з врахуванням прогнозованих значень на першому етапі.

3.1. Задача першого етапу. Прогнозування обсягів продажу усіх видів продукції

Підприємство продає m видів товарів A_m впродовж n періодів $t_1,t_2,\ldots,t_n$.

Відомі обсяги продажів кожного з цих товарів A_i за вказані періоди

$$y_1^i, y_2^i, \ldots, y_n^i.$$

Необхідно для кожного виду продукції A_i, i=1,…,m визначити прогнозоване значення a_i продажу в період n +1:

$$a_i = y_{n+1}^i = c_i + d_i t_{n+1}, \quad i=1,\ldots,m \qquad (1)$$

де c_i, d_i – коефіцієнти лінійної регресії

$$y^i = c_i + d_i t, \quad i=1,\ldots,m,$$

які визначаються за допомогою методу найменших квадратів.

Отримані розв'язки задачі (1)– прогнозні значення продажів a_i, i=1,…,m у періоді n+1 – використовуються в якості параметрів моделі другого етапу формування оптимального асортименту продукції в періоді n+1.

3.2. Задача другого етапу. Формування асортименту продукції на основі вирішення задачі про оптимальне використання ресурсу

Для формування асортименту продукції на наступний період передбаченона закупівлютоварів $A_i, i = 1, \ldots, m$ виділити N гривень.

Необхідно сформувати такий асортимент продукції x_i, i=1,…,m, де x_i – кількість продукції виду A_i, який би забезпечив максимальний прибуток, якщо попит на продукцію кожного виду складає відповідно a_i, i=1,…,m(результат вирішення задачі (1)), витрати на закупку одиниці продукції A_i, i=1,…,mскладають p_i, прибуток від продажу одиниці продукції дорівнює c_i, а витрати на зберігання одиниці непроданої продукції складають t_i.

Якщо обсяг проданої продукції менший за попит на цей вид продукції, то будемо вважати, що за вказаний період було втрачено долю ринку за вказаним видом продукції, що відображено в моделі коефіцієнтом втрат долі ринку на одиницю продукції k_i.

Для побудови моделі для кожного виду продукції A_i, i=1,…,m розглянемо можливі випадкипланування його обсягу в загальному асортименті:

1. $a_i < x_i$;
2. $a_i > x_i$;
3. $a_i = x_i$.

Якщо обсяг продукції x_iперевищуватиме попит на цей вид продукції (випадок 1), то прибуток, отриманий з продажу цього виду продукції складатиме$a_i \cdot c_i$, а витрати складатимуться з витрат на закупівлю та зберігання надлишкової кількості продукції$x_i \cdot p_i + (x_i - a_i)t_i$.

Якщо обсяг продукції x_iбуде меншим за попит на цей вид продукції (випадок 2), то прибуток, отриманий з продажу цього виду продукції складатиме$x_i \cdot c_i$, а витрати складатимуться з витрат на закупівлю та штрафу за втрату долі ринку на цей вид продукції$x_i \cdot p_i + (a_i - x_i)k_i$.

У випадку, коли обсяг продукції дорівнюватиме попиту (випадок 3), прибуток складатиме$a_i \cdot c_i$, а витрати - $a_i \cdot p_i$.

Таким чином, задача формування асортименту продукції може бути сформульована як багатокроковий процес, де на кожному кроці приймається рішення про планування обсягу продукції x_i, i=1,…,m, тобто є задачею про оптимальне використання ресурсу.

Будемо по кожному виду продукції розглядати три можливі випадки планування його обсягу в загальному асортименті, як показано вище, вважаючи, що $|a_i - x_i| = \Delta_i$. Значення розмірів витрат $P_i(x_i)$ та прибутку$C_i(x_i)$ в кожному випадку по кожному виду продукції $A_i, i = 1, \dots, m$подамо у вигляді таблиці1.

Отже, задача полягає в формуванні такого асортименту продукції x_i, i=1,…,m, який забезпечить максимальний прибуток при значеннях витрат і прибутків по кожному виду продукції, визначених в таблиці 1 та значеннях попиту на кожен вид продукції a_i, i=1,…,m,отриманих в результаті вирішення задачі етапу 1 - прогнозування попиту на кожен вид продукції.

Таблиця1 – Розмір витрат та прибуткупо видам продукції, що планується

	Товар A_1		*Товар* A_2		…	*Товар* A_m	
	$P_1(x_1)$	$C_1(x_1)$	$P_2(x_2)$	$C_2(x_2)$		$P_m(x_m)$	$C_m(x_m)$
$x_i = a_i$	$x_1 p_1$	$x_1 c_1$	$x_2 p_2$	$x_2 c_2$		$x_m p_m$	$x_m c_m$
$x_i = a_i - \Delta_i$	$x_1 p_1 + \Delta_1 k_1$	$x_1 c_1$	$x_2 p_2 + \Delta_2 k_2$	$x_2 c_2$		$x_m p_m + \Delta_m k_m$	$x_m c_m$
$x_i = a_i + \Delta_i$	$x_1 p_1 + \Delta_1 t_1$	$a_1 c_1$	$x_2 p_2 + \Delta_2 t_2$	$a_2 c_2$		$x_m p_m + \Delta_m t_m$	$a_m c_m$

4. Опис алгоритму

Задача формування асортименту продукції може бути вирішена за два етапи. На першому етапі в результаті вирішення задачі прогнозування визначаються значення попиту на кожен вид продукції. Для цього для визначення параметрів лінійної регресії по кожному виду продукції застосовується метод найменших квадратів [2].

На другому етапі вирішення отримані значення попиту на кожен вид продукції використовують як параметри моделі в задачі про оптимальне використання ресурсу. Для ефективного вирішення цієї задачі може бути застосований метод динамічного програмування [1].

Оптимальна підструктура в динамічному програмуванні означає, що оптимальне рішення підзадач меншого розміру може бути використано для розв'язання вихідної задачі[1].

У загальному випадку ми можемо вирішити задачу, в якій присутній оптимальна підструктура, пророблюючи наступні три кроки:

1. розбиття задачі на підзадачі меншого розміру;
2. знаходження оптимального рішення підзадач рекурсивно, пророблюючи такий же трьохкроковий алгоритм;
3. використання отриманого рішення підзадач для конструювання рішення вихідної задачі.

Підзадачі вирішуються розподілом їх на підзадачі ще меншого розміру і так далі, поки не приходять до завдання, що вирішується тривіально.

Розглянемо схему розв'язання задач, що використовуються в роботі.

4.1. Метод найменших квадратів

Задачаполягає в знаходженні коефіцієнтів лінійної регресії, при яких функція двох змінних a і b $F(a,b)=\sum_{i=1}^{n}(y_i-(ax_i+b))^2$ приймає найменше значення. Тобто, при даних a й b сума квадратів відхилень експериментальних даних від знайденої прямої буде найменшою.

