e-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2021. № 1
p-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2021. № 1

УДК 004.8

РОЗРОБКА МЕТОДУ ІДЕНТИФІКАЦІЇ СТАНУ КОМП'ЮТЕРНОЇ СИСТЕМИ НА ОСНОВІ АЛГОРИТМУ «ISOLATION FOREST»

Гавриленко С. Ю. – д-р техн. наук, доцент, професор кафедри «Обчислювальна техніка та програмування», Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.

Шевердін І. В. – аспірант кафедри «Обчислювальна техніка та програмування», Національний технічний університет «Харківський політехнічний інститут», Харків, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто задачу ідентифікації стану комп'ютерної системи. Об'єктом дослідження є процес ідентифікації стану комп'ютерної системи. Предметом дослідження є методи та засоби ідентифікації стану комп'ютерної системи.

Мета. Метою роботи є розробка методу ідентифікації стану комп'ютерної системи.

Метод. Розроблено метод ідентифікації стану комп'ютерної системи на основі комплексного використання процедури групування нерозмічених вихідних даних та технології машинного навчання на основі алгоритму «Isolation Forest», який надає можливість ідентифікувати стан комп'ютерної системи і виділити назву процесу, який спричинив аномальний стан. Для цього запропоновано процедуру та розроблено програмний додаток для збору статистичних даних у вигляді подій функціонування операційної системи та виконано їх аналіз. Отримано, що найбільш інформативними є операції читання та запису. Для формування єдиного датасету, операції читання та запису зіставлено з назвою процесу та об'єднано в один масив груп подій, що надалі дозволяє виділити процес, який спричиняє аномальний стан комп'ютерної системи. За результатами дослідження, у якості складової методу ідентифікації стану комп'ютерної системи використано ансамблевий алгоритм «Isolation Forest». Проведено оцінку точності та оперативності розробленого методу ідентифікації стану комп'ютерної системи .

Результати. Розроблений метод реалізований програмно і досліджений під час розв'язання задачі ідентифікації аномалій функціонування комп'ютерної системи.

Висновки. Проведені експерименти підтвердили працездатність запропонованого методу, що надає можливість рекомендувати його для практичного використання з метою підвищення оперативності ідентифікації стану комп'ютерної системи та використання його у якості експрес-методу. Перспективи подальших досліджень можуть полягати в розробці ансамблю нечітких дерев рішень на основі запропонованого методу, оптимізації його програмних реалізації.

КЛЮЧОВІ СЛОВА: комп'ютерна система, події операційної системи, аномальний стан, ідентифікація, машинне навчання, алгоритм «Isolation Forest».

АББРЕВІАТУРИ

КС – комп'ютерна система;

ОС – операційна система;

NB – метод Байєса (Naive Bayes);

KNN – метод *k* найближчих сусідів (k Nearest Neighbors);

DT – метод дерев рішень (Decision Trees);

SVM – метод опорних векторів (Support Vector Machine);

RF – метод випадкового лісу (Random Forest);

IF – метод ізолюючого лісу (Isolation Forest);

J48 – алгоритм C4.5, реалізований на мові програмування Java;

iTree – ізолююче дерево ухвалення рішень.

НОМЕНКЛАТУРА

X – вихідні дані (події ОС);

x_{ij} – об'єкт події ОС із набору вихідних даних;

m– кількість показників об'єкту;

k – кількість класів;

C – множина класів;

f – алгоритм класифікації з областю визначення X та областю значення C;

AS – показник аномальності (Anomaly Score).

K – статистичне ядро, симетрична, але не обов'язково додатна функція з інтегралом рівним одиниці;

h – параметр згладжування, $h > 0$;

X – вихідні дані у вигляді нерозміченого масиву груп подій;

$\hat{f}_h(x)$ – функція щільності розподілу імовірності випадкової величини;

$H(i)$ – гармонічне число;

γ = 0,5772156649 (константа Ейлера);

$h(x)$ – глибина гілки, що містить це спостереження, що еквівалентно кількості розщеплень, необхідних для ізоляції цієї точки;

$E(h(x))$ – середнє значення $h(x)$ з набору Isolation Tree;

$c(n)$ – середнє значення $h(x)$ n-спостережень.

ВСТУП

При вирішенні завдань, пов'язаних з діагностикою та захистом комп'ютерних інформаційних ресурсів, центральною є задача оперативного виявлення аномальної поведінки комп'ютерної системи в умовах зовнішніх впливів.

Проведені дослідження існуючих комп'ютеризованих систем ідентифікації станів дозволили ви-

DOI 10.15588/1607-3274-2021-1-11

явити ряд обмежень їх використання. Так при появі аномалій, породжених вторгненнями в КС з невстановленими або нечітко визначеними властивостями, сучасні методи не завжди залишаються ефективними і вимагають тривалих часових та програмно-апаратних ресурсів для їх відповідної адаптації, що призводить до зниження показників оперативності ідентифікації стану КС.

На сьогодні комп'ютерна система характеризується великим обсягом показників її функціонування. Це призводить до наявності труднощів з адекватного відбору показників для ідентифікації стану КС в умовах зовнішніх впливів і розробки критерію оцінки, що відповідає обраним показникам.

Об'єктом дослідження є процес ідентифікації стану комп'ютерної системи.

Предметом дослідження є методи та засоби ідентифікації стану КС.

Існує безліч методів ідентифікації, які використовують різний математичний апарат і різні підходи при реалізації [1–3]. Одним із найбільш поширених методів аналізу великих обсягів даних (data mining) є методи машинного навчання (machine learning). Однак, ефективність цих методів залежить від конкретної розв'язуваної задачі.

Метою роботи розробка методу ідентифікації стану комп'ютерної системи.

1 ПОСТАНОВКА ЗАДАЧІ

Будемо вважати, що функціонування комп'ютерної системи є сукупністю подій операційної системи. Виконати аналіз подій операційної системи та сформувати вихідні дані X у вигляді нерозміченого масиву груп подій, зіставлених з назвою процесу, тобто кожен об'єкт $x_i \in X$ задати у вигляді деякого вектору $C = \{c_{i1}, c_{i2}, ..., c_{im}\}$. Тоді постановка завдання ідентифікації стану комп'ютерної системи визначається наступним чином. Нехай $C = \{c_1, c_2, ..., c_k\}$ – кінцева множина класів. Існує невідоме відображення f: $X \to C$, причому його значення відомі тільки на елементах кінцевої сукупності $T = \{(x_1, c_1), ..., (x_n, c_k) \subset X \times C$. Потрібно побудувати алгоритм f: $X \to C$, здатний класифікувати довільний стан КС $x_i \in X$. Для цього, для кожного елементу кінцевої сукупності (x_i, c_i) визначити критерій класифікації у вигляді функції *AS*. Визначити поріг бінаризації. Поріг бінаризації задається якщо відома приблизна частка аномалій в даних (для цього вибирається відповідний квантиль) або розраховується за умови щоб дисперсія між класами була мінімальною, $D \to \min$.

Використовуючи значення функції *AS* та значення порогу, детектувати наявність аномалій: в заданій множині X для кожного елемента $x_i \in X$ видати 0, якщо цей об'єкт відноситься до класу нормальних даних, і 1, якщо цей об'єкт є аномальним. За наявності аномального стану, визначити процес, що його спричинив.

2 ОГЛЯД ЛІТЕРАТУРИ

Функціонування КС характеризується великою кількістю процесів. Для аналізу цих даних і їх класифікації використовуються складні математичні алгоритми, що базуються на машинних методах навчання. Найбільш популярні алгоритми машинного навчання наведено в [4,5]. Так, прикладом імовірнісного методу класифікації є метод Байєса [6]. Перевагою методу є: висока швидкість роботи, легка інтерпретація результатів роботи алгоритму, проста реалізація алгоритму у вигляді програми. Незважаючи на наведені переваги, метод Байеса має не достатню точність класифікації і нездатний враховувати залежність результату класифікації від поєднання ознак.

Метод k найближчих сусідів відноситься до метричних методів і вважається найпростішим класифікатором [4, 7]. Перевагою даного методу є проста реалізація, наявність гарної теоретичної бази, адаптація під потрібне завдання вибором метрики або ядра. До недоліків відносяться: недостатня продуктивність в реальних завданнях, так як число сусідів, які використовуються для класифікації, буде досить великим; труднощі в наборі відповідних ваг і визначенням, які ознаки необхідні для класифікації; залежність від обраної метрики відстані між об'єктами.

Одним із найкращих методів класифікації є метод опорних векторів [8]. Недоліки методу опорних векторів полягають в наступному: неможливість калібрування ймовірності попадання в певний клас, підходить тільки для вирішення завдань з 2 класами, параметри моделі складно інтерпретувати.

Нейронні мережі також активно використовуються у зв'язку з появою великих обсягів даних і великих обчислювальних можливостей [9]. Їх ефективність досить висока, тому що вони генерують фактично велике число регресійних моделей (які використовуються в рішенні задач класифікації статистичними методами). Однак, будь-який метод, заснований на нейронних мережах, ніколи не дасть класифікатор потрібної якості, якщо набір навчальної вибірки не буде достатньо повним для того завдання, з якою доведеться працювати в системі.

Метод дерев рішень відноситься до логічних методів класифікації [10]. Головною перевагою методу є висока продуктивність навчання і прогнозування, такі дерева рішень можна легко візуалізувати і інтерпретувати. Недоліком методу є відносно невисока точність прогнозів, так як побудова класифікатору істотно залежать від вхідних параметрів [11]; структури даних, природи їх виникнення [12]. За умови відсутності розмітки даних, має місце проблема проведення відбору моделей та перевірки якості їх роботи (за допомогою кроссвалідації або тестування на відкладеній вибірці [13]. Для подолання вищенаведених недоліків розроблено методи, засновані на використанні ансамблів з декількох класифікаторів (сотень і навіть ти-

DOI 10.15588/1607-3274-2021-1-11

сяч). Ансамблі покращують якість, знижують залежність моделей від досліджених даних та вхідних параметрів, підвищуючи стабільність результатів. За якістю одержуваних прогнозів, ансамблі з декількох моделей часто перевершують інші методи [14–19].

Складовими ансамблевих алгоритмів можуть бути класифікатори з учителем та без учителя. Класифікатори без учителя є більш оперативними, оскільки не потребують навчання. Такі методи не потребують розмічених даних і намагаються самостійно знайти шаблони безпосередньо з вихідних даних. При цьому в більшості практичних додатків розмітка нормальних та аномальних класів даних відсутня, у зв'язку з чим проблема виявлення аномалій розглядається як завдання навчання без учителя [15, 16]. Використання дерев рішень у якості базових класифікаторів ансамблів рішень дозволяє, автоматично виконати відбір інформативних предикатів з урахуванням можливості взаємодії між ними.

Таким чином, проведені дослідження надали можливість виявити ряд обмежень використання існуючих методів, що у сукупності з наявністю різних типів даних, що характеризують стан функціонування комп'ютерної системи, призводить до суттєвої розбіжності якості та слабкої практичної придатності окремих класифікаторів. Крім того відомі методи виконують тільки ідентифікацію стану КС та не визначають процес, який спричинив аномальний стан.

У зв'язку з цим особливої актуальності набувають питання удосконалення та розробки нових методів ідентифікації стану КС.

3 МАТЕРІАЛИ ТА МЕТОДИ

Відповідно до постановки задачі, в рамках даного дослідження розроблено метод ідентифікації стану КС, який відрізняється від відомих методів використанням у якості класифікатора ансамблю iTree та наявністю процедури ідентифікації процесу, що спричинив цей аномальний стан.

Для формування вихідних даних виконано аналіз подій операційної системи Windows 10, а саме: ім'я процесу, вид операції, шлях до файлу виконання (табл. 1).

Аналіз подій ОС показав, що всі процеси ОС взаємодії з апаратною пам'яттю, так чи інакше можливо звести до операцій читання та запису. До операції запису слід також віднести операцію видалення чи створення, файлу або ключа реєстру.

Назва процесу, операції читання та запису і шлях до файлу виконання є важливим індикатором роботи ОС та надає можливість визначити, який процес і скільки раз ініціював операції та над якими ключами чи файлами він виконував дію.

Отримано, що статистика операцій читання та запису є закономірною для окремих процесів та характеризує стан комп'ютерної системи. Як правило, системні події мають відносно невелику кількість операцій запису та велику кількість операцій читання. Окремі вірусні події характеризуються великою кількістю операцій запису (в тому числі, запису системних конфігурацій налаштування ОС), які при нормальному стані функціонування ОС виконуються зрідка.

У якості вихідних даних було виділено найбільш розповсюджені операції читання файлів та ключів реєстру (RegOpenKey, RegQueryValue, RegQueryKey, ReadFile, QueryDirectory, RegEnumKey, QueryBasicInformationFile, QueryStandardInformationFile, RegEnumValue, QueryNameInformationFile, QuerySecurityFile, RegCreateKey, FileSystemControl, QueryRemoteProtocolInformation, QueryNetworkOpenInformationFile, RegQueryKeySecurity, QueryAllInformationFile, QueryAttributeTagFile, QueryNormalizedNameInformationFile, QueryEAFile, QueryIdInformation, NotifyChangeDirectory, QueryPositionInformationFile, QueryStreamInformationFile, RegQueryMultipleValueKey, QueryEaInformationFile, QueryFileInternalInformationFile, QueryLinks та операції запису (WriteFile, RegSetValue, CreateFile, SetEndOfFileInformationFile, RegCreateKey, FileSystemControl, RegDeleteValue, SetDispositionInformationFile, SetRenameInformationFile, SetAllocationInformationFile, RegDeleteKey, SetBasicInformationFile, SetPositionInformationFile, SetSecurityFile, SetValidDataLengthInformationFile, SetLinkInformationFile, SetEAFile).

Так як присутня сукупність параметрів, то для формування вхідного датасету запропоновано процедуру, яка базується на багатофакторному групуванні даних (рис. 1). Для комбінації двох категорій у один датасет, об'єднано операції читання та запису в один масив груп подій, та зіставлено з назвою процесу.

На рис. 1 наведено приклад результату роботи програмного додатку групування вихідних даних за назвою процесу, за шляхом до файлів операційної системи, операціями запису та читання, котрі були описані вище. У результаті отримано масив даних, який містить назву процесу, шлях до файлу виконання чи ключа реєстру, назву функції та інформацію про кількість операцій читання або запису для неї.