4.2. Алгоритм вирішення задачі про оптимальне використання ресурсів

Задача формування асортименту може бути представлена як багатокроковий процес прийняття рішень, коли на кожному кроці приймається рішення про закупку *j*-го виду продукції,і на кроки 1..*j* використані кошти в розміріy_j.Для вирішення її методом динамічного програмування побудуємо для нього рекурентне співвідношення.

Нехай $f_j(y_j)$ – максимальний прибуток, який можна отримати шляхом розподілу y_j одиниць ресурсу на купівлю продуктів з номерами від 1 до j включно. Тоді можна купити *j*-йвид продукції в кількості x_j, якщо буде використано ресурсу $P_j(x_j)\leq y_j$, де $x_j \in D_j$, $D_j=\{a_j-\Delta_j, a_j, a_j-\Delta_j\}$.

Якщо *j*-й продукт закупається в кількості x_j і для цього витрачено ресурсу в кількості $P_j(x_j)$, то на закупівлю з 1-го до *j–1*-го продуктів буде використано $y_j-P_j(x_j)$ одиниць ресурсу, а максимальний прибуток від розподілу y_j одиниць ресурсу на закупівлю продуктів від 1-го до*j*-го, якщо на кроці *j* закупається *j*-й продукт в кількості x_j складає

$$C_j(x_j)+ f_{j-1}(y_j- P_j(x_j)) \qquad (2)$$

Вираз (2) є умовно-максимальним прибутком, тому максимізуємо його по допустимих значеннях x_j:

$$f_j(y_j)=\max_{x_j\in D_j | Pj(x_j)\leq y_j}\{C_j(x_j)+ fj-1(yj-Pj(xj))\}, \qquad (3)$$

де $f_0(y_1-P_1(x_1))=0$.

Отже, метою вирішення задачі є знаходження $f_m(N)$, що визначається співвідношенням (3).

4.3. Планування досліджень ефективності розробленої двохетапної схеми вирішення задачі формування асортименту продукції

Для дослідження ефективності розробленої схеми вирішення задачі формування асортименту продукції необхідно для деякого планового періоду *n*+1, …, *n*+*s*визначити асортимент продукції у відповідності із запропонованою схемою та порівняти за розрахунковим значенням прибутку із асортиментом, який формується за традиційною схемою, що використовується в системі.

Висновок

В даній роботі були розглянуті методи аналізу асортименту продукції інтернет магазину. Показано, що задача формування асортименту може бути вирішена в два етапи. Запропоновано схему її вирішення, яка полягає у вирішенні на першому етапі задачі прогнозування для отримання прогнозних значень попиту на всі види продукції, які використовуються як параметри моделі в задачі другого етапу. На другому етапі запропоновано формувати асортимент продукції в результаті вирішення задачі про оптимальне використання ресурсів, використовуючи метод динамічного програмування. Запроновано схему дослідження ефективності двохетапної схеми вирішення задачі формування асортименту продукції.

Перелік посилань

1. Зайченко Ю.П. Дослідження операцій. Підручник. – К.:ВД «Слово», 2006
2. Доросинский Л.Г. Основы теории принятия решений: LAP LAMBERT Academic Publishing, 2014

УДК 004.912

ДВОРНИК В. А.

КЛАСИФІКАЦІЯ ПРОФІЛІВ СОЦІАЛЬНИХ МЕРЕЖ КОРИСТУВАЧІВ НА ОСНОВІ П'ЯТИФАКТОРНОЇ МОДЕЛІ ОСОБИСТОСТІ

У даній статті розглянуто проблему підбору кадрів для ІТ-компаній. Розглянуто практичне застосуванняметодів класифікаціїна прикладі задачі класифікації профілів соціальних мереж кандидатів при підборі кадрів для ІТ-компаній на основі п'ятифакторної моделі особистості. Приведено постановку задачі.Розглянуто визначення п'ятифакторної моделі особистості. Визначено, до яких моделей зводяться досліджувані проблемні ситуації, та які методи можуть бути застосовані до розв'язання поставленої задачі, також висвітлено переваги та недоліки цих методів.

КЛЮЧОВІ СЛОВА: КЛАСИФІКАЦІЯ, ПРОФІЛЬ СОЦІАЛЬНОЇ МЕРЕЖІ, ВЕЛИКА П'ЯТІРКА, КАНДИДАТ, РЕКРУТЕР, ПІДБІР КАДРІВ.

This article discusses the problem of recruitment for IT companies. The practical application of classification methods on the example of the problem of profiles' classification of candidates' social networks in the selection of personnel for IT companies on the basis of a five-factor modelis considered. The statement of the problem is given. The definition of the five-factor model of personality is considered. Also it was determined to which models the researched problem situations are reduced, and which methods can be applied to the solution of the settask, also the advantages and disad vantages of the semethod sarehighlighted.
KEYWORDS:CLASSIFICATION, SOCIAL NETWORK PROFILE, BIG FIVE PERSONALITY TRAITS CANDIDATE, RECRUITMENT, RECRUITER.

1. Вступ

На сьогоднішній день підбір кандидатів для найму з широкого кола кандидатів є основоположним питанням. Іноді менеджери з підбору персоналу витрачають багато енергії і часу в пошуках гідного і благонадійного співробітника в свою компанію.

В умовах жорсткої конкуренції на ринку, коли репутація та корпоративна безпека компанії встають на перший план, кожен підприємець мріє про чесних, відповідальних та ефективних співробітників. Завдання підібрати такий персонал лягає на плечі кадровика. Іноді HR-фахівці впадають у відчай, розчаровані після чергової невдалої співбесіди. Проте можна було б заощадити їх час та сили, попередньо оцінивши кандидата в відкритих джерелах в інтернеті [1].

2. Постановка задачі

У задачі класифікації профілів соціальних мережмайбутніх працівників при підборі кадрів для ІТ-компаній в якості вхідної інформації розглядатимуться профілі соцмереж та їхнє наповнення: особиста інформація, пости, групи.

На виході отримаємо велику п'ятірку особистості для кожного профілю з відсотком, який вказує на присутність у профілі тих чи інших ознак.

Для класифікації вхідних даних пропонується побудова класифікатора особистості претендента.

Таким чином, нехай X – множина описів об'єктів, тобто профілів соціальних мереж, Y – множина номерів (чи назв) класів великої п'ятірки. Існує невідома цільова залежність-відображення$y^*: X \to Y$, значення якої відомі лише на елементах скінченної навчальної вибірки$X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$.
Потрібно побудувати алгоритм$a: X \to Y$, здатний класифікувати довільний об'єкт $x \in X$.

3. Огляд «великої п'ятірки»

Велика п'ятірка – це ієрархічна модель особистості, що представляє п'ять головних рис, що складають особистість людини.

Велика п'ятірка була сформована наприкінці 80-их років XX ст. у межах лексичної моделі, що розвиває

напрацювання Г. Олпорта, Р.-Б. Кеттела, Л.-Л. Терстоуна.