Таблиця 1 – Події операційної системи Windows 10

Назва атрибуту	Тип даних	Опис події	Приклад події
ProcessName	Строкове значення	Назва процесу	Explorer.exe …
Operation	Строкове значення (Категорія)	Вид операції	RegCloseKey; ReadFile; RegOpenKey …
Path	Строкове значення	Шлях до файлу виконання або ключ реєстру	C:\Windows\System32\NgcCtnrSvc.dll; HKCU\Software\Classes …

DOI 10.15588/1607-3274-2021-1-11

	ProcessName	ImagePath	Operation_x	ReadCount	Operation_y	WriteCount
0	Explorer.EXE	C:\Windows\Explorer.EXE	RegOpenKey	48	RegCreateKey	4
1	Explorer.EXE	C:\Windows\Explorer.EXE	RegOpenKey	48	RegSetValue	4
2	Explorer.EXE	C:\Windows\Explorer.EXE	RegQueryKey	4	RegCreateKey	4
3	Explorer.EXE	C:\Windows\Explorer.EXE	RegQueryKey	4	RegSetValue	4
4	Explorer.EXE	C:\Windows\Explorer.EXE	RegQueryValue	37	RegCreateKey	4
5	Explorer.EXE	C:\Windows\Explorer.EXE	RegQueryValue	37	RegSetValue	4
6	MsMpEng.exe	C:\ProgramData\Microsoft\Windows Defender\plat...	FileSystemControl	2	WriteFile	15
7	MsMpEng.exe	C:\ProgramData\Microsoft\Windows Defender\plat...	QueryAllInformationFile	4	WriteFile	15
8	MsMpEng.exe	C:\ProgramData\Microsoft\Windows Defender\plat...	QueryNameInformationFile	6	WriteFile	15
9	SearchFilterHost.exe	C:\Windows\system32\SearchFilterHost.exe	QueryNameInformationFile	3	0	0
10	SearchFilterHost.exe	C:\Windows\system32\SearchFilterHost.exe	RegOpenKey	3	0	0

Рисунок 1 – Приклад результату групування

Аналіз результату групування дозволив отримати статистику масиву груп подій. Статистика результату включає кількість операцій читання і запису (count), середнє значення (mean), середнє квадратичне відхилення (std), мінімальне (min) та максимальне (max) значення, а також значення для 25-го (25%), 50-го (50%), 75-го (75%) перцентилів (рис. 2), які надалі можуть бути використані для аналізу даних, що надходять до класифікатору та вибору типу класифікатору.

Для оцінки отриманого масиву груп подій на предмет наявності аномалій було використано непараметричний метод оцінки функції щільності імовірності випадкової величини за вибіркою, а саме метод ядрової оцінки щільності розподілу:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i).$$

Результати аналізу розподілу даних у вигляді графіку розподілу наведено на (рис. 3). Як видно із графіків для операцій читання і запису, розподіли даних в обох випадках далекі від нормального закону розподілу. Операції читання та запису мають позитивний довгий хвіст, основна частина розподілу зосереджена зліва. Хвостовий розподіл набагато перевищує піки справа, що потенційно може сигналізувати про наявність аномальних викидів. Присутність невеликих хвостів є індикатором аномальної зміни стану окремих процесів.

	ReadCount	WriteCount
count	273.000000	44.000000
mean	1530.347985	21.681818
std	6167.488208	20.658657
min	1.000000	3.000000
25%	9.000000	8.000000
50%	55.000000	15.000000
75%	415.000000	17.000000
max	34847.000000	61.000000

Рисунок 2 – Статистика розподілу операцій читання і запису

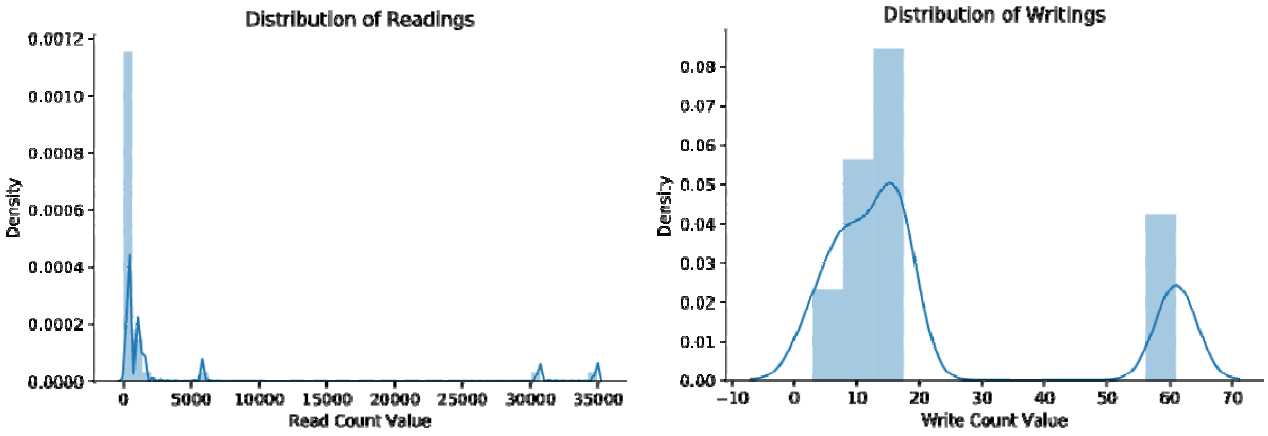

Рисунок 3 – Графіки розподілу операцій читання та запису

DOI 10.15588/1607-3274-2021-1-11

Отримані результати розподілу вихідних даних показали, що вони можуть бути використаними у якості вихідних даних методу ідентифікації аномалій функціонування комп'ютерної системи.

У якості складової методу ідентифікації стану КС використано алгоритм на основі ізолюючого лісу. IF – це ансамблевий алгоритм без учителя, який є варіацією ідеї випадкового лісу [17–19] та замість спроби побудувати модель звичайних екземплярів, ізолює аномальні точки в наборі даних.

IF є ансамблем iTree, де секції дерев створюються шляхом першого випадкового вибору об'єкту, а потім вибору випадкового значення поділу між мінімальним і максимальним значеннями обраного об'єкта.

Мірою нормальності спостереження за даним деревом є глибина гілки $h(x)$, що містить це спостереження, що еквівалентно кількості розщеплень, необхідних для ізоляції цієї точки. Аномальні значення потрапляють в листя на невеликій глибині дерева і тим самим детектуються. Показником аномальності є функція *AS*, яка для даного об'єкту видає деякий «рейтинг» аномальності.

Оцінка аномалії *AS* екземпляра *x* визначається як [19]:

$$AS(x,n) = 2^{-\frac{E(h(x))}{c(n)}},$$

де $c(n)$ визначається наступним чином [20]:

$$c(n) = 2H(n-1) - (2(n-1)/n),$$

де $H(\mathrm{i})$ визначається як [20]:

$$H(i) = \ln(i) + \gamma.$$

Чим більше значення аномалії *AS*, тим більше вірогідність того, що досліджуваний об'єкт є аномальним.

Відповідно до алгоритму IF будується необхідне число дерев та проводиться класифікація об'єкту. Об'єкт класифікації буде віднесено кожним деревом до одного з двох класів: нормального чи аномального. Прийняття рішення відносно класу об'єкту виконується методом простого голосування, тобто на основі мета-алгоритму беггінгу

Перевагою цього методу є можливість якісної обробки, як безперервних, так і дискретних даних з великим числом ознак і класів, в тому числі з пропущеними значеннями ознак. Складність ізолюючого дерева – $O(n \log n)$, що ефективніше більшості інших алгоритмів. Метод не потребує істотних затрат пам'яті, на відміну від, наприклад, метричних методів, які часто потребують побудови матриці попарних відстаней, стійкий до прокляття розмірності.

Для порівняння, у якості складової методу ідентифікації стану КС з метою виявлення аномалій також було досліджено алгоритм KNN. KNN. – це простий непараметричний алгоритм, де для класифікації використовуються відстані (зазвичай евклідові), порахована до усіх інших об'єктів [4, 7].

4 ЕКСПЕРИМЕНТИ

Результати використання алгоритму IF для ідентифікації стану КС наведено на рис. 4–7. Рис. 4–5 відображають залежність *AS* від значення кількості операцій читання або запису (Readings or Writings) для стану КС у режимі простою без запуску будь яких процесів, який можливо трактувати як нормальний стан. Як видно із рис 4–5 за результатами ідентифікації аномального стану функціонування КС не виявлено.

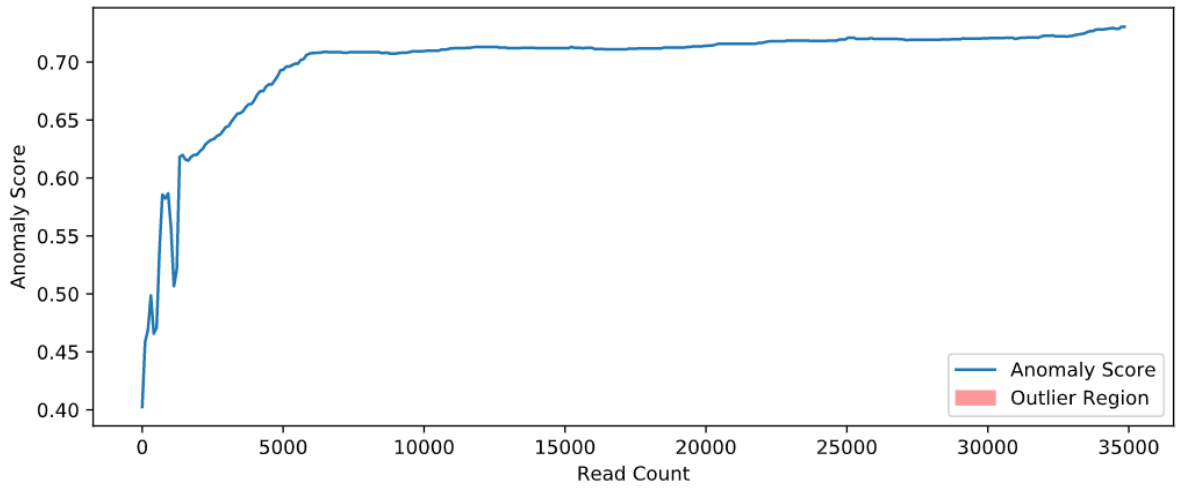

Рисунок 4 – Результат ідентифікації нормального стану КС алгоритмом IF для операції читання

DOI 10.15588/1607-3274-2021-1-11

На рис. 6–7 наведено результати ідентифікації стану функціонування КС за умови запуску великої кількості процесів, в тому числі шкідливого програмного забезпечення, що призводить до аномального стану функціонування КС. Графіки відображають залежність *AS* від значення кількості операції читання, які надалі зіставляються з назвою процесу, що дозволяє визначити ініціюючий події процес з масиву груп подій. На рис. 6 виділені дві світлі області для групи подій зі значенням кількості операцій читання (Readings) 2700–3400, які відповідають піку оцінки *AS*.

На рис. 7 піком оцінки *AS* є кількість операцій запису (Writings), яка приймає значення 50–60. За звичай, це групи подій одного аномального ініціюючого процесу.

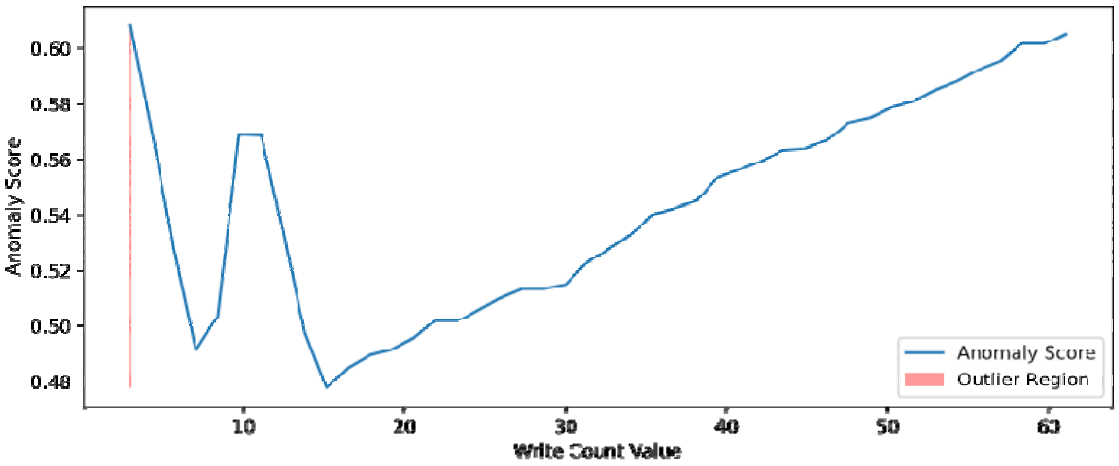

Рисунок 5 – Результат ідентифікації нормального стану КС алгоритмом IF для операції запису

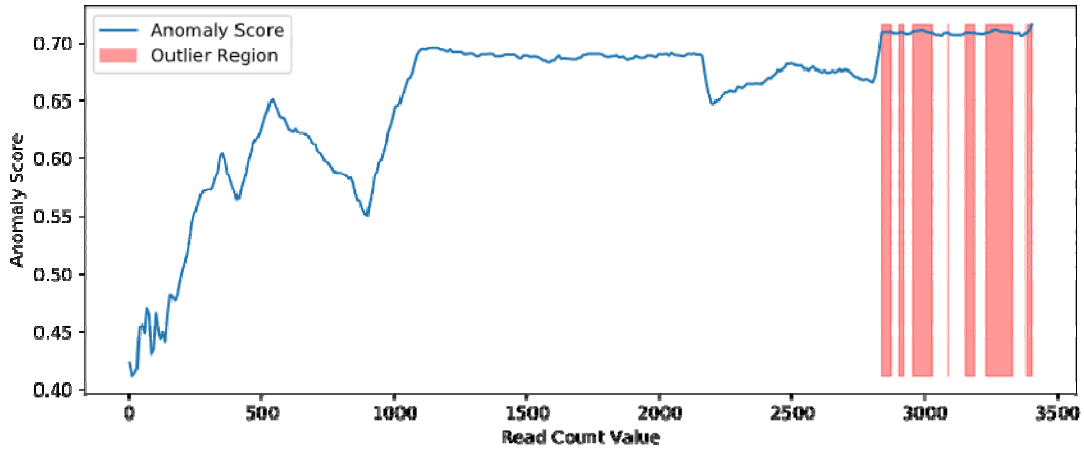

Рисунок 6 – Результат ідентифікації аномального стану КС алгоритмом IF для операції читання

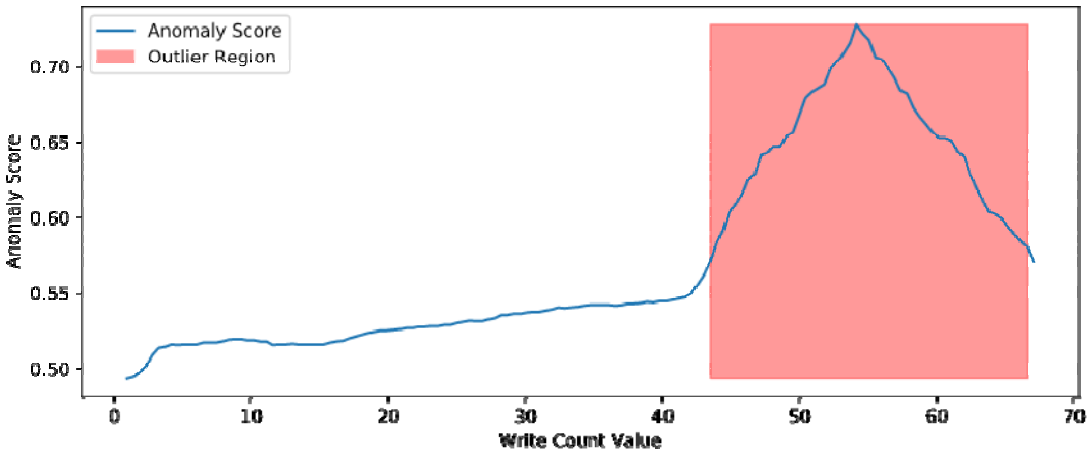

Рисунок 7 – Результат ідентифікації аномального стану КС алгоритмом IF для операції запису

DOI 10.15588/1607-3274-2021-1-11

Приклад результату роботи процедури ідентифікації процесу, який є причиною аномального стану наведено на (рис. 8). Результат містить назву аномального процесу (ProcessName), шлях до файлу виконання (ImagePath), тип операції читання (Operation_x), кількість операцій читання (ReadCount), тип операції запису (Operation_y), кількість операцій запису (WriteCount).

```
ProcessName                              7zG.exe
ImagePath          C:\Program Files\7-Zip\7zG.exe
Operation_x             QueryBasicInformationFile
ReadCount                                      17
Operation_y                             WriteFile
WriteCount                                     66
```

Рисунок 8 – Приклад результату ідентифікації

Подальші дослідження пов'язані з використанням багатовимірного аналізу системних подій, а саме одночасним використанням двох параметрів: читання і запису (змінні ReadCount і WriteCount), що дозволяє збільшити точність ідентифікації стану КС.

Для порівняння, у якості методів ідентифікації стану КС з метою виявлення аномалій було використано два алгоритми: IF та KNN.

На рис. 9–10 наведено результати ідентифікації стану КС у режимі очікування. Функціонування КС у такому режимі можливо зіставити з нормальним станом функціонування. Обидва алгоритми IF та KNN зафіксували кількість аномалій на рівні статистичної похибки (5%). Це означає, що системні події схожі між собою за параметрами запису та читання.

Для моделювання аномального стану КС виконано запуск великої кількості процесів, в тому числі архівування файлів додатком «7Zip» (процес «7zG.exe») та шкідливого програмного забезпечення (рис. 11, 12).

Обидва алгоритми IF та KNN виділили аномальні процеси. Однак, метод на основі алгоритму KNN показав меншу точність класифікації. Як видно з рис. 11 велика кількість аномальних подій, які виділено крапками, не належать виділеній аномальній області (outliers). На противагу, точність методу на основі алгоритму IF є набагато більшою (рис. 12). Метод виділив область не аномальних подій (inliers) та більш точно ідентифікував ділянки аномальних подій (outliers). Порівняльний аналіз ідентифікації стану КС, за умови запуску різних системних процесів, у вигляді проценту виявлених аномальних подій алгоритмами KNN та IF наведено в (табл. 2). Як видно із таблиці, алгоритм IF виявив більшу кількість аномалій для усіх процесів. Такі результати співпадають з результатами ідентифікації стану КС методом на основі ансамблю дерев рішень, побудованих за алгоритмом J48 [20]. Таким чином, можливо зробити висновок, що алгоритм IF є більш якісним та може бути використаним у якості складової методу ідентифікації стану комп'ютерної системи.

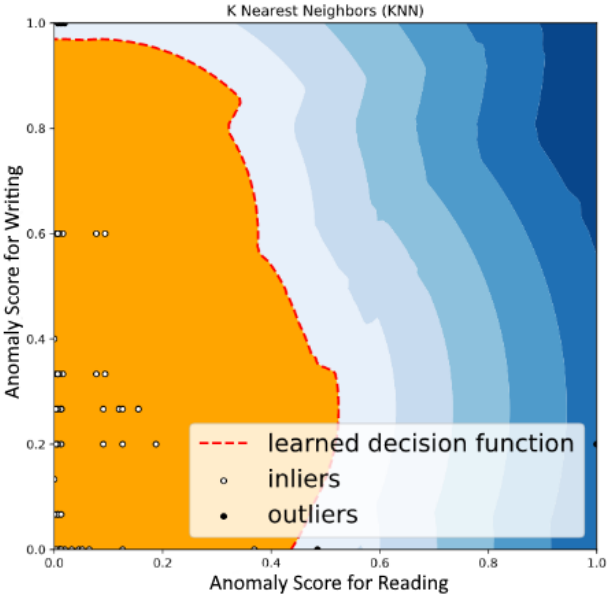

Рисунок 9 – Результати багатовимірного пошуку аномалій для системних подій методом KNN

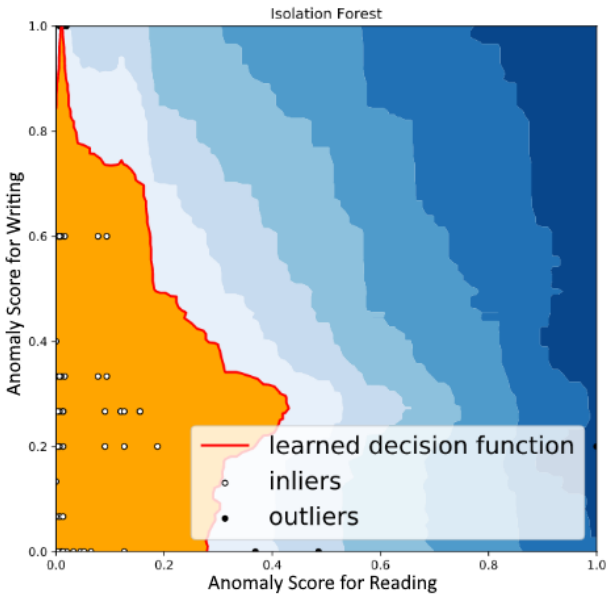

Рисунок 10 – Результати багатовимірного пошуку аномалій для системних подій методом IF

DOI 10.15588/1607-3274-2021-1-11

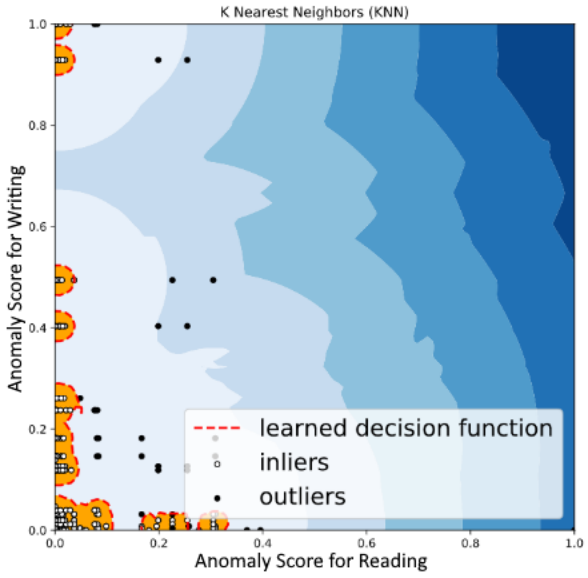

Рисунок 11 – Результати багатомірного пошуку аномалій для подій архівування «7Zip» методом KNN

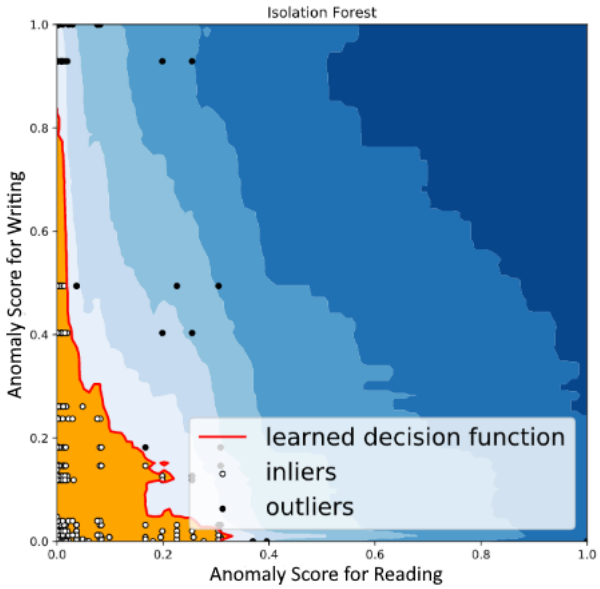

Рисунок 12 – Результати багатомірного пошуку аномалій для подій архівування «7Zip» методом IF

Таблиця 2 – Порівняльний аналіз ідентифікації стану КС, за умови запуску різних системних процесів, у вигляді проценту виявлених аномальних подій алгоритмами KNN та IF

Назва процесу	Процент виявлених аномальних подій алгоритмом IF	Процент виявлених аномальних подій алгоритмом KNN
SystemProcesses	5,00	5,00
ZipFile7Zip	8,93	8,38
ZipFileSystemZipper	8,76	7,23
OpenFoldersAndFiles	8,80	8,51
ExtractFilesSystemZipper	8,91	5,43
ExtractFiles7Zip	8,79	2,77
EditingTxtFile	8,87	4,13
DeleteToRecycleBin	43,36	38,51
DeleteFilesPerm	54,08	43,81
CopyFiles71	8,68	8,18
CopyFiles	9,09	6,85
VirusPetya	24,94	21,27

5 РЕЗУЛЬТАТИ

Якість класифікації методу на основі алгоритму IF оцінено за допомогою ROC-аналізу. Як видно із рис.13 площа під ROC AUC дорівнює 81.43%, тобто даний алгоритм є якісним. Точність класифікації складає 89.83%.

Для оцінки оперативності ідентифікації стану КС розробленим методом виконано порівняльний аналіз з методом ідентифікації на основі ансамблю дерев рішень, побудованих за алгоритмом J48 [20].

Отримано, що швидкість ідентифікації стану КС методом на основі алгоритму J48 є, в середньому, майже в 10 раз меншою відносно швидкості методом на основі алгоритму KNN та в 21 раз меншою відносно швидкості методом на основі алгоритму IF (рис. 14).

6 ОБГОВОРЕННЯ

При вирішенні завдань, пов’язаних з діагностикою та захистом комп’ютерних інформаційних ресурсів було виявлено ряд обмежень використання існуючих комп’ютеризованих систем ідентифікації стану КС. Поява аномалій, породжених вторгненнями в КС з невстановленими, або нечітко визначеними властивостям, наявність великого обсягу параметрів функціонування КС, відсутність розмічених даних призводить до труднощів з адекватного відбору показників її аномальної поведінки в умовах зовнішніх впливів і розробки критерію оцінки, що відповідає обраним показникам. Крім того, за умови відсутності розмітки даних, має місце проблема проведенням відбору моделей та перевірки якості їх роботи, що у сукупності з вищеописаними причинами

DOI 10.15588/1607-3274-2021-1-11

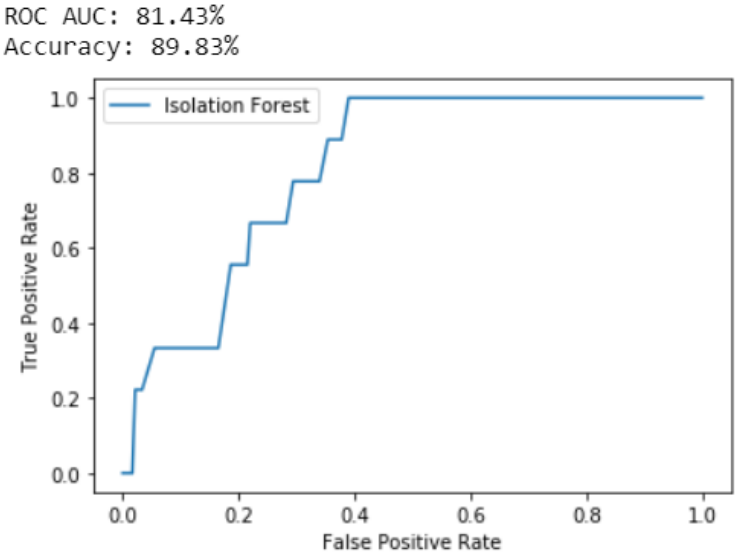

Рисунок 13 – Оцінка якості класифікації алгоритмом IF

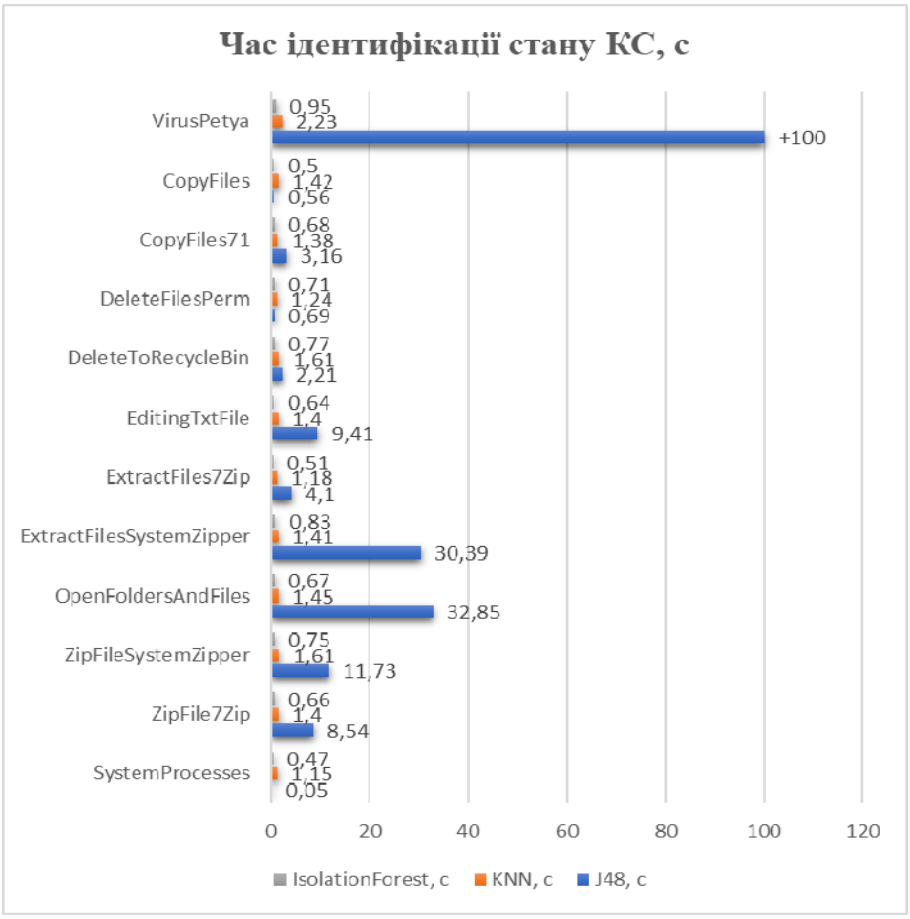

Рисунок 14 – Час ідентифікації стану КС

призводить до суттєвої розбіжності якості та слабкої практичної придатності окремих методів. Більше того, існуючі методи, в основному, тільки ідентифікують стан КС, але не надають можливість виділити назву процесу, який спричинив її аномальний стан.