Її основна ідея полягає у тому, що всі суттєві поведінкові та психологічні відмінності обов'язково фіксують у мові, тому побутові і літературні висловлювання, що характеризують людську зовнішність, відображають системоутворювальне ядро особистості.

П'яти факторна модель, або модель «Великої п'ятірки», була підтверджена і психометричними дослідженнями. Виокремлені фактори мають високу конвергентну валідність, виявляються в різних підходах.

Для кращого запам'ятовування в англомовній літературі п'ять рис складають в акронім «OCEAN».

Openness – відкритість досвіду. Схвальне сприйняття мистецтва, емоцій, пригод, незвичайних ідей, цікавість та різноманітність досвіду

Conscientiousnessозначає надійність та сумлінність. Тенденція бути організованим та надійним, демонструвати самодисципліну, діяти слухняно, прагнути до досягнення і віддавати перевагу плановані, а не спонтанній поведінці, безтурботності та недбалості.

Extraversion– екстраверсія. Енергія, позитивні емоції, комунікабельність, тенденція шукати стимуляцію в компанії інших та балакучість. На противагу маємо спокій та пасивність.

Agreeableness– доброзичливість, приємність. Тенденція бути співчутливим та кооперативним, а не підозрілим,ворожим, та егоїзтичним. Це також є показником довірливого характеру, а також, чи є людина загалом добре вихованою чи ні.

Neuroticism–невротизм. Тенденція бути схильним до психологічного стресу. Тенденція легко відчувати неприємні емоції, такі як гнів, тривога, депресія, пригніченість та уразливість. Невротизм також відноситься до ступеня емоційної стійкості та імпульсного контролю, і його іноді називають «емоційною стійкістю».

П'ятифакторна теорія рис особистості ґрунтується на таких постулатах:

– всі дорослі люди можуть бути охарактеризовані специфічною комбінацією особистісних рис, що впливають на думки, відчуття і поведінку (про індивідуальність);

– риси особистості, що вивчаються, є ендогенними базовими тенденціями (про походження);

– риси розвиваються в дитинстві, остаточно формуються в дорослому віці і зберігають свою незмінність у адаптованих суб'єктів (про розвиток);

– риси організовані ієрархічно, від вузьких і специфічних до широких, узагальнених диспозицій (про структуру) [2].

4. Опис методу розв'язання задачі

Розглянемо методи класифікації. Класифікація – це система розподілу предметів, явищ або понять певної області на класи, розділи і розряди. В свою чергу класифікація текстів (документів) (англ. Documentclassification) – завдання комп'ютерної лінгвістики, яка полягає у віднесенні документа до однієї з декількох категорій на підставі змісту документа [3].

В контексті машинного навчання класифікація відноситься до навчання з учителем. Такий тип навчання передбачає, що дані, що подаються на входи системи, вже помічені, а важлива частина ознак вже розділена на окремі категорії або класи. Тому мережа вже знає, яка частина входів важлива, а яку частину можна самостійно перевірити.

При навчанні без учителя в систему подаються непомічені дані, і вона повинна спробувати сама розділити ці дані на категорії.

Процес навчання моделі – це подача даних для нейромережі, яка в результаті повинна вивести певні шаблони для даних. В процесі навчання моделі з учителем на вхід подаються ознаки і мітки, а при прогнозуванні на вхід класифікатора подаються тільки ознаки.

Прийняті мережею дані діляться на дві групи: набір даних для навчання і набір для тестування. Не варто перевіряти мережу на тому ж наборі даних, на яких вона навчалася, так як модель вже буде «заточена» під цей набір.

На сьогоднішній день є багато різних алгоримів класифікації, наведемо основні:

– метод k-найближчих сусідів (K-Nearest Neighbors);

– метод опорних векторів (SupportVectorMachines);

– класифікатор дерева рішень (DecisionTreeClassifier) / Випадковий ліс (RandomForests);

– наївний баєсівский метод (NaiveBayes);

– лінійний дискримінантний аналіз (LinearDiscriminantAnalysis);

– багатошаровий персептрон по Румельхарту (Multi-layerPerceptronClassifier) [4].

Метод k-найближчих сусідів працює за допомогою пошуку найкоротшої дистанції між тестованим об'єктом і найближчими до нього класифікованими об'єктами з навчального набору. Об'єкт, який классифікується буде відноситись до того класу, до якого належить найближчий об'єкт набору. На рисунку 1 показано ідею методу k-найближчих сусідів [4].

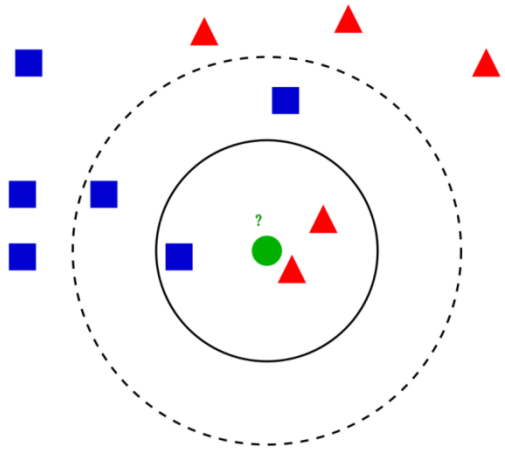


Рис. 1. Ідея методу k-найближчих сусідів

Метод опорних векторів використовує процес пошуку площини вирішення, яка може розділити позитивні та негативні приклади у багатовимірному просторі функції, в якому навчальні документи представлені як вектори.

Формальніше, опорно-векторна машина будує гіперплощину, або набір гіперплощин у просторі високої або нескінченної вимірності, які можна використовувати для класифікації, регресії та інших задач.

Інтуїтивно, добре розділення досягається гіперплощиною, яка має найбільшу відстань до найближчих точок тренувальних даних будь-якого з класів. На рисунку 2 помітно, що H_1 не розділяє ці класи. H_2 розділяє, але лише з невеликим розділенням. H_3 розділяє їх із максимальним розділенням.

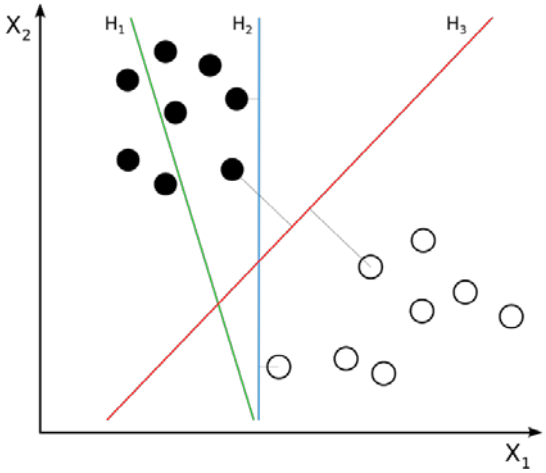


Рис. 2. Ідея методу опорних векторів

Цей метод розроблений Володимиром Вапником у 1995 році, був вперше застосований до задачі класифікації текстів ТорстеномДжохімсом. У своєму первинному вигляді алгоритм вирішував задачу розрізнення об'єктів двох класів. Метод набув популярність завдяки своїй високій ефективності. Багато дослідників і досі використовують його в роботах, присвячених класифікації текстів [5].