Саме тому, проведені дослідження дозволили запропонувати метод ідентифікації стану КС на основі процедури групування вихідних даних та використання технології машинного навчання на основі алгоритму IF, який надає можливість не тільки ідентифікувати стан КС, але і виділити аномальні процеси. Проведені експерименти дозволили оцінити

DOI 10.15588/1607-3274-2021-1-11

точність та оперативність ідентифікації стану КС, практичну значимість та перспективи подальших досліджень.

ВИСНОВКИ

Таким чином, у роботі вирішено завдання підвищення оперативності ідентифікації стану функціонування комп'ютерної системи.

Наукова новизна отриманих результатів полягає в тому, що вперше запропоновано метод ідентифікації стану КС, який відрізняється комплексним використанням процедури групування нерозмічених вихідних даних та технології машинного навчання на основі алгоритму «Isolation Forest», що надає можливість ідентифікувати стан комп'ютерної системи і виділити назву процесу, який спричинив аномальний стан.

Для формування вихідних даних розроблено процедуру збору інформації у вигляді подій функціонування операційної системи. Виконано аналіз показників функціонування КС та оцінку їх інформативності. Аналіз подій ОС показав, що найбільш інформативними показниками її функціонування є операції читання та запису. Для формування єдиного датасету, операції читання та запису об'єднано в один масив груп подій та зіставлено з назвою процесу, що надалі надає можливість виділити назву процесу, який спричиняє аномальний стан КС.

У якості складових методу ідентифікації стану КС досліджено алгоритми: IF та KNN.

Отримано, що алгоритм IF є більш якісним та може бути використаним у якості складової методу ідентифікації стану КС.

Проведено оцінку якості запропонованого методу ідентифікації стану КС за допомогою ROC-аналізу та виконано оцінку оперативності. Отримано, що алгоритм є якісним: ROC AUC складає 81,43%, а точність виявлення аномалій складає 89,83%. При цьому, швидкість ідентифікації стану КС на основі процедури групування вихідних даних з використанням технології машинного навчання на основі алгоритму IF є, в середньому, в 21 раз вищою, ніж швидкість ідентифікації на основі ансамблю дерев рішень, побудованих за алгоритмом J48.

Практична значимість полягає в тому, що розроблений метод реалізований програмно і досліджений під час розв'язання задачі ідентифікації стану комп'ютерної системи.

Проведені експерименти підтвердили працездатність запропонованого методу, що надає можливість рекомендувати його для практичного використання у якості експрес-методу аналізу стану КС та не тільки ідентифікувати стан КС, але і виділити назву аномального процесу.

Перспективи подальших досліджень можуть полягати в розробці ансамблю нечітких дерев рішень на основі запропонованого методу, оптимізації його програмної реалізації та підвищення якості класифікації.

ПОДЯКИ

Робота виконана за підтримки міжнародного проекту Ерасмус + «Digital competence framework for Ukrainian teachers and other citizens», dComFra (598236-EPP-1-2018-1-IT-EPPKA2-CBHE-SP).

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Kelleher J. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples and Case Studies / J. Kelleher, B. Namee, A. Archi // The MIT Pres. – 2015. – 642 p.
2. Identification of the state of an object under conditions of fuzzy input data / [S. Semenov, O. Sira, S. Gavrylenko, N. Kuchuk] // Eastern-European Journal of Enterprise Technologies. – 2019. – Vol. 1, № 4 (97). – P. 22–29. DOI: 10.15587/1729-4061.2019.157085
3. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень / С. О. Субботін. – Запоріжжя : ЗНТУ, 2008. – 341 с.
4. Большаков А. С. Обнаружение аномалий в компьютерных сетях с использованием методов машинного обучения. Телекоммуникационные устройства и системы / А. С. Большаков, Е. В. Губанкова // Телекоммуникационные устройства и системы. – 2020. – Т. 10, № 1. – С. 37–42.
5. Линдигрин А. Н. Сравнительный анализ методов машинного обучения в задачах обнаружения сетевых аномалий / А. Н. Линдигрин // Известия Тульского государственного университета. Технические науки. – 2019. – № 12. – С. 400–404.
6. Wang S. Adapting naive Bayes tree classification / S. Wang, L. Jiang, C. Li // Knowledge and Information system. – 2015. – Vol. 44, № 1. – P. 77–89. DOI: 10.1007/s10115-014-0746-y
7. Кокорева Я. Поэтапный процесс кластерного анализа данных на основе алгоритма кластеризации k-means / Я. Кокорева, А. Макаров // Молодой ученый. – 2015. – № 13. – С. 126–128.
8. Catania C. Autonomous Labelling Approach to Support Vector Machine Algorithms for Network Traffic Anomaly Detection / Carlos Catania, Facundo Bromberg, Carlos Garcia Garino // Expert Systems lications: An International Journal Archive. – 2012. – No. 39. – P. 45–49. DOI: 10.1016/j.eswa.2011.08.068
9. Pankaj M. Long Short Term Memory Networks for Anomaly Detection in Time Series / Malhotra Pankaj// ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. – 2015. – P. 89–94.
10. Ben–Gal I. Efficient Construction of Decision Trees by the Dual Information Distance Method / Irad Ben-Gal, Alexandra Dana, Niv Shkolnik, Gonen Singer // Quality Technology & Quantitative Management. – 2014. – Vol. 11, № 1. – P. 133–147. DOI: 10.1080/16843703.2014.11673330
11. Aggarwal C. Theoretical foundations and algorithms for outlier ensembles / C. Aggarwal, S. Sathe // ACM SIGKDD Explorations Newsletter. – 2015. – Vol. 17, № 1. – P. 24–47. DOI: 10.1145/2830544.2830549
12. Zimek A. Ensembles for unsupervised outlier detection: challenges and research questions a position paper / A. Zimek, R. Campello, J. Sander // Acm Sigkdd

DOI 10.15588/1607-3274-2021-1-11

Explorations Newsletter. – 2014. – Vol. 15, № 1. – P. 11–22. DOI: 10.1145/2594473.2594476.
13. Aggarwal C. Outlier ensembles: position paper / C. Aggarwal // ACMSIGKDD Explorations Newsletter. – 2017. – Vol. 14, № 2. – P. 49–58. DOI: 10.1145/2481244.2481252
14. Rafika B. Boosted Decision Trees for Lithiasis Type Identification / Boutalbi Rafika, Chitibi Kheir Eddine // International Journal of Advanced Computer Science and Applications. – 2015. – Vol. 6, № 6. – P. 197–202. DOI: 10.14569/IJACSA.2015.060628.
15. Chandola, V. Anomaly detection: A survey / V. Chandola, A. Banerjee, V. Kumar // ACM Comput. Surv. – 2009. – № 41. – P. 15–58. DOI: 10.1145/1541880.1541882
16. Chowdhury M. Malware Analysis and Detection Using Data Mining and Machine Learning Classification / M. Chowdhury, A. Rahman., Rz. Islam // International Conference on Applications and Techniques in Cyber Security and Intelligence. – 2018. – P. 266–274.
17. Breiman L. Random Forests / L. Breiman // Statistics Department University of California Berkeley: Machine Language. – 2001. – P. 5–32.
18. Шелухин О. И. Применение алгоритма «изолирующий лес» для решения задач обнаружения аномалий / О. И. Шелухин, М. В. Полковников // Решение. – 2019. – С. 186–188.
19. Fei Tony L. Isolation forest / Liu Fei Tony, Ting Kai Ming, Zhou Zhi-Hua // Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. – 2008. – P. 413–422. DOI: 10.1109/ICDM.2008.17
20. Gavrylenko S. The ensemble method development of classification of the computer system state based on decisions trees / S. Gavrylenko, I. Sheverdin, M. Kazarinov // Advanced Information Systems. – 2020. – P. 5–10. DOI: 10.20998/2522-9052.2020.3.01

Стаття надійшла до редакції 12.10.2020.
Після доробки 27.12.2020.

УДК 004.8

РАЗРАБОТКА МЕТОДА ИДЕНТИФИКАЦИИ СОСТОЯНИЯ КОМПЬЮТЕРНОЙ СИСТЕМЫ НА ОСНОВЕ АЛГОРИТМА «ISOLATION FOREST»

Гавриленко С. Ю. – д-р техн. наук, доцент, профессор кафедры «Вычислительная техника и программирование», Национальный технический университет «Харьковский политехнический институт», Харьков, Украина.

Шевердин И. В. – аспирант кафедры «Вычислительная техника и программирование», Национальный технический университет «Харьковский политехнический институт», Харьков, Украина.

АНОТАЦІЯ

Актуальность. Рассмотрена задача идентификации состояния компьютерной системы. Объектом исследования является процесс идентификации состояния компьютерной системы. Предметом исследования являются методы и средства идентификации состояния компьютерной системы.

Цель. Целью работы является разработка метода идентификации состояния компьютерной системы.

Метод. Разработан метод идентификации состояния компьютерной системы на основе комплексного использования процедуры группировки неразмеченных исходных данных и технологии машинного обучения на основе алгоритма «Isolation Forest», который предоставляет возможность идентифицировать состояние компьютерной системы и выделить название процесса, который вызвал аномальный состояние. Для этого предложена процедура и разработано программное приложение для сбора статистических данных в виде событий функционирования операционной системы и выполнен их анализ. Получено, что наиболее информативными являются операции чтения и записи. Для формирования единого датасета, операции чтения и записи сопоставлены с названием процесса и объединены в один массив групп событий, что в дальнейшем позволяет выделить процесс, который вызывает аномальный состояние компьютерной системы. По результатам исследования, в качестве составляющей метода идентификации состояния компьютерной системы использовано ансамблевый алгоритм «Isolation Forest». Проведена оценка точности и оперативности разработанного метода идентификации состояния компьютерной системы.

Результаты. Разработанный метод реализован программно и исследован при решении задачи идентификации аномалий функционирования компьютерной системы.

Выводы. Проведенные эксперименты подтвердили работоспособность предложенного метода, позволяет рекомендовать его для практического использования с целью повышения оперативности идентификации состояния компьютерной системы и использования его в качестве экспресс-метода. Перспективы дальнейших исследований могут заключаться в разработке ансамбля нечетких деревьев решений на основе предложенного метода, оптимизации его программных реализации.

КЛЮЧЕВЫЕ СЛОВА: компьютерная система, события операционной системы, аномальный состояние, идентификация, машинное обучение, алгоритм «Isolation Forest».

UDC 004.8

DEVELOPMENT OF METHOD TO IDENTIFY THE COMPUTER SYSTEM STATE BASED ON THE «ISOLATION FOREST» ALGORITHM

Gavrylenko S. Y. – Dr. Sc., Associate Professor, Professor at Department of Computer Engineering and Programming, National Technical University «Kharkiv Polytechnik Institute», Kharkiv, Ukraine.

Sheverdin I. V. – Post-graduate student at Department of Computer Engineering and Programming, National Technical University «Kharkiv Polytechnik Institute», Kharkiv, Ukraine.

DOI 10.15588/1607-3274-2021-1-11

ABSTRACT

Context. The problem of identification a computer system state was investigated. The object of the research is the identification process of the computer system state. The subject of the research is computer system state identifying means and methods.

Objective. The purpose of the work is to develop a method for identifying the computer system state.

Method. The method has been developed for identifying a computer system state based on integrated use the procedure for grouping unlabeled initial data and using machine learning technology based on the «Isolation Forest» algorithm, which provides to identify a computer system state and to distinguished the process name that initiated the abnormal state. Therefore, for collecting statistical data in the form of operating system functioning events, data method has been proposed and developed along with software. The analysis of functioning events has been performed. The result of analysis showed that the most informative are read and write operations. To set up a single dataset, read and write operations compared with the process name and combined into one array of event groups, so that it is possible to single out the process that causes the abnormal state of the computer system. As a result of the research, the «Isolation Forest» algorithm has been selected as a component of the method for identifying the computer system state. An accuracy and efficiency assessment of the developed method of identifying a computer system state has been carried out.

Results. The developed method is implemented and investigated when solving the problem of identifying anomalies in the functioning of computer systems.

Conclusions. The experiments carried out confirmed the efficiency of the proposed method. It allows us recommended the method for practical use in order to improve efficiency of identifying the computer system state and use it as an express method. Areas for further research may lie in the creation of the ensemble of fuzzy trees based on the proposed method and optimization of this software implementation.

KEYWORDS: computer system, operating system events, abnormal state, identification, machine learning, Isolation Forest algorithm.