Класифікатор дерева рішень розбиває дані на все менші і менші підмножини на основі різних критеріїв, тобто у кожної підмножини своя категорія. З кожним поділом кількість об'єктів певного критерію зменшується.

Внаслідок розбиття виникає ієрархія операторів «якщо-то», які класифікують дані.

Класифікація закінчиться, коли мережа дійде до підмножини тільки з одним об'єктом. Якщо об'єднати кілька подібних дерев рішень, то вийде так званий Випадковий Ліс (англ. RandomForest).

Наївнийбаєсівський класифікатор обчислює ймовірність приналежності об'єкта до якогось класу. Ця ймовірність обчислюється з шансу, що якась подія відбудеться, з опорою на вже на події, що відбулися.

Кожен параметр об'єкта, який класифікується вважається незалежним від інших параметрів.

Лінійний дискримінантний аналіз працює шляхом зменшення розмірності набору даних, проектуючи всі точки даних на лінію. Потім він комбінує ці точки в класи, базуючись на їх відстані від центральної точки.

Цей метод відноситься до лінійних алгоритмів класифікації, тобто, він добре підходить для даних з лінійною залежністю [4].

Багатошаровий персептрон – це керований алгоритм навчання, який вчить функцію, навчаючись на наборі даних, де є кількість вимірів для введення та кількість вимірів для результату. Враховуючи набір ознак і цільовий показник, він може навчитися нелінійному апроксиматору функцій для класифікації або регресії. Він відрізняється від логістичної регресії тим, що між вхідним і вихідним шаром може бути один або кілька нелінійних шарів, які називаються прихованими шарами. На рисунку 3 показаний один прихований шарперсептрону зі скалярним виходом [6].

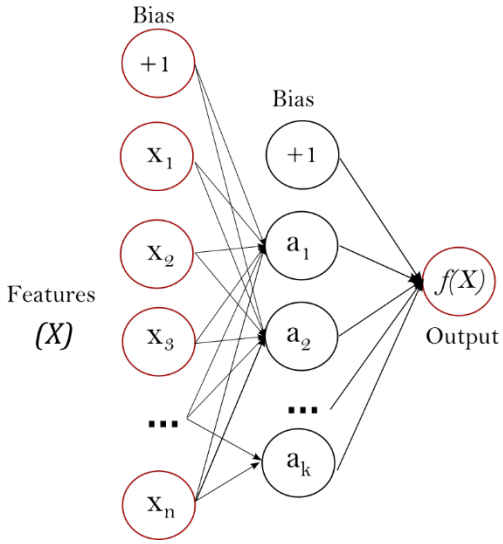


Рис. 3. Багатошаровий персептрон

4. Обґрунтування вибору методів дослідження

Розглянуті алгоритми класифікації достатньо прості, легко реалізуються та показують достатньо високу якість роботи. Незважаючи на це, алгоритми мають і свої недоліки. Наведемо далі переваги та недоліки методу *k*-найближчих сусідів:

Переваги:

– простота реалізації;

– класифікацію, проведену даним алгоритмом, легко інтерпретувати шляхом

пред’явлення користувачеві декількох найближчих об’єктів.

Недоліки:

– неефективна витрата пам’яті;

– пошук найближчого сусіда передбачає порівняння об’єкта, який класифікується з усіма об’єктами вибірки, що вимагає лінійного по довжині вибірки числа операцій.

Наведемо далі переваги та недоліки методу методу опорних векторів:

Переваги:

– задача добре вивчена та має єдине рішення;

– метод опорних векторів еквівалентний двошарової нейронної мережі.

Недоліки:

– нестійкість до шуму;

– не описані загальні методи побудови ядер і просторів, найбільш придатних для конкретного завдання;

– необхідно підбирати константу за допомогою крос-валідації.

Одним з головних недоліків методу дерев рішень для задач класифікації текстів є тойфакт, що алгоритм побудови дерев рішень надає однакову вагу «позитивним» і «негативним» розгалуженням у вузлах. Велика кількість «негативних» гілок в описі рубрики можепризводити до правил, котрі важкоінтерпретуються та«перенавчанню» алгоритму класифікації.

До переваг зазначеного методу слід віднести той факт, що побудоване дерево легкопіддається аналізу, результат роботи алгоритму можна інтерпретувати в наочних термінах.

Основні переваги наївного баєсівського класифікатора:

– простота реалізації;

– низькіобчислювальні витрати при навчанні та класифікації.

Основним недоліком методу є відносно невисока якість класифікації в більшості реальних завдань.

Перевага методу лінійного дискримінантного аналізу– порівняльна простота реалізації та інтерпретації результатів. Недолік – чутливість до розподілу вихідних даних, коли навіть невелика їх зміна призводить до значних змін результатів класифікації.

Метод багатошарового персептрону має високу точність вобробці одночасно лінійних і нелінійних прикладів, але прийняття рішень щодо класифікаціїважко формалізуються у зв’язку з природою організації нейронної мережі та представляютьнетривіальну задачу з урахуванням масштабованості з обмеженими обчислювальнимиресурсами [5].

Висновки

Застосування методів класифікаціїдо аналізу соціальних мереж майбутніх працівників дає можливість отримативелику п’ятірку особистості для кожного профілю з відсотком, який вказує на присутність у профілі тих чи інших ознак, тобто визначити психотип людини. Для менеджерів з підбору персоналу це насамперед можливість заощадження їх часу та сил, попередньо оцінивши кандидата в відкритих джерелах в інтернеті.

У дослідженнях методурозв’язання було розглянутовизначення п’ятифакторної моделі особистості, також розглянуто методи класифікації та виявлено, що розглянуті алгоритми класифікації достатньо прості та показують високу якість роботи. Також наведено переваги та недоліки розглянутих методів класифікації.

Список використаних джерел

1. Аналіз особистості по соціальним мережам як ефективний метод підбору кадрів [Електронний ресурс] – Режим доступу до ресурсу: https://srccs.su/analiz-lichnosti-po-sotsialnym-setyam-podbor-kadrov/
2. Галян І.М. Психодіагностика: Навчальний посібник / І.М.Галян. -К.: «Академвидав», 2009. – 463 с.
3. Batura, Tatiana. (2017). Методи автоматичної класифікації текстів. Міжнароний журнал Програмні продукти і системи. 23. 85-99. 10.15827/0236-235X.117.085-099.
4. Огляд методів класифікації в машинному навчанні [Електронний ресурс] – Режим доступу до ресурсу: https://tproger.ru/translations/scikit-learn-in-python/
5. Волосюк Ю.В. Методи класифікації текстових документів в задачах TextMining /Ю.В. Волосюк // Наукові записки Українського науково-дослідного інституту зв'язку. –2014. – №. 6. – С. 76-81.
6. Моделі нейронних мереж [Електронний ресурс] – Режим доступу до ресурсу: https://scikit-learn.org/stable/modules/neural_networks_supervised.html

УДК 519.8

МАРТИНЮК Ю.Ю.
СПЕРКАЧ М.О.

ІНФОРМАЦІЙНА СИСТЕМА ПЛАНУВАННЯ РЕСУРСІВ ІТ-ПРОЕКТІВ

Важливу роль в управлінні ІТ-проєктами відіграє процес планування ресурсів. Одними з основних типів ресурсів, що потребують додаткової уваги в управлінні проєктами є планування людських та часових ресурсів. У роботі наведено існуючі підходи до управління проектами. Сформульовано постановку задачі. Наведено дослідження властивостей задачі. Запропоновано евристичні алгоритми розв'язання задачі, що базуються на методах жадібного алгоритму та алгоритму локального пошуку. Проведено дослідження ефективності алгоритмів щодо швидкодії та ефективності роботи обраних методів.
КЛЮЧОВІ СЛОВА: УПРАВЛІННЯ ПРОЕКТАМИ, SCRUM, БЕКЛОГ, СПРИНТ, РЕСУРС, ТЕОРІЯ РОЗКЛАДІВ, МІНІМІЗАЦІЯ ЧАСУ ВИКОНАННЯ ЗАВДАНЬ

The resource planning process has an important role in IT project management. One of the main types of resources that requires additional attention in project management - it is human and time resource planning. The work presents the existing approaches to project management. The statement of the problem is formulated. There is given a study of the properties of the problem. Heuristic algorithms for solving the problem are based on the methods of greedy algorithm and local search algorithm are proposed. Also there is spent the research of efficiency of algorithms concerning speed and efficiency of work of the chosen methods.
KEY WORDS: PROJECT MANAGEMENT, SCRUM, BACKLOG, SPRINT, RESOURCE, SCHEDULE THEORY, MINIMIZATION OF TASK PERFORMANCE

Вступ

ІТ-індустріясьогодення є однією з потужніших сфер економіки зокрема та всієї людської діяльності. Кожну нову справу можна представити як проєкт, для успішної реалізації якого, виникає необхідність у чіткому плануванні та менеджменті. Успіхреалізаціїпроектівнапряму залежить від рівня ефективностіуправління проектом.

Управління проектом [1] — це методологія планування, організації, лідерства, розподіленнялюдських, фінансових та матеріальних ресурсів під час реалізації проекту, яка спрямована на досягнення його цілей за допомогоювикористання сучасних методів, технік та технологій управління проектом.

Успіх проекту залежить від ряду показників, які визначають його суть та впливають на стан ІТ-проекту за увесь час його життєвого циклу. Ця множина елементів проекту власне і є об'єктами управління. Управління людськими ресурсами та часом є одними з важливих об'єктів управління при реалізації проєкту.

Управління людськими ресурсами в такого роду проектахнесуть в собі також визначення потреби, чисельного і кваліфікаційного складу учасників проекту на всі етапи здійснення проекту та інше.

Також, в процесі роботи над проектом потрібно враховувати роботу надуправлінням часом, яка включає у себе наступне: постановка робіт та їхньої тривалості виконання, початкових та кінцевих термінів проекту, його частин, оптимізацію відповідних характеристик; прогнозування строків завершення виконання робіт, етапів і всього проекту; прийняття рішень для ліквідації попердньо неврахованих тимчасових відхилень від плану. Управління часом виконуєтьсязавдяки процесам тимчасового аналізу проекту та відповідних його частин, календарного проектування робіт, контролю процесу виконання робіт, перевірка їх актуальності та змін.

Під час виконання проекту загрозу становлять самечасові ресурси, які найчастіше бувають порушеними через некоректне планування людських ресурсів з урахуванням усіх обмежень, що призводить до відхилення відстроків виконання проекту та втрати матеріальних і фінансових ресурсів проекту. На допомогу плануванню ресурсів приходить теорія розкладів.

У цій роботі буде описано процес управління ресурсами, а саме ефективне планування реалізації проекту учасниками команди за мінімальний час роботи.Процес управління

проектом буде виконуватися за допомогою методології Scrum.

Аналіз існуючих рішень

Проблема ефективного управління проектами була опрацьована вченими багатьох країн. У роботі [2] автор пропонує власну систему для автоматизації процесу управління проектами, де проекти обов'язково потребують попереднього планування, організації, обговорення, спільної роботи учасників команди і багато іншого, при цьому особливу увагу приділяють контролю виконання та аналізу ризиків.

Робота[3] представляє собою приклад системи для автоматизації керування проектами, як розширення вже існуючої, дуже популярної системи JIRA. Команда розробників зробила фокус саме на аналітичні властивості системи на різних рівнях керування проектами.

Компанія Майкрософт запропонувала свою систему автоматизації керування проектами [4]. Серед основних особливостейякої можнавиділити наступні: можливість розподілу ресурсів та отримання побудованої моделівиконання проекту поетапно, яка враховує всі наявні ресурси, прогнозовані ризики, пріоритетність.

Метою роботи є мінімізація часу виконання проекту за рахунок створення ефективних розкладів виконання завдань учасниками проектної команди.

Постановка задачі

Отримано замовлення на виконання певного IT-проекту за певний проміжок часу. Проведено аналіз надісланого проекту та визначено необхідні ресурси (їх види та об'єми) і вимоги до його реалізації. Процес управління реалізацією проекту буде виконуватись за допомогою використання методології Scrum, гнучка методологія управління проектами, якає частиною сімейства Agile.

Згідно з результатамианалізуумов реалізації проекту було визначено множину завдань до виконання всього проекту, які необхідно реалізувати. Це є процесом формування беклогу продукту.Далі кожному завданню присвоюється пріоритет щодо її черговості виконання (іншими словами: пріоритет - першочерговість виконання завдання) та тривалість виконання цього завлання еталонним виконавцем (еталонний виконавець працює з коефіцієнтом продуктивності 1).

Існує класифікація пріоритетів виконання завдання: високий, середній та низький (завдання з високим пріоритетом виконуються першочергово, далі відповідно задачі з найнижчим пріоритетом виконуються в останню чергу, або можуть бути взагалі невиконаними або такими, які відкладено до реалізації в наступних етапах).