REFERENCES

1. Kelleher, J., B. Namee, A. Archi Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies, *The MIT Pres*, 2015, 642 p.
2. Gavrylenko S., Semenov S., Sira O., Kuchuk N. Identification of the state of an object under conditions of fuzzy input data. *Eastern-European Journal of Enterprise Technologies*, 2019, Vol. 1, No. 4 (97), pp. 22–29. DOI: 10.15587/1729-4061.2019.157085
3. Subbotin S.O. Podannya j obrobka znan u sistemah shtuchnogo intelektu ta pidtrimki prijnyattya rishen. Zaporizhzhya, ZNTU, 2008, 341 p.
4. Bolshakov A.S., Gubankova E.V. Obnaruzhenie anomalij v kompyuternyh setyah s ispolzovaniem metodov mashinnogo obucheniya. *Telekommunikacionnye ustrojstva i sistemy*, 2020, Vol. 10, No. 1, pp. 37–42.
5. Lindigrin A. N. Sravnitelnyj analiz metodov mashinnogo obucheniya v zadachah obnaruzheniya setevyh anomalij, *Izvestiya Tulskogo gosudarstvennogo universiteta. Tehnicheskie nauki*, 2019, No. 12, pp. 400–404.
6. Wang S., Jiang L., Li C. Adapting naive Bayes tree classification, *Knowledge and Information system*, Vol. 44, No. 1, pp. 77–89. DOI: 10.1007/s10115-014-0746-y
7. Kokoreva Ya., Makarov A. Poetapnyj process klasternogo analiza dannyh na osnove algoritma klasterizacii k-means, *Molodoj uchenyj*, 2015, No. 13, pp. 126–128.
8. Carlos A., Catania, Facundo Bromberg, Carlos Garcia Garino. An Autonomous Labelling Approach to Support Vector Machine Algorithms for Network Traffic Anomaly Detection, *Expert Systems lications: An International Journal Archive*, 2012, No. 39, pp. 45–49. DOI: 10.1016/j.eswa.2011.08.068
9. Malhotra Pankaj, Long Short Term Memory Networks for Anomaly Detection inTime Series, *ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.
10. Irad Ben-Gal, Alexandra Dana, Niv Shkolnik, Gonen Singer. Efficient Construction of Decision Trees by the Dual Information Distance Method, *Quality Technology & Quantitative Management*, 2014, Vol. 11, No. 1, pp. 133–147. DOI: 10.1080/16843703 .2014.11673330
11. Aggarwal C. C., Sathe S. Theoretical foundations and algorithms for outlier ensembles, *ACM SIGKDD Explorations Newsletter*, 2015, Vol. 17, No. 1, pp. 24–47. DOI: 10.1145/2830544.2830549
12. Zimek A., Campello R. J. G. B., Sander J. Ensembles for unsupervised outlier detection: challenges and research questions a position paper, *Acm Sigkdd Explorations Newsletter*, 2014, Vol. 15, No. 1, pp. 11–22. DOI: 10.1145/2594473.2594476
13. Aggarwal C. C. Outlier ensembles: position paper, *ACMSIGKDD Explorations Newsletter*, 2017, Vol. 14, No. 2, pp. 49–58. DOI: 10.1145/2481244.2481252
14. Boutalbi Rafika, Chitibi Kheir Eddine. Boosted Decision Trees for Lithiasis Type Identification, *International Journal of Advanced Computer Science and Applications*, 2015, Vol. 6, No. 6, pp. 197–202.
15. Chandola V., Banerjee A., Kumar V. Anomaly detection:survey, *ACM computing surveys (CSUR)*, 2009, No. 41, pp. 15–58. DOI: 10.1145/1541880.1541882.
16. Chowdhury M. Malware Analysis and Detection Using Data Mining and Machine Learning Classification, *International Conference on Applications and Techniques in Cyber Security and Intelligence, ATCI*, 2018, pp. 266–274.
17. Breiman, L. Random Forests, *Machine Language*, 2001, No. 45 (1), pp. 5–32.
18. Sheluhin O. I., Polkovnikov M. V. Primenenie algoritma «izoliruyushij les» dlya resheniya zadach obnaruzheniya anomalij. *Reshenie,* 2019, No. 1, pp. 186–18.
19. Liu, Fei Tony, Ting, Kai Ming and Zhou, Zhi-Hua. Isolation forest, *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, December 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17
20. Gavrylenko S., Sheverdin I., Kazarinov M. The ensemble method development of classification of the computer system state based on decisions trees, *Advanced Information System*, 2020, Vol. 4, No. 2, pp. 5–10. DOI: 10.20998/2522-9052.2020.3.01

DOI 10.15588/1607-3274-2021-1-11

UDC 004.896

IMPLEMENTATION OF THE INDICATOR SYSTEM IN MODELING OF COMPLEX TECHNICAL SYSTEMS

Leoshchenko S. D. – Post-graduate student of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

Subbotin S. A. – Dr. Sc., Professor, Head of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

Oliinyk A. O. – PhD, Associate Professor, Associate Professor of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

Narivs'kiy O. E. – Dr. Sc., Professor, Technical Director, Ukrspecmash ltd., Berdyansk, Ukraine.

ABSTRACT

Context. The problem of determining the optimal topology of a neuromodel, which is characterized by a high level of logical transparency in modeling complex technical systems, is considered. The object of research is the process of applying an indicator system to simplify and select the topology of neuromodels.

Objective of the work is to develop and use a system of indicators to determine the level of complexity of the modeling problem and gradually select the optimal logically transparent topology of the neuromodel.

Method. A method is proposed for selecting an optimal, logically transparent neural network topology for modeling complex technical systems using a system of corresponding indicators. At the beginning, the method determines the overall level of complexity of the modeling task and, using the obtained estimate, determines the method for further optimization of the neuromodel. Then, using Task data and input data characteristics, the method allows to obtain the most optimal structure of the neural model for further modeling of the system. The method reduces trainingvtime and increases the level of logical transparency of neuromodels, which significantly expands the practical use of such models, without using neuroevolution methods, which may not be justified by resource-intensive tasks.

Results. The developed method is implemented and investigated in solving the problem of modeling the dynamics of pitting processes of steel alloys. Using the developed method made it possible to reduce the training time of the model by 22%, depending on the computing resources used. The method also increased the level of logical transparency of the model by reducing the number of computing nodes by 50%, which also indicates faster and more efficient use of resources.

Conclusions. The conducted experiments confirmed the operability of the proposed mathematical support and allow us to recommend it for use in practice in the design of topologies of neuromodels for further solving modeling, diagnosis and evaluation problems. Prospects for further research may consist in the development of methods for structural optimization of previously synthesized models and the development of new methods for feature selection.

KEYWORDS: complexity assessment, indicator system, modeling, neuromodel, sampling, training, error, gradient.

ABBREVIATIONS

A3C is asynchronous advantage actor-critic
ANN is an artificial neural net;
CAwP is complexity is associated with people;
OC is organized complexity;
OS is organized simplicity;
RC is random complexity;
RNN is recurrent neural network.

NOMENCLATURE

K_{input} is a number of element types in the neural network;

K_{corrY} is a number of independent variables that strongly correlate with the original features;

K_{imp} is a number of the most significant independent variables among factors;

$K_{ntcorrX}$ is a number of independent variables that are weakly dependent on others or do not correlate with each other;

n is a number of input features that characterize sample instances;

N_i is a multiple neurons at the network input;

N_{i_l} is a neuron at the network input;

N_o is a multiple neurons at the network output;

N_{o_p} is a neuron at the network output;

N_h is a multiple neurons of the hidden network layer;

N_{h_r} is a hidden network layer neuron;

$Num_{elemtype}$ is a number of element types in the neural network;

NN is a neural network;

NN_{struct} is a structure of neural network;

l is a number of neurons at the network input;

$Lev_{accmeas}$ is a measurement accuracy level;

Lev_{fctr} is a level of significant and less significant and/or non-significant factors;

Lev_{manag} is a level of possible control and management;

Lev_{task} is a conditional difficulty level of the task;

$Lev_{smplfctn}$ is a level of possible simplification of the structure;

DOI 10.15588/1607-3274-2021-1-12

m is a number of dependent (categorical) features of sample instances;

p is a number of neurons at the network output;

q is a number of connections between neurons in the network;

r is number of neurons in the hidden network layer;

$Sample$ is a data set;

w is a multiple of connections between neurons;

w_q is a connection between neurons in the network;

x_n is a independent attribute of the sample instance;

X is a set of independent attribute (variables);

y_m is a value of the dependent variable (attribute) of the sample instance;

Y is a set of values of dependent variables.

INTRODUCTION

Usually, it can be concluded that ANN can developed hidden knowledge from data: forecasting, classification, pattern recognition skills, etc. are formed, but its logical structure usually remains hidden from the user [1–3]. The problem of manifestation (contrast) of this hidden logical structure is solved by reducing neural networks to a special logically transparent sparse form.

When designing an ANN the main questions usually arise [1]:

1) determine the set of neurons ($N = \{N_i, N_o, N_h\}$, where $N_i = \{N_{i_1}, N_{i_2}, ..., N_{i_l}\}$, $N_o = \{N_{o_1}, N_{o_2}, ..., N_{o_p}\}$, $N_h = \{N_{h_1}, N_{h_2}, ..., N_{h_r}\}$, $l = 1,2,...,|N_i|$, $p = 1,2,...,|N_o|$, $r = 1,2,...,|N_h|$) required to solve the problem;

2) determine the structure NN_{struct} (topology, number of layers, etc.) of the ANN.

Usually, after this or even at the testing stage, a third task/question arises: how to make the work of the ANN understandable to the user (logically transparent) and what benefits such an understanding can bring [4].

When solving the first question, the researcher is faced with two opposite points of view [5, 6]. One of them claims that the more neurons you use, the more reliable the network will be. Proponents of this position cite the example of the human brain. Indeed, the more neurons there are, the greater the number of connections between them, and the more complex tasks a neural network can solve. In addition, if you use a deliberately larger number of neurons than is necessary to solve the problem, then the neural network will definitely train. If you start with a small number of neurons, the network may not be able to train how to solve the problem, and the whole process will have to be repeated first with a large number of neurons. This point of view is popular among neural network software developers and is confirmed by a number of criteria [4–6].

The second point of view is based on an empirical rule: the more configurable parameters, the worse the approximation of the function in those areas where its values were previously unknown [4–6]. From a mathematical point of view, the tasks of training neural networks are reduced to continuing the function of a given finite number of points for the entire domain of definition. The second approach defines the required number of neurons as the minimum required. The main disadvantage is that this minimum required number is unknown in advance, and the procedure for determining it by gradually increasing the number of neurons is very time-consuming [5]. Based on the experience of various groups in the field of medical diagnostics, space navigation and psychology, it can be noted that all these tasks never required more than a few dozen neurons.

In the end, a network with a minimum number of neurons should better approximate the function, but determining this minimum number of neurons requires a lot of intellectual costs and experiments on network training. If the number of neurons is excessive, then you can get the result on the first attempt, but there is a risk of building a poor approximation. And as a specific methodology, you can use the approach: double the number of neurons in the network after each failed learning attempt.

An alternative approach is the use of neuroevolution methods [7–9]. When the ANN population is formed at the beginning. And then, in the course of using evolutionary operators (mutation, selection, crossover), the best solutions are obtained. In the form of big data with a large number of hidden relationships or incomplete data (or with the approach of teaching without a teacher), such methods and approaches show good results (high level of accuracy of work, choice of the optimal structure of the ANN). However, for not very complex tasks, such methods may not become justifiably resource-intensive, because they will work with populations of ANNs and constantly evaluate them [8]. However, there is a more reliable way to estimate the minimum number of neurons: using the contrast procedure [10]. In addition, the contrast procedure allow to answer the second question: what should be the network structure.

Therefore, using this example, we can conclude that at the beginning it is still necessary to more accurately assess the complexity of the problem in order to determine the strategy for finding the optimal structure of the neuromodel in the next one.

In this case, the term optimal structure of the ANN will be understood as the highest possible level of logical transparency of the ANN [1]. So, one of the main disadvantages of neural networks, from the point of view of many users, is that the ANN solves the problem, but cannot provide an explanation of exactly how the answer was received. In other words, you can't extract an algorithm for solving the problem from a trained network. In the case of using simple feed-forward ANN with a fixed and small number of neurons in the hidden layer, this can still be studied. However, for more complex topologies, we recommend using special methodologies, such as the contrast procedure.

That is why the scientific and applied task of developing new methods for assessing the complexity of the problem and then selecting the optimal structure of the ANN as a neuromodel in solving applied problems, such as modeling, classification and evaluation, is relevant.

DOI 10.15588/1607-3274-2021-1-12

The object of study is the process of using an indicator system to select the topology of a neuromodel.

Existing methodologies are general recommendations and do not have a solid mathematical basis for defining clear rules.

The subject of study is a system of criteria for determining the optimal structure of a neuromodel.

To date there are methods of neuroevolutionary synthesis of ANN, but they are quite resource-intensive, which may not be justified for simpler practical tasks. Therefore, the paper proposes a new approach based on presenting information about the task and evaluating input data.

The purpose of the work is to develop and use a system of indicators to determine the level of complexity of the modeling problem and gradually select the optimal logically transparent topology of the neuromodel.

1 PROBLEM STATEMENT

Let it be a problem which can be characterized by a conditional level of complexity (Lev_{task}). The difficulty level is a specific integral estimate consisting of several characteristics:

– the sample $Sample = \langle X, Y \rangle$ of input data, where $X = \{x_1, x_2, ...x_n\}$ is the set of independent variables-features, and $Y = \{y_1, y_2, ...y_m\}$ is the set of values of dependent variables, and-the number of input features that characterize sample instances;

– $Lev_{smplfctn}$ is the level of possible simplification of the structure (for example, the amount of input data);

– Lev_{fctr} is a total number of significant and less significant and/or non-significant factors;

– $Lev_{accmeas}$ is a level of measurement accuracy;

– Lev_{manag} is the level of possible control and management.

Based on this the problem can be present as follows: for the synthesis of ANN (NN), it is necessary to define a set of neurons $N = \{N_i, N_o, N_h\}$ consisting of subsets of input $N_i = \{N_{i_1}, N_{i_2}, ..., N_{i_l}\}, l = 1,2,...,|N_i|$, output $N_o = \{N_{o_1}, N_{o_2}, ..., N_{o_p}\}, p = 1,2,...,|N_o|$, hidden neurons $N_h = \{N_{h_1}, N_{h_2}, ..., N_{h_r}\}, r = 1,2,...,|N_h|$ and a set of weights of connections between neurons $w = \{w_q\}$. Having determined the values of the elements of sets, we can consider the synthesis of ANN is complete.

To do this, it will be used an estimate of the complexity of the problem, after receiving which we will solve how we will design the ANN topology NN_{struct} in the future. If the problem can be classified as OS by its level of complexity, then using the feature selection, we will get the most informative ones and using the study of input data, it will be indicated the required number of neurons in the hidden layer of the network $N_h = \{N_{h_1}, N_{h_2}, ..., N_{h_r}\}, r = 1,2,...,|N_h|$. After that, it will be possible to perform neuromodel training.

2 REVIEW OF THE LITERATURE

Complexity is a characteristic that reflects the extent to which the project or implementation of a system or element is difficult to understand and verify (ISO/IEC/IEEE 24765).

A complex system is a system consisting of many interacting components (subsystems), as a result of which a complex system acquires new properties that are absent at the subsystem level and cannot be reduced to the properties of the subsystem level.

The complexity of a system is determined by the number of its constituent elements and possible connections between them. The degree of complexity is measured by the variety of the system. Diversity characterizes the number of possible states of the system.

According to the work of Peter M. Senge [11], system complexity exists in two main forms.

Complexity of granularity (structural complexity) occurs as a result of a large number of systems, system elements, and established relationships in any of the two main topologies (hierarchy or network). This complexity is related to the systems as they are; namely, their static existence.

Dynamic complexity (behavioral complexity) is related to the relationships that arise between ready-made, functioning systems in the course of their operation, that is, between expected and even unexpected behavior that actually occurs.

Weaver [12] formulated the initial point of view, identifying the following categories of complexity: organized simplicity, organized complexity, random complexity. These categories and later reflections, in particular Flood and Carson [13] and the author of the book, give grounds for using the following classification of complexity.

OS occurs when there are a small number of essential factors and a large number of less significant and/or non-essential factors. At first, the situation may seem complicated, but after studying it, less significant and insignificant factors can be excluded from the picture and hidden simplicity can be discovered [12, 13].

OC prevails in such physical and abstract systems, the structure of which is organized in such a way as to be understandable, and therefore pliable to scientists in describing complex behavior and structuring the process of creating complex systems and managing their life cycles. This is a wealth that should not be oversimplified [12, 13].

RC occurs when there are many variables that exhibit random, random behavior to a high degree. It can also be the result of a lack of necessary control over the structure of complex heterogeneous systems due to inadequate architecture management during the system lifecycle (creeping complexity) [12, 13].

CAwP occurs where the perception of any system causes a sense of complexity. In this context, people become systems of observation. This category also can be linked to systems in which people are elements and can

DOI 10.15588/1607-3274-2021-1-12

contribute significantly to organized simplicity, organized complexity, and disorderly complexity. Reasonable or unreasonable behavior of individuals in specific situations is naturally a significant factor in relation to complexity [12, 13].

3 MATERIALS AND METHODS

As already reported, it will be used the difficulty assessment of the task at the beginning for future work. So, based on the estimate, it will be determined which of the 4 types of Weaver [12, 13] classification the task belongs to. The main distribution is shown in Fig. 1. This is how the input data of the problem is studied at the beginning: the larger the sample size of the input data, the higher the score will be, because such tasks require either more complex operations for preprocessing input data (data reduction, feature selection, etc.), or more complex topologies of the ANN to encode all the features and track their relationship.

Next, it will defined the level of possible simplification of the structure ($Lev_{smplfctn}$). Usually it will be talked about the general possibility of deleting small information input data. So, $Lev_{smplfctn} = -1$ if the number of input features is not large (by default, it is suggested up to 200, but it is possible to increase/decrease due to the technical resources used) and simplification is not possible, $Lev_{smplfctn} = 0$ if the number is large (from 200 to 1000), but it is possible to reduce them, and $Lev_{smplfctn} = 1$ if the number is large and the reduction is not possible (when working online) or too resource-intensive.

The study of the total number of significant and less significant and/or non-significant factors requires preliminary factor analysis [14–18]. In view of the simplification of this stage, it is proposed to perform the usual factor analysis using the principal component method. So, $Lev_{fctr} = -1$ if the factor analysis did not show a large number of significant factors; $Lev_{fctr} = 0$ if the number of significant factors is not large, but there are a certain number of less significant factors, or during the factor analysis there were correlation matrices that were poorly determined (their determinant was 0); and $Lev_{fctr} = 1$ if the factor analysis showed a large number of significant factors and/or a large number of less significant ones.

During determining the level of measurement accuracy it will be guided by two main cases:

– $Lev_{accmeas} = 1$ if the measurements are related to the human factor, then people (or human sensory systems) become observation systems;

– $Lev_{accmeas} = 0$ if measurements are made using special sensors (sensor systems).

The level of possible control and management can also show only two states, so:

– $Lev_{manag} = 1$ if the system exhibits highly random, disorderly behavior, or this behavior is the result of a lack of necessary control over the structure of complex heterogeneous systems;

– $Lev_{manag} = 0$ if the system was under control (human or automated) and even if certain signs (e.g. deformities) were under control.

It should be noted that this paper will provide mechanisms exclusively for the first category: OS [12, 13]. After all, it should be noted that if a system, for example, after evaluating a problem, its complexity is determined as an OC, then at the beginning it is recommended to synthesize a neuromodel (for example, using neural network methods), and only then carry out structural optimization, if necessary. This is explained by the fact that in this case it is not possible to select information features according to the problem conditions, but it is possible to simplify and exclude links in the model. If the score is defined as CAwP, then it is recommended to check the truth of the input data before selecting information features. In the form of RC, it is recommended to use a special class of neural network methods for the synthesis of neuromodels, namely swarm intelligence methods, or use the RNN topology with the A3C method to train it [19, 20]. This approach will allow for much better data processing, because swarm intelligence methods do not have the disadvantages of most neuroevolutionary methods and emulate the behavior of independent agents [7–9], and RNN allows you to encode most hidden relationships in its structure.

Therefore, if the task was evaluated by the OS, we will proceed to the second stage. First, you need to determine the number of element types in the neural network. In our case $Num_{elemtype} = 3$, by default, because it is assumed that only input, hidden and output neurons are used. Next, you need to determine the number of independent variables that strongly correlate with the original features K_{corrY}. Additionally, the number of independent variables that are weakly dependent on others or do not correlate with each other is determined $K_{ntcorrX}$. Additionally, information about the number of the most significant independent variables among the factors is pulled up from the previous stage K_{imp}. After that, the number of neurons in the hidden layer can be determined by the formula:

$$\left|N_h\right| = K_{input} - K_{corrY} - K_{imp} - K_{ntcorrX}. \qquad (1)$$

4 EXPERIMENTS

To research the developed method, it will be used the problem of modeling the dynamics of pitting processes of steel alloys [16, 17]. Pitting (spot) corrosion is a type of corrosion destruction to which only passive metals and alloys are subjected. Pitting corrosion is observed in nickel, zirconium, chromium – nickel, chromium, aluminum alloys, etc. during pitting corrosion, only certain areas of the surface are destroyed, on which deep lesions are formed-pitting (point ulcers). AISI 304 steel is used in

DOI 10.15588/1607-3274-2021-1-12

many areas, for example, in the manufacture of prefabricated and welded metal structures, components of pipe fittings, as well as household equipment: balcony and staircase fences, kitchen equipment, automobile exhaust systems [21, 22]. Such structural materials have high corrosion resistance in many aggressive environments, but can be subjected to pitting corrosion in solutions containing chloride (C_{Cl}). That is why it is an urgent task to build models of the dependence of pitting formation on the characteristics of the steel used and reversible solutions. A promising basis for such models is the ANN, due to their ability to learn from experimental data, generalize data, and extract knowledge from data [1–3]. ANNs can be used to build models of dependencies based on observations in the absence of uncertainty of dependencies in the data. Table 1 shows a fragment of the input data.

The data sample consisted of 50 point observations characterized by the values of input features: x_1 is pH (medium characteristic, dimensionless); x_2 is C_{Cl}, mg/l (chloride concentration in the medium); x_3 is $V_{ок}$, Vol. % (volume of oxides in steel); x_4 is the amount of oxides up to 1.98 microns in size per 100 visual obstacles of an optical microscope ($\times 320$) in steel; x_5 is the amount of oxides ranging in size from 1.98 microns to 3.95 per 100 visual interference of an optical microscope ($\times 320$) in steel; x_6 is $L_{ок}$, microns (average distance between oxides in steel); x_7 is d_3, microns (average diameter of austenite grain in steel); x_8 is P_α, Vol. % (volume-ferrite in steel); x_9 is C the mass percentages (carbon content in Steel); x_{10} is Mn, % (manganese content in steel); x_{11} is Si, % (silicon content in Steel); x_{12} is Cr, % (chromium content in steel); x_{13} is Ni, % (nickel content in steel); x_{14} is N, % (nitrogen content in steel); x_{15} – Ti, % (titanium content in Steel); x_{16} is S, % (sulfur content in steel); x_{17} is P, % (phosphorus content in steel); x_{18} is X_o, m^3/kg (specific magnetic susceptibility of austenite).

Dependent variables: y_1 is ΔCr, mg (10^{-5}) the chromium losses from steel after its exposure in solution; y_2 is ΔFe, mg (10^{-5}) the iron losses from steel after its exposure in solution; y_3 is ΔNi, mg (10^{-5}) the nickel losses from steel after its exposure in solution; $y_4 = ZCr$; y_5 if $ZCr < 1$, then steel in solution with such parameters (signs x_1, x_2) is subjected to pitting corrosion with the formation of stable pitting, otherwise, steel in solution with such parameters (signs x_1, x_2) is subjected to pitting corrosion with the formation of metastable pitting; $y_6 = ZNi$; y_7 if $ZNi < 1$, then pittings grow intensively, otherwise, pittings do not increase in size intensively.

5 RESULTS

The results of factor analysis are demonstrated at the table 2. This results will be used for evaluate the level of task and for calculations of number hidden neurons.

Pairwise correlation coefficients of input features are demonstrated at table 3 and will be used in future calculations of number hidden neurons.

Pairwise correlation coefficients of input and output features are demonstrated at table 4 and will be used in future calculations of number hidden neurons.

Table 5 demonstrate the comparison of neuromodel learning outcomes.

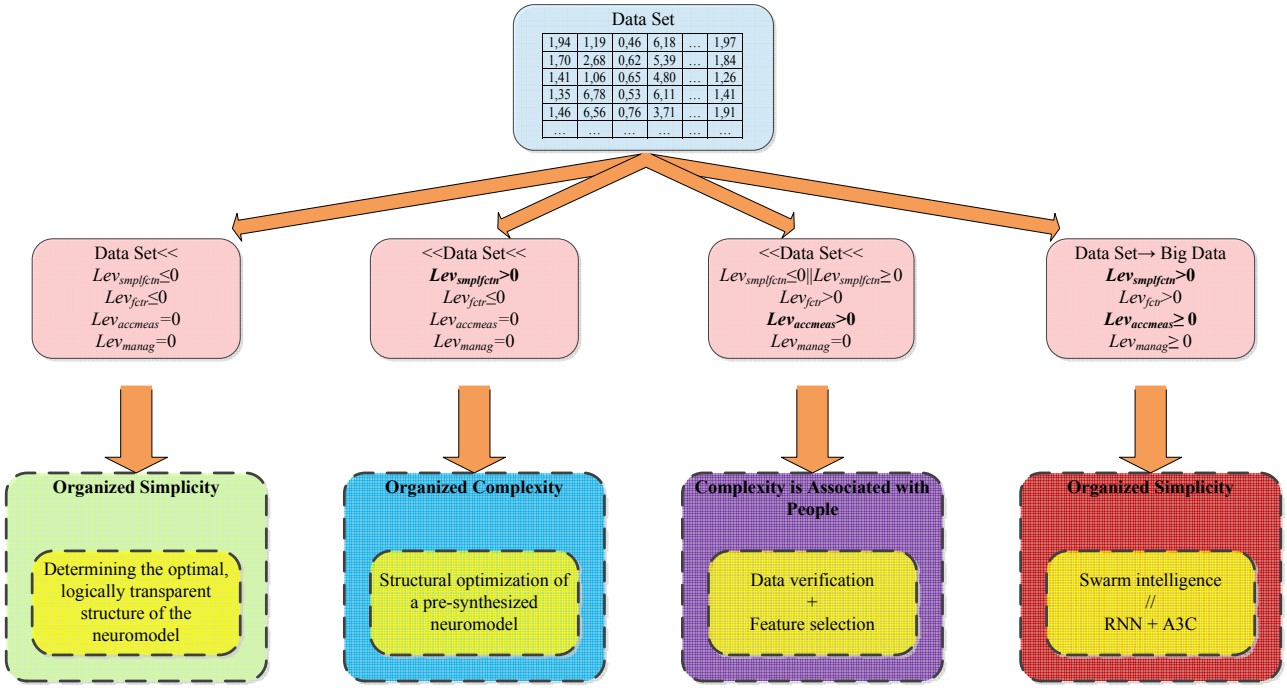

Figure 1 – General scheme for determining the level of difficulty

DOI 10.15588/1607-3274-2021-1-12

Table 1 – General information about data set

x_1	x_2	x_3	…	x_{18}	y_1	y_2	y_3	…	y_7
4	300	0,0199	…	2,27	0,01421	0,18776	0,01620	…	0
4	300	0,0216	…	2,27	0,01109	0,63335	0,01337	…	0
5	300	0,0199	…	2,27	0,00275	0,00296	0,01479	…	1
5	300	0,0216	…	2,27	0,00379	0,00275	0,01400	…	1
…	…	…	…	…	…	…	…	…	…
8	600	0,0216	…	2,27	0,00412	0,02114	0,03708	…	1

Table 2 – Results of factor analysis

X	Factor 1	Factor №2	Factor №3
x_1	–0.000000	–0.000000	–0.894181
x_2	0.000000	0.000000	–0.447706
x_3	–0.989883	0.031646	–0.000000
x_4	0.677879	–0.644594	0.000000
x_5	–0.980149	0.158833	0.000000
x_6	0.817799	0.493587	–0.000000
x_7	0.837964	–0.210144	–0.000000
x_9	0.533363	0.743210	0.000000

Table 3 – Pairwise correlation coefficients of input features

r	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{16}	x_{17}	x_{18}
x_1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
x_2	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
x_3	0.00	0.00	1.00	–0.68	0.99	–0.83	–0.80	0.94	–0.46	0.15	–0.06	0.32	–0.13	–0.49	–0.16	0.25	0.89
x_4	0.00	0.00	–0.68	1.00	–0.78	0.18	0.59	–0.79	0.00	–0.39	–0.49	–0.86	0.58	–0.09	–0.40	0.31	–0.62
x_5	0.00	0.00	0.99	–0.78	1.00	–0.75	–0.80	0.96	–0.39	0.20	0.04	0.44	–0.23	–0.40	–0.05	0.15	0.89
x_6	0.00	0.00	–0.83	0.18	–0.75	1.00	0.54	–0.71	0.69	–0.02	0.32	0.17	–0.16	0.82	0.61	–0.65	–0.73
x_7	0.00	0.00	–0.80	0.59	–0.80	0.54	1.00	–0.61	0.28	–0.08	0.18	–0.31	0.13	0.13	–0.34	–0.03	–0.60
x_8	0.00	0.00	0.94	–0.79	0.96	–0.71	–0.61	1.00	–0.30	0.16	0.11	0.44	–0.20	–0.40	–0.19	0.10	0.93
x_9	0.00	0.00	–0.46	0.00	–0.39	0.69	0.28	–0.30	1.00	–0.61	–0.18	–0.03	0.35	0.92	0.48	–0.94	–0.11
x_{10}	0.00	0.00	0.15	–0.39	0.20	–0.02	–0.08	0.16	–0.61	1.00	0.87	0.70	–0.95	–0.38	0.12	0.42	–0.20
x_{11}	0.00	0.00	–0.06	–0.49	0.04	0.32	0.18	0.11	–0.18	0.87	1.00	0.83	–0.93	–0.01	0.26	0.01	–0.23
x_{12}	0.00	0.00	0.32	–0.86	0.44	0.17	–0.31	0.44	–0.03	0.70	0.83	1.00	–0.88	0.17	0.54	–0.26	0.17
x_{13}	0.00	0.00	–0.13	0.58	–0.23	–0.16	0.13	–0.20	0.35	–0.95	–0.93	–0.88	1.00	0.09	–0.38	–0.11	0.16
x_{14}	0.00	0.00	–0.49	–0.09	–0.40	0.82	0.13	–0.40	0.92	–0.38	–0.01	0.17	0.09	1.00	0.77	–0.94	–0.28
x_{16}	0.00	0.00	–0.16	–0.40	–0.05	0.61	–0.34	–0.19	0.48	0.12	0.26	0.54	–0.38	0.77	1.00	–0.68	–0.24
x_{17}	0.00	0.00	0.25	0.31	0.15	–0.65	–0.03	0.10	–0.94	0.42	0.01	–0.26	–0.11	–0.94	–0.68	1.00	–0.02
x_{18}	0.00	0.00	0.89	–0.62	0.89	–0.73	–0.60	0.93	–0.11	–0.20	–0.23	0.17	0.16	–0.28	–0.24	–0.02	1.00