Для реалізації проекту та його завдань можуть бути долучені різні типи кваліфікованих фахівців, що будуть працювати надвідповідними напрямками реалізації проекту, їхня робота може бути пов'язаною (або ж не пов'язаною) між собою і, навіть, паралельно виконуватись. У даній роботі розглянемо ситуацію, коли виконавці працюють над одним і тим же напрямком та є взаємозамінними, а також є можливість працювати паралельно.Учасники команди можуть відрізнятися лише коефіцієнтами їхньої продуктивності, яка визначається рівнем знань та вмінь, які має кваліфікований фахівець.

Під час управління проектом за допомогою методології Scrum, та із сформованим беклогом продукту (множина завдань), виконується попереднє планування першого етапу роботи уреалізації проекту, ітерації визначеної та фіксованої довжини, яка називається спринтом.Тривалість такого спринта 2-4 тижні. Під час планування проектного спринта,за допомогою пріоритетів, з беклогу продукту виділяється підмножина завдань, які мають високий пріоритет. Кількість цих завдань має бути такою, щоб вона не перевищувалазначення продуктивності команди. Рисунок 1представляє процес виділення підмножини завдань з загальної множини. Ця підмножна завдань стає беклогомпершого спринта.

БЕКЛОГ ПРОДУКТУ		
Номер завдання	Пріоритет	Тривалість (години)
1	Високий	4
2	Високий	7
3	Низький	3
4	Високий	2
5	Середній	5
6	Середній	7
7	Високий	9
8	Низький	2
9	Середній	1
10	Низький	4
…		
80	Середній	8
Всього: 80 завдань		**200 годин**

БЕКЛОГ СПРИНТА		
Номер завдання	Пріоритет	Тривалість (години)
1	Високий	4
2	Високий	7
4	Високий	2
7	Високий	9
…		
Всього: 30 завдань		**90 годин**

Рис.1 – Виділення беклогу спринта з беклогу продукту

Коли починається етап формування беклогу наступного спринта, беклог продукту знову детально переглядається, можуть змінюватися пріоритети відповідних завдань, з'являтися нові завдання, змінюватися та ділитися на дрібніші. Як і на першому етапі, беклог для наступного спринта виділяється з урахуванням коефіцієнта продуктивності команди, який базується на оцінці успішності попереднього спринта.

Виконання беклогу спринта триває до директивного строку – часу, до якого повинні бути проведені і остаточно завершені всі завдання спринта.На рисунку 2 наведено схематичний приклад розподілення завдань між виконавцями.

БЕКЛОГ СПРИНТА		
Завдання	**Пріоритет**	**Години**
1	Високий	5
2	Високий	6
3	Високий	2
7	Високий	7
…	…	…
ВСЬОГО: 30 завдань		**90 годин**

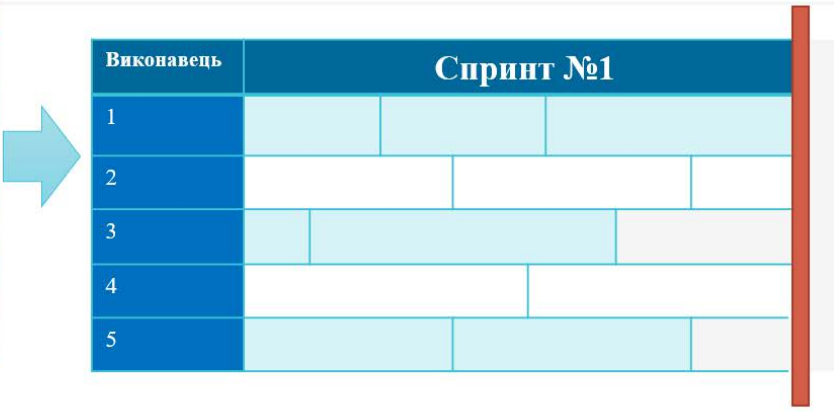


Рис.2 – Процес розподілення беклогу спринта на виконавців

Критерієм поставленої задачі є мінімізація часу виконання проекту та ефективне планування ресурсів, які є обмеженими.

Математична постановка задачі

Дано множину завдань $J = \{1,..., j,..., n\}$ та множину учасників команди (виконавців) $M = \{1,..., i,..., m\}$. Усі завдання поступаютьв нульовий момент часу,всі завдання маютьспільний директивний термін d, а процес обробки кожного завдання виконується без зупинки до завершення виконання завдання.Кожне завдання може бути призначене на виконання на будь-якого виконавця (учасника команди).Кожен виконавець має відповідний ~~свій~~ коефіцієнт продуктивності реалізації завдань - $v_i > 0$, для кожного завданнявідома еталонна тривалість виконання $t_j > 0$, $j = 1,..., n$ (тривалість виконання завдання «еталонним» учасником команди, коефіцієнт продуктивності якого дорівнює одиниці ($v_i = 1$)) , звідки випливає, що тривалість виконання j-ого завдання i-им виконавцем,

враховуючи коефіцієнт продуктивності виконавця становить:

$$t_{ij} = t_j v_i \qquad (1)$$

Нехай тоді C_j – момент завершення j-ого завдання, C_i - момент завершення виконання завдань i-им виконавцем, $C_{\max} = \max_j\{C_j\}$ – це максимальний момент завершення виконання завдань (момент завершення виконання останнього завдання), $Z_j = \max_j\{0, C_j - d\}$, – запізнення виконання завдання; $E_j = \max_j\{0, d - C_j\}$ – випередження виконання завдання; $E = \sum_{j=1}^{n} E_j$ – сумарне запізнення завдання; $Z = \sum_{j=1}^{n} Z_j$ – сумарне випередження завдання.

Критерій 1. Необхідно побудувати розклад реалізації завдань, при якому досягає мінімуму максимальний час завершення завдань:

$$C_{\max} \to \min \qquad (2)$$

Критерій 2. Необхідно побудувати розклад реалізації завдань, при якому завдання будуть розподілені між учасниками команди рівномірно, враховуючи їх коефіцієнт продуктивності:

$$\max_j\{E_j, Z_j\} \to \min \qquad (3)$$

Алгоритми розв'язання поставленої задачі

Для розв'язання задачі теорії розкладів, яка належить до класу NP та маючи побудовані різні критерії оптимальності, буде доречно використати евристичні алгоритми розв'язання задачі теорії розкладів. Важливо зазначити, що не можна гарантувати, що оптимальне рішення буде визначено, проте евристичні алгоритми помітно скорочують час розв'язання схожих задач, що забезпечує якнайшвидше отримання результату. Тому для розв'язання вищеописаної задачі, ми обрали наступні алгоритми: алгоритм локального пошуку[4] та жадібний алгоритм[5].

Жадібний алгоритм. Цей алгоритм простий для розуміння і реалізації, працює досить швидко, відомо багато задач різної складності, які можна розв'язати за допомогою жадібного алгоритму.

Для того, щоб побудувати план виконання завдань за критерієм 1, необхідно відсортувати завдання в порядку спадання їх тривалості виконання $t_1 > t_2 > ... > t_j > ... > t_n$. Далі виконуємо кроки зі схеми алгоритму нище.