Table 4 – Pairwise correlation coefficients of input and output features

r	y_1	y_2	y_3	y_4	y_5	y_6	y_7
x_1	–0.0240	–0.3045	0.0292	0.1242	0.1260	0.1317	0.1132
x_2	–0.3102	–0.2726	0.1651	–0.2519	–0.3563	–0.2464	–0.1601
x_3	0.1198	0.0172	0.2271	0.2153	0.0417	–0.0259	0.1020
x_4	0.0757	0.0193	–0.0486	–0.1540	0.0086	0.1167	–0.0194
x_5	0.0862	0.0104	0.2032	0.2145	0.0337	–0.0461	0.0909
x_6	–0.2262	–0.0467	–0.2617	–0.1657	–0.0708	–0.0532	–0.1225
x_7	–0.1016	–0.0253	–0.1831	–0.2068	0.0060	–0.0075	–0.0460
x_8	0.0404	–0.0113	0.1788	0.1881	0.0445	–0.0828	0.1013
x_9	–0.3200	–0.1464	–0.1721	–0.0998	–0.0484	–0.1651	–0.0282
x_{10}	0.1397	0.1280	–0.0134	0.0441	–0.0052	0.0852	–0.0676
x_{11}	–0.0254	0.0618	–0.1145	–0.0126	–0.0192	–0.0108	–0.0820
x_{12}	–0.0717	0.0230	–0.0507	0.0833	–0.0306	–0.0713	–0.0511
x_{13}	–0.0442	–0.0918	0.0562	–0.0462	0.0249	–0.0267	0.0806
x_{14}	–0.2920	–0.1089	–0.1929	–0.0861	–0.0731	–0.1287	–0.0718
x_{16}	–0.1545	–0.0229	–0.1222	0.0133	–0.0840	–0.0535	–0.0961
x_{17}	0.3191	0.1352	0.1549	0.0456	0.0576	0.1793	0.0345
x_{18}	0.0025	–0.0547	0.1910	0.1750	0.0472	–0.1046	0.1274

DOI 10.15588/1607-3274-2021-1-12

Table 5 – Comparison of neuromodel learning outcomes

Target variable	Number of hidden neurons	Training Time, s	Average error	Independent variables used	Number of hidden neurons	Training Time, s	Average error
y_1	8	4.2400	0.00336	x_2. x_9	4	3.2190	0.00165
y_2	8	5.6440	0.00686	x_1. x_2. x_9. x_{18}	4	4.3480	0.00639
y_3	8	4.3280	0.00641	x_1. x_2. x_8. x_9. x_{18}	4	3.3090	0.00181
y_4	8	5.5470	0.00344	x_1. x_2. x_9. x_{18}	4	4.9320	0.16357
y_5	8	4.4030	0.00316	x_1. x_2. x_9. x_{18}	4	3.3320	0.00191
y_6	8	7.8150	0.72003	x_1. x_2. x_9. x_{18}	4	4.9700	0.84592
y_7	8	4.4480	0.00501	x_1. x_2. x_9. x_{18}	4	4.3400	0.00357

6 DISCUSSION

Let's analyze the input data and sample it. So, $Lev_{smplfctn} = 0$ because the number of input attributes is not large and it is possible to reduce them.

In our case $Lev_{fctr} = 0$, due to the fact that during the factor analysis there were correlation matrices that were poorly determined (their determinant was 0). Factor loads are shown in table 2.

Under the conditions of the problem, measurements were made using special sensors, so $Lev_{accmeas} = 0$. And $Lev_{manag} = 0$ because the experimental system was always under supervision and control.

In table 3 shows the coefficients of paired correlation of input features with each other. It should be noted that x_{15} does not correlate at all.

From the results of table 3, we can conclude that the signs x_1, x_2 and x_{15}, ta do not depend on other signs, and all other signs are strongly related.

Table 4 shows the pairwise correlation coefficients between input and output features. It should be noted that x_{15} does not correlate at all.

From the results of table 4, it can be concluded that the most individually significant for the initial features are the following: x_1, x_2 and x_9.

So we get the coefficients $K_{corrY} = 3$, $K_{ntcorrX} = 3$, $K_{imp} = 8$. The result is the most optimal number of hidden neurons $|N_h| = 4$.

It will be tested and compared the average error values for the complete data and for the data after feature selection and with the optimal number of hidden neurons in the hidden layer defined by (1). The comparison results are shown in Table 5. The Forward–Backward greedy algorithm was used to select features [23].

Based on the results of the initial analysis, the task was assigned to the OS group. After all, the input sample was not excessive, and the risks of human influence were excluded. The only significant complicating factor is the poorly conditioned correlation matrices (their determinant was 0), which was confirmed by factor analysis.

Therefore, at the beginning , neuromodel training was tested using all input data, and since the results of factor analysis revealed 8 most significant variables: for Factor 1 it is $x_3 - x_7$; for Factor 2 it is x_4, x_9; for Factor 3 it is x_1, then at the beginning $|N_h| = 8$.

Analyzing the initial results, it should be noted that there is a fairly large difference between the training time of the model: from 4.24 to 7.82. at the same time, the size of the average error was quite acceptable and ranged from 0.00316 to 0.72003.

The results obtained on the data after reduction showed that the accuracy in some cases (for y_1, y_2, y_3, y_5 and y_7) improved, and the time was significantly reduced, moreover, the time distribution between training iterations was distributed more evenly. However, for a number of cases (y_4, y_6), the accuracy of the model has decreased. This may be due to a slight influence of the eliminated features, which may be of a group nature. Therefore, we can conclude that to feature selection in this case, we should use methods that can select not only individual features, but also groups of features [24, 25].

CONCLUSIONS

The urgent scientific and applied problem of determining the optimal and logically transparent structure of a neuromodel is solved.

The scientific novelty lies in the fact that a method has been developed for selecting the optimal, logically transparent topology of a neural network for modeling complex technical systems using a system of corresponding indicators. Using the problem complexity assessment and a system of indicators, the method allows you to determine the most optimal and logically transparent structure of the neural network.

The practical significance lies in the fact that the developed method allows you to use clear mechanisms to increase the level of logical transparency of the model based on a comprehensive assessment of the problem. When testing the method, the learning time of neuromodels was accelerated by an average of 22%. Therefore, the results of experiments indicate a real simplification of the structure and more rational use of computing resources.

Prospects for further research are the development of methods for structural optimization of pre-synthesized neuromodels in the case of complexity of the OC type problem and the use of feature selection methods that would allow taking into account group information content.

DOI 10.15588/1607-3274-2021-1-12

ACKNOWLEDGEMENTS

The work was carried out with the support of the state budget research projects of the state budget of the National University "Zaporozhzhia Polytechnic" "Intelligent methods and software for diagnostics and non-destructive quality control of military and civilian applications" (state registration number 0119U100360) and "Development of methods and tools for analysis and prediction of dynamic behavior of nonlinear objects".

REFERENCES

1. Goldberg Y. Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies). California , Morgan & Claypool Publishers, 2017, 310 p.
2. Aggarwal C.C. Neural Networks and Deep Learning: A Textbook . Berlin, Springer, 2018, 520 p.
3. Valigi N. Zero to AI, A non-technical, hype-free guide to prospering in the AI. New York, Manning Publications, 2020, 264 p.
4. Artasanchez A. Artificial Intelligence with Python, Your complete guide to building intelligent apps using Python 3.x. Birmingham , Packt Publishing, 2020, 618 p.
5. Oliinyk A., Subbotin S., Leoshchenko S., Ilyashenko M., Myronova N., Mastinovsky Y. Additional training of neuro-fuzzy diagnostic models, *Radio Electronics, Computer Science, Control*, 2018, No. 3, pp. 113–119. DOI: 10.15588/1607-3274-2018-3-12.
6. Leoshchenko S., Oliinyk A., Subbotin S., Zaiko T. Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems, *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 9–12 October 2018, proceedings*. Kharkiv, IEEE, 2018, pp. 411–416. DOI: 10.1109/INFOCOMMST.2018.8632015
7. Iba H. Evolutionary Approach to Machine Learning and Deep Neural Networks, Neuro-Evolution and Gene Regulatory Networks. New York , Springer, 2018, 258 p.
8. Omelianenko I. Hands-On Neuroevolution with Python, Build high-performing artificial neural network architectures using neuroevolution-based algorithms. Birmingham , Packt Publishing, 2019, 368 p.
9. Blokdyk G. Neuroevolution of augmenting topologies, Second Edition. Ohio , 5STARCooks, 2018, 128 p.
10. How is Contrast Encoded in Deep Neural Networks? [Electronic resource]. Access mode, https://arxiv.org/abs/1809.01438
11. Senge P. M. The Fifth Discipline, The Art & Practice of The Learning Organization. New York, Doubleday, 2006, 445 p.
12. Weaver W. Science and complexity, *American Scientist*, 1948, Vol. 36, No. 4, pp. 536–544.
13. Flood R.L. Dealing with Complexity. Berlin, Springer, 1993, 296 p.
14. Spiegelhalter D. The Art of Statistics, How to Learn from Data. New York , Basic Books, 2019, 448 p.
15. Bruce P. Practical Statistics for Data Scientists, 50 Essential Concepts, 1st Edition. California , O'Reilly Media, 2017, 318 p.
16. Finch W. H. Exploratory Factor Analysis [Text] , 1st Edition. California , SAGE Publications, 2019, 144 p.
17. Rencher A. C. Methods of Multivariate Analysis [Text] , 3rd Edition. New Jersey , John Wiley & Sons, 2012. 800 p.
18. Dean A. Design and Analysis of Experiments (Springer Texts in Statistics), 2nd Edition. Berlin, Springer, 2017, 865 p.
19. Sewak M. Deep Reinforcement Learning, Frontiers of Artificial Intelligence / M. Sewak. Berlin, Springer, 2020. 220 p.
20. Calix R. A Deep Learning Algorithms, Transformers, gans, encoders, cnns, rnns, and more. Traverse City, Independently published, 2020, 428 p.
21. Narivs'kyi O.E. Corrosion fracture of platelike heat exchangers, *Materials Science*, 2005, Vol. 41, No. 1, pp. 122–128.
22. Narivs'kyi O.E. Micromechanism of corrosion fracture of the plates of heat exchangers, Materials Science, 2007, Vol. 43, No. 1, pp. 124–132.
23. Kuhn M. Feature Engineering and Selection, A Practical Approach for Predictive Models (Chapman & Hall/CRC Data Science Series). London, Chapman and Hall (CRC Press), 2019, 310 p.
24. Oliinyk A., Subbotin S., Lovkin V., Ilyashenko M., Blagodariov O. Parallel method of big data reduction based on stochastic programming approach, *Radio Electronics, Computer Science, Control*, 2018, № 2, pp. 60–72.
25. Oliinyk A., Leoshchenko S., Lovkin V., Subbotin S., Zaiko T. Parallel data reduction method for complex technical objects and processes, *Dependable Systems, Services and Technologies, (DESSERT'2018) , The 9th IEEE International Conference, Kyiv, 24–27 May, 2018 : proceedings*. Los Alamitos, IEEE, 2018, pp. 528–531.

Received 12.12.2020.
Accepted 18.02.2021.

УДК 004.896

ЗАСТОСУВАННЯ ІНДИКАТОРНОЇ СИСТЕМИ ПРИ МОДЕЛЮВАННІ СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ

Леощенко С. Д. – аспірант кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя Україна.

Субботін С. О. – д-р техн. наук, професор, завідувач кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

Олійник А. О. – канд. техн. наук, доцент кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

Нарівський О. Е. – д-р техн. наук, професор, технічний директор, ТОВ «Укрспецмаш», Бердянськ, Україна.

DOI 10.15588/1607-3274-2021-1-12

АНОТАЦІЯ

Актуальність. Розглянуто задачу визначення оптимальної топології нейромоделі, що відрізняється високим рівнем логічної прозорості при моделюванні складних технічних систем. Об'єктом дослідження є процес застосування індикаторної системи для спрощення та вибору топології нейромоделі.

Мета роботи полягає у розробці та використанні системи індикаторів для визначення рівню складності задачі моделювання та поступовому підборі оптимальної логічно прозорої топології нейромоделі.

Метод. Запропоновано метод для підбору оптимальної, логічно прозорої топології нейронної мережі для моделювання складних технічних систем з використанням системи відповідних індикаторів. На початку метод визначає загальний рівень складності задачі моделювання та використовуючи отриману оцінку визначає спосіб подальшої оптимізації нейромоделі. Потім використовуючи дані про задачу та характеристики вхідних даних метод дозволяє отримати найбільш оптимальну структуру нейронної моделі для подальшого моделювання системи. Метод дозволяє скоротити час навчання та підвищити рівень логічної прозорості нейромоделі, що значно розширює практичне використання таких моделей, без використання нейроеволюційних методів, що можуть бути не виправдано ресурсоємними при ряді задач.

Результати. Розроблений метод реалізовано та досліджено при вирішенні задачі моделювання динаміки піттінгових процесів стальних сплавів. Використання розробленого методу дозволило скоротити час навчання моделі на 22%, в залежності від використовуваних обчислювальних ресурсів. Також метод дозволило підвищити рівень логічної прозорості моделі скоротивши кількість обчислювальних вузлів на 50%, що також свідчить про прискорення та більш раціональне використання ресурсів.