Схема роботи жадібного алгоритму у кроках:

Крок 1. Відсортувати масив завдань у порядку спадання тривалості виконання завдань $t_1 > t_2 > ... > t_j > ... > t_n$.

Крок 2. Визначити i-ого виконавця, де $i = 1, 2, ..., m$, у якого момент завершення виконання завдань $C_i = C_{\min}$ і призначити на нього j-те завдання з найбільшою тривалістю t_1.

Крок 3. Повторити Крок 2, допоки nзавданьз множини $J = \{1, ..., j, ..., n\}$ не буде розподілено на mвиконавців.

Крок 4. Отримали розклад G

Алгоритм локального пошуку. Цей алгоритм зарекомендував себе як один з найкращих евристичних алгоритмів розв'язання задач, точні вирішення яких потребують експоненційних затрат часу.

Першим етапом побудови плану виконання завдань є побудова початкового розкладу за допомогою жадібного алгоритму.

Схема роботи алгоритму локального пошуку:

Крок 1. Відсортувати масив завдань у порядку спадання тривалості виконання завдань $t_1 > t_2 > ... > t_j > ... > t_n$.

Крок 2. Визначити i-ого виконавця, $i = \overline{1, m}$, у якого момент завершення виконання завдань $C_i = C_{\min}$ і призначити на нього j-те завдання з найбільшою тривалістю t_1.

Крок 3. Повторити Крок 2, допоки nзавдань з множини $J = \{1, ..., j, ..., n\}$ не буде розподілено на mвиконавців.

Крок 4. Отримано розклад G

Крок5.З розкладу G визначитизначення C_i між усіма виконавцями. З них обрати $C_{\max}$ та $C_{\min}$.

Крок 6. Повторитикрок 1, крок 2 та крок 3 для обраних виконавців з моментами

завершення виконання завдань $C_{\max}$ та $C_{\min}$ для побудови розкладу G'.

Крок 7.Внести отриманий розклад G' до загального початкового розкладу G на місце виконавців зі значеннями $C_{\max}$ та $C_{\min}$ – отримали розклад ітерації 1 - G_1.

Крок 8. Для розкладу G_1 повторити крок 5, крок 6 та крок 7. Отримаємо розклад G_2.

Крок 9.Повторити Крок 8, допоки не буде достягнуто умову припинення пошуку: коли отримали найкращий розклад G_q за всі попередні на ітерації q, $q = 1, 2, ..., l$, а наступні три ітерації $q+1, q+2, q+3$ звідповідними розкладами $G_{q+1}, G_{q+2}, G_{q+3}$ дали гірший (або такий же) результат, ніж G_q

Результати проведених досліджень

Проведено експериментальні дослідження для роботи двох алгоритмів:результати збіжності двох алгоритмів порівнюються на рисунку 3 для задач великої розмірності. Важливо відмітити, що алгоритм локального покушу (LS) плавно сходиться після 210 розмірності, тоді як жадібний(GA) сходиться після 250 та 270 відповідно.

На рисунку 4 представлено результати досліджень, що відповідають швидкості роботи алгоритмів. Експерименти показують, що алгоритм локального пошуку добре працює у створенні найкращих рішень, незважаючи на дещо довший час виконання.

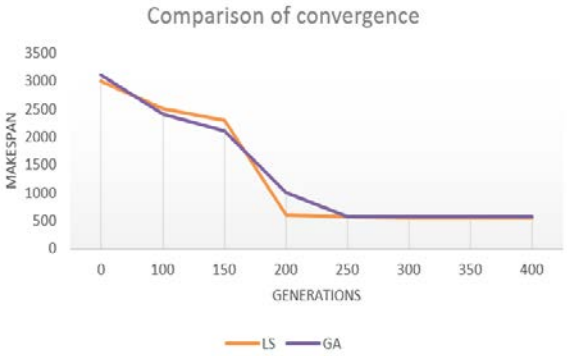


Рис. 3 – Результати збіжності двох алгоритмів

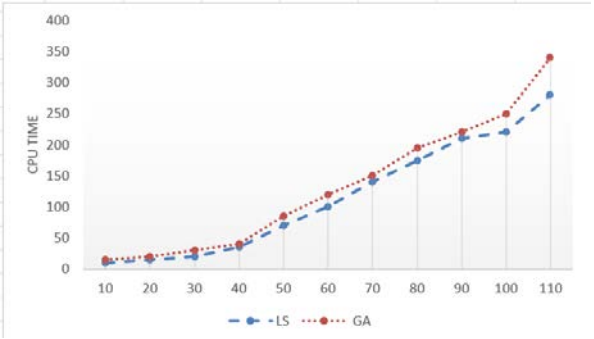


Рис. 4 - Порівняння витраченого середнього часу процесора для роботи GA та LS алгоритмів, за 20 повторень

Висновки

У роботі представлено та детально описано змістовну постановку задачі, яка є актуальною проблемою у IT і яка потребує розв'язання для майбутнього використання рішень у плануванні різних ресурсів проектів. Було наведено приклади робіт, що вже реалізовані, визначено їхні недоліки та потенційні покращення. Побудовано математичну модель задачі побудови розкладу реалізації завдань для виконавців проекту. Розглянутоевристичні алгоритми розв'язання: жадібний алгоритм та алгоритм локального пошуку. Розв'язання поставленої задачіобраними алгоритмами надало можливість отриматипровести експериментальні дослідження і виявити, що алгоритм локального пошуку здатний знаходити краще рішення за меншу кількість ітерацій, аніж жадібний алгоритм. Також було встановлено, що для задач великої розмірності, час роботи алгоритмів відрізняється: жадібний алгоритм працює швидше, але не гарантовано отримання найкращого результату розв'язання.Більш детально результати експериментальних досліджень будуть розглянуті у наступних роботах.

Перелік посилань

1. Чорна М. В. Проектний аналіз. // Харків: Консум, 2003. – 228 с.
2. Автоматизация планирования проектов с помощью параметрического моделирования в системе T-FLEX CAD [Електронний ресурс] – Режим доступу до ресурсу:https://sapr.ru/article/14919
3. The 4 Phases of Project Management – and How Process Automation Can Help [Електронний ресурс] – Режим доступу до ресурсу: https://www.nintex.com/blog/the-4-phases-of-project-management-and-how-process-automation-can-help/

4. Методология внедрения Microsoft Project Server/Online [Електронний ресурс] – Режим доступу до ресурсу: http://leoconsulting.com.ua/library/articles-about-pm/291-metodologiya-vnedreniya-microsoft-project-server-online
5. Метод підйому (локального пошуку) [Електронний ресурс] – Режим доступу до ресурсу: https://studfile.net/preview/3766334/page:6/
6. Жадібні алгоритми [Електронний ресурс] – Режим доступу до ресурсу: https://dl.sumdu.edu.ua/textbooks/95351/522264/index.html

УДК519.1,519.686

КЛИМЕНКО Д.С.
КЛИМЕНКО О.М.

ВИКОРИСТАННЯМЕТОДІВ ДИСКРЕТНОЇ МАТЕМАТИКИ ДЛЯ РОЗРОБКИ ТЕСТ-КЕЙСІВ З МЕТОЮ ПОВНОГО ПОКРИТТЯРОЗГАЛУЖЕННОЇ БІЗНЕСЛОГІКИЗІ СКЛАДНИМИ УМОВАМИ

Робота присвячена розробці алгоритмів для визначення кількості тест-кейсів для повного покриття бізнес-логіки з великою кількість умов та розгалужень. У роботірозглядається підхід до дизайну тест-кейсів, використовуючи методи дискретної математики.

КЛЮЧОВІ СЛОВА: дискретна математика, теорія графів, повні бінарні дерева, закінчені бінарні дерева, класи еквівалентності, техніки тест дизайну, таблиці розв’язків, покриття рішень, покриття гілок.

The work is devoted to the development of algorithms for determining the number of test cases to fully cover business logic with numbers of conditions and branches. The work considers the approach to the design of test cases using the methods of discrete mathematics.

KEYWORDS: discrete mathematics, graph theory, complete binary trees, complete binary trees, test design techniques, solution tables, solution coverage, branch coverage.

1. Вступ

Якість програмного забезпечення завжди було одним з найважливіших питань для користувачів і для розробників. Тестування програмного забезпечення - перевірка відповідності між реальним і очікуваним поведінкою програми, що здійснюється на кінцевому наборі тестів, обраному певним чином. У більш широкому сенсі, тестування - це одна з технік контролю якості, що включає в себе активності з планування робіт (Test Management), проектування тестів (Test Design), виконання тестування (Test Execution) і аналізу отриманих результатів (Test Analysis).

2. Тест-дизайн

Тест-дизайн - це розробка, створення тестів. Кожен тест спрямований на перевірку конкретного припущення. Наприклад: «Що буде, якщо користувач помилково клацне тут не лівою, а правою кнопкою миші».

Тестувальник (QA) моделює набір тестових випадків (тест-кейсів), щоб перевірити, як додаток поводиться в різних умовах. Завдання фахівця - знайти баланс і виявити максимальну кількість помилок при необхідному мінімумі тестових сценаріїв. При цьому потрібно перевірити всі найбільш важливі кейси, оскільки час тестування обмежена.

Повне тестування (Exhaustive Testing - ET) - це крайній випадок. В межах цієї техніки необхідно перевірити всі можливі комбінації вхідних значень, і в принципі, це повинно знайти всі проблеми. На практиці застосування цього методу є досить складним, через величезну кількості вхідних значень.Але є такі сфери розробки програмного забезпечення, наприклад, медицина, де це зробити вкрай важливо. Наразі будемо розглядати смаме такий варіант, коли треба провести повне тестування.

3. Таблиці прийняття рішень

У покритті множинних умов для кожного рішення слід оцінювати всі комбінації умов.

Візьмемо наприклад такий код:

if (A||B)thenprint C

Тут ми маємо 2 булевих вирази A і B, тому набір тестів для покриття з кількома умовами буде таким:

тест-кейс1: A = TRUE, B = TRUE

тест-кейс 2: A = TRUE, B = FALSE

тест-кейс 3: A = FALSE, B = TRUE

тест-кейс 4: A = FALSE, B = FALSE

Тобто, існує 4 тести для 2 умов. Подібним чином буде 8 тестів для 3 умов.Отже, можемо сказати, що якщо є n умов, буде 2^n тестів.

4. Аналіз граничних значень

Аналіз граничних значень (Boundary Value Analysis - BVA) - це метод проектування тестів, коли тест-кейси розробляються з використанням граничних значень, і використовується для перевірки діапазону.

Наприклад: магазин у місті пропонує різні знижки залежно від вартості покупок, здійснених особою. Для того, щоб протестувати програмне забезпечення, яке обчислює знижки, ми можемо визначити діапазони значень покупок, які приносять різні знижки. Якщо покупка знаходиться в діапазоні від $ 1 до $ 50, не має знижок, покупка від $ 50 до $ 200 має знижку 5%, а покупка від $ 201 і вище має знижку 10%.

Можна ідентифікувати 3 дійсних розділи еквівалентності та 1 недійсний розділ, як показано нижче:

недійсний розділ$0;

дійсний розділ (без знижок)$1–50;

дійсний розділ (5%)$51–200;

дійсний розділ (10%)$201 –вище.

З цієї таблиці можемо визначити граничні значення кожного розділу: для недійсного розділу 0; для дійсного розділу (без знижок): 1, 50; для дійсного розділу (знижка 5%): 51, 200; для дійсного розділу (знижка 10%): 201, максимально дозволене число в програмному додатку.

Тобто маючи mрозділів еквівалентності (або класів еквівалентності) отримуємо не менше 2m-1 граничних значень, тобто тест-кейсів.

5. Покриття гілок

Як правило бізнес логіка програми складається з послідовних розгалужень, тобто має вигляд повного бінарного дерева або закінченого бінарного дерева. Наприклад:

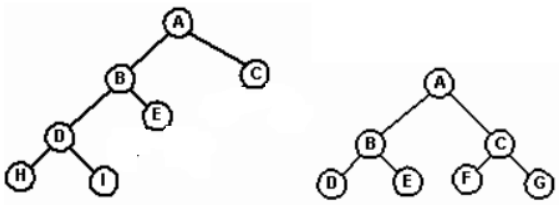


Рис. 1. Закінчене бінарне дерево (глубини 3) та повне бінарне дерево (глубини 2)

Якщо k – це глубина дерева, то кількість тест-кейсів буде дорівнювати кількостілистків, а саме відk+1 (у випадку мінімального закінченного бінарного дерева) до 2^k (у випадку повного бінарного дерева)за умовою, що кожна внутрішня вершина, це проста умова, тобто умова без логіних операцій.

6. Об’єднання вище зазначених методів

Об’єднавши методи тест дизайну, що були наведені вище, можна отримати кількість тест-кейсів, що необхідно розробити для повного покриття тестами бізнес логіки програмного додатку.

Висновок

В ході даної роботи був розглянутий підхід до розробки алгоритма для визначення кількості тест-кейсів для повного покриття бізнес-логіки з великою кількість умов та розгалуженьвикористовуючи методи Дискретної математики.

Література

1. SoftwareTestingMentor.Boundaryvalueanalysis [Електроннийресурс] – Режимдоступудо ресурсу: https://www.softwaretestingmentor.com/boundary-value-analysis/

2. Tutorialspoint[Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/software_testing_dictionary/branch_testing.htm

3. Нікольський Ю.В., Пасічник В.В., Щербина Ю. М. Дискретна математика. – К.: Видавнича група BHV, 2007.