Висновки. Проведені експерименти підтвердили працездатність запропонованого математичного забезпечення і дозволяють рекомендувати його для використання на практиці при проектуванні топологій нейромоделей для подальшого вирішення задач моделювання, діагностування та оцінювання. Перспективи подальших досліджень можуть полягати в розробці методів структурної оптимізації попередньо синтезованих моделей та розробці нових методів відбору інформаційних ознак.

КЛЮЧОВІ СЛОВА: оцінка складності, система індикаторів, моделювання, нейромодель, вибірка, навчання, помилка, градієнт.

УДК 004.896

СИНТЕЗ И ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ С ВЕРОЯТНОСТНЫМ КОДИРОВАНИЕМ СТРУКТУРЫ

Леощенко С. Д. – аспирант кафедры программных средств Национального университета «Запорожская политехника», Запорожье Украина.

Субботин С. А. – д-р. техн. наук, профессор, заведующий кафедрой программных средств Национального университета «Запорожская политехника», Запорожье Украина.

Олейник А. А. – канд. техн. наук, доцент, доцент кафедры программных средств Национального университета «Запорожская Политехника», Запорожье Украина.

Наривский А. Э. – д-р техн. наук, профессор, технический директор ООО «Укрспецмаш», Бердянсь, Украина.

АННОТАЦИЯ

Актуальность. Рассмотрена задача определения оптимальной топологии нейромоделей, которая отличается высоким уровнем логической прозрачности при моделировании сложных технических систем. Объектом исследования является процесс применения индикаторной системы для упрощения и выбора топологии нейромодели.

Цель работы заключается в разработке и использовании системы индикаторов для определения уровня сложности задачи моделирования и постепенном подборе оптимальной логически прозрачной топологии нейромоделей.

Метод. Предложен метод для подбора оптимальной, логично прозрачной топологии нейронной сети для моделирования сложных технических систем с использованием системы соответствующих индикаторов. В начале метод определяет общий уровень сложности задачи моделирования и используя полученную оценку определяет способ дальнейшей оптимизации нейромоделей. Затем используя данные о задаче и характеристики входных данных метод позволяет получить наиболее оптимальную структуру нейронной модели для дальнейшего моделирования системы. Метод позволяет сократить время обучения и повысить уровень логической прозрачности нейромодели, что значительно расширяет практическое использование таких моделей, без использования нейроэволюционных методов, которые могут быть не оправданно ресурсоемкими при ряде задач.

Результаты. Разработанный метод реализован и исследован при решении задачи моделирования динамики питтинговых процессов стальных сплавов. Использование разработанного метода позволило сократить время обучения модели на 22%, в зависимости от используемых вычислительных ресурсов. Также метод позволил повысить уровень логической прозрачности модели сократив количество вычислительных узлов на 50%, что также свидетельствует об ускорении и более рациональном использовании ресурсов.

Выводы. Проведенные эксперименты подтвердили работоспособность предложенного математического обеспечения и позволяют рекомендовать его для использования на практике при проектировании топологии нейромоделей для дальнейшего решения задач моделирования, диагностирования и оценивания. Перспективы дальнейших исследований могут заключаться в разработке методов структурной оптимизации предварительно синтезированных моделей и разработке новых методов отбора информативных признаков.

КЛЮЧЕВЫЕ СЛОВА: оценка сложности, система индикаторов, моделирование, нейромодель, выборка, обучение, ошибка, градиент.

DOI 10.15588/1607-3274-2021-1-12

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Goldberg Y. Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies) / Y. Goldberg, G. Hirst. – California : Morgan & Claypool Publishers, 2017. – 310 p.
2. Aggarwal C. C. Neural Networks and Deep Learning: A Textbook / C. C. Aggarwal. – Berlin : Springer, 2018. – 520 p.
3. Valigi N. Zero to AI: A non-technical, hype-free guide to prospering in the AI era / N. Valigi, G. Mauro. – New York : Manning Publications, 2020. – 264 p.
4. Artasanchez A. Artificial Intelligence with Python: Your complete guide to building intelligent apps using Python 3.x / A. Artasanchez, P. Joshi. – Birmingham : Packt Publishing, 2020. – 618 p.
5. Additional training of neuro-fuzzy diagnostic models / [A. Oliinyk. S. Subbotin, S. Leoshchenko et al] // Radio Electronics, Computer Science, Control. – 2018. – № 3. – P. 113–119. DOI: 10.15588/1607-3274-2018-3-12.
6. Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems / [S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko] // 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 9–12 October 2018 : proceedings. – Kharkiv: IEEE, 2018. – P. 411–416. DOI: 10.1109/INFOCOMMST.2018.8632015
7. Iba H. Evolutionary Approach to Machine Learning and Deep Neural Networks: Neuro-Evolution and Gene Regulatory Networks / H. Iba. – New York : Springer, 2018. – 258 p.
8. Omelianenko I. Hands-On Neuroevolution with Python: Build high-performing artificial neural network architectures using neuroevolution-based algorithms / I. Omelianenko. – Birmingham : Packt Publishing, 2019. – 368 p.
9. Blokdyk G. Neuroevolution of augmenting topologies: Second Edition / G. Blokdyk. – Ohio : 5STARCooks, 2018. – 128 p.
10. How is Contrast Encoded in Deep Neural Networks? / A. Akbarinia, K. R. Gegenfurtner [Electronic resource]. – Access mode: https://arxiv.org/abs/1809.01438
11. Senge P. M. The Fifth Discipline: The Art & Practice of The Learning Organization / P. M. Senge. – New York : Doubleday, 2006. – 445 p.
12. Weaver W. Science and complexity / W. Weaver // American Scientist. – 1948. – Vol. 36, No. 4. – P. 536–544.
13. Flood R.L. Dealing with Complexity / R. L. Flood, E. R. Carson. – Berlin : Springer, 1993. – 296 p.
14. Spiegelhalter D. The Art of Statistics: How to Learn from Data / D. Spiegelhalter. – New York : Basic Books, 2019. – 448 p.
15. Bruce P. Practical Statistics for Data Scientists: 50 Essential Concepts: 1st Edition / P. Bruce, A. Bruce. – California : O'Reilly Media, 2017. – 318 p.
16. Finch W. H. Exploratory Factor Analysis: 1st Edition / W. Holmes Finch. – California : SAGE Publications, 2019. – 144 p.
17. Rencher A. C. Methods of Multivariate Analysis: 3rd Edition / A C. Rencher, W. F. Christensen. – New Jersey : John Wiley & Sons, 2012. – 800 p.
18. Dean A. Design and Analysis of Experiments (Springer Texts in Statistics): 2nd Edition / A. Dean, D. Voss, D. Draguljić. – Berlin : Springer, 2017. – 865 p.
19. Sewak M. Deep Reinforcement Learning: Frontiers of Artificial Intelligence / M. Sewak. – Berlin : Springer, 2020. – 220 p.
20. Calix R. A Deep Learning Algorithms: Transformers, gans, encoders, cnns, rnns, and more / R. A Calix. – Traverse City : Independently published, 2020. – 428 p.
21. Narivs'kyi O. E. Corrosion fracture of platelike heat exchangers / O. E. Narivs'kyi // Materials Science. – 2005. – 41, № 1. – P. 122–128.
22. Narivs'kyi O.E. Micromechanism of corrosion fracture of the plates of heat exchangers / O. E. Narivs'kyi // Materials Science. – 2007. – 43, №1. – p. 124–132.
23. Kuhn M. Feature Engineering and Selection: A Practical Approach for Predictive Models (Chapman & Hall/CRC Data Science Series) / M. Kuhn, K. Johnson. – London: Chapman and Hall (CRC Press), 2019. – 310 p.
24. Parallel method of big data reduction based on stochastic programming approach / [A. Oliinyk, S. Subbotin, V. Lovkin, M. Ilyashenko, O. Blagodariov] // Radio Electronics, Computer Science, Control. – 2018. – № 2. – P. 60–72.
25. Parallel data reduction method for complex technical objects and processes / [A. Oliinyk, S. Leoshchenko, V. Lovkin et al.] // Dependable Systems, Services and Technologies, (DESSERT'2018) : The 9th IEEE International Conference, Kyiv, 24–27 May 2018 : proceedings. – Los Alamitos: IEEE, 2018. – P. 528–531.

DOI 10.15588/1607-3274-2021-1-12

e-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2021. № 1
p-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2021. № 1

UDC 519.876.2:336

METHOD OF SPECTRAL CLUSTERING OF PAYMENTS AND RAW MATERIALS SUPPLY FOR THE COMPLIANCE AUDIT PLANNING

Neskorodieva T. V. – PhD, Associate Professor, Head of the Department of Computer Science and Information Technology, Vasyl' Stus Donetsk National University, Vinnytsia, Ukraine.

Fedorov E. E. – Dr. Sc., Associate Professor, Professor of the Department of Robotics and Specialized Computer Systems, Cherkasy State Technological University, Cherkasy, Ukraine.

ABSTRACT

Context. The analytical procedures used in the audit are currently based on data mining techniques. The work solves the problem of increasing the efficiency and effectiveness of analytical audit procedures by clustering based on spectral decomposition. The object of the research is the process of auditing the compliance of payment and supply sequences for raw materials.

Objective. The aim of the work is to increase the effectiveness and efficiency of the audit due to the method of spectral clustering of sequences of payment and supply of raw materials while automating procedures for checking their compliance.

Method. The vectors of features are generated for the objects of the sequences of payment and supply of raw materials, which are then used in the proposed method. The created method improves the traditional spectral clustering method by automatically determining the number of clusters based on the explained and sample variance rule; automatic determination of the scale parameter based on local scaling (the rule of K-nearest neighbors is used); resistance to noise and random outliers by replacing the k-means method with a modified PAM method, i.e. replacing centroid clustering with medoid clustering. As in the traditional approach, the data can be sparse, and the clusters can have different shapes and sizes. The characteristics of evaluating the quality of spectral clustering are selected.

Results. The proposed spectral clustering method was implemented in the MATLAB package. The results obtained made it possible to study the dependence of the parameter values on the quality of clustering.

Conclusions. The experiments carried out have confirmed the efficiency of the proposed method and allow us to recommend it for practical use in solving audit problems. Prospects for further research may lie in the creation of intelligent parallel and distributed computer systems for general and special purposes, which use the proposed method for segmentation, machine learning and pattern recognition tasks.

KEYWORDS: audit planning, clustering, spectral decomposition, medoids, sequence of payment and supply of raw materials.

ABBREVIATIONS

NJW is a Ng, Jordan, Weiss method;

PAM is the partitioning around medoids;

EM is an expectation-maximization;

DBSCAN is a density-based spatial clustering of applications with noise;

OPTICS is an ordering points to identify the clustering structure;

DIANA is a divisive analysis;

SOM is a self-organizing map;

ART is a adaptive resonance theory;

TP is a true positive;

TN is a true negative;

FP is a false positive;

FN is a false negative.

NOMENCLATURE

A is a set of clustering objects;

a_i is an i-th object of clustering;

n is a number of objects of clustering;

X is an set of feature vectors from the space R^q;

x_i is a feature vector of i-th object of clustering from the space R^q;

q is a number of features in feature vector x_i;

$\breve{X}$ is a set of K-nearest feature vectors;

$\breve{x}_i$ is a feature vector of K-nearest to feature vector x_i;

δ is a threshold for determining the number of clusters;

K is a number of nearest neighbors;

σ_i is a scale parameter for the i-th feature vector;

$\mathbf{S}$ is a symmetric similarity matrix;

$\mathbf{D}$ is a diagonal degree matrix;

$\mathbf{L}$ is a normalized symmetric Laplace matrix;

$\mathbf{I}$ is a unit matrix;

λ_i is an i-th eigenvalue;

$\mathbf{w}_i$ is an i-th eigenvector;

c is a number of clusters;

R^2 is a coefficient of determination;

$\mathbf{V}$ is a principal component matrix;

$\tilde{X}$ is a set of feature vectors from the space R^c;

$\tilde{x}_i$ is an i-th feature vector from the space R^c;

$\check{X}$ is a set of feature vectors from the space R^c, which not corresponding to medoids;

A_k is a k-th cluster;

$\mathbf{\Lambda}$ is a set of indicator functions;

$\chi_{A_k}(\cdot)$ is an indicator function A_k (returns 1 or 0 depending on the belonging of the object to the k-th cluster);

D_{ik} is a square of distance between i-th object and medoid of k-th cluster;

$F(\cdot)$ is a target function;

DOI 10.15588/1607-3274-2021-1-13

y^* is a best target function value;

y is a target function value;

M is a set of cluster centroids;

$\mathbf{m}_k$ is a centroid of k-th cluster for the space R^q;

$\tilde{M}$ is a set of medoids of cluster;

$\tilde{\mathbf{m}}_i$ is a medoid of k-th cluster for the space R^c;

$\hat{\mathbf{m}}$ is a preserved medoid for space R^c;

$N(0,1)$ is a function, that returns standard normal distributed random number;

v^2 is a variance of Gaussian additive noise;

Accuracy is an accuracy;

Precision is a precision;

Recall is a recall;

F is a balanced F-measure;

θ_d is a types set of paid raw materials;

θ_k is a types set of raw materials obtained;

s_d is a type of paid raw materials;

s_k is a type of raw material received;

δ_{s_d} is a cost of paid raw material of type s_d;

v_{s_k} is a number of received raw material of type s_k.

INTRODUCTION

The analytical procedures used in the audit are currently based on data mining techniques [1, 2]. In an automated audit system, the task of auditing expenses at the top level is decomposed into tasks of checking the sequence of displaying data of the middle level. First of which – display is paid-received. This is a mapping of the multidimensional data of payment for raw materials to suppliers to/in the set of multidimensional data for the delivery of raw materials. At the lower level, if there are no violations in accounting, this mapping should be one-to-one. In order to reduce the volume of checks at the lower level, the audit system analyzes the aggregated indicators of payment and delivery at the middle level or formed sets (clusters) of multidimensional data of the lower level. Also, when designing an IT audit, the goal is to automate the analysis to form recommended solutions. According to the method of generalized set mapping, at the middle level, generalized properties of data sets (condensation points, isolated points) are analyzed, that is, the density structure of each of the sets is determined, and then they are compared.

For analysis, pay and delivery sequence data can be aggregated over quantization periods:

1) for all suppliers;
2) by the nomenclature of raw materials.

Analysis of data on payment and supply of raw materials is carried out to form recommended solutions for the following audit tasks.

1. The task of the external audit is to check the completeness of accounting for settlements with suppliers.
2. Tasks of internal audit to verify compliance with contractual policies. The contractual policy of the enterprise is a set of rules characterizing the delivery time after payment, the nomenclature of raw materials, technical or physical characteristics, prices (discounts).
3. The task of the internal audit of pricing policy when concluding contracts (identifying a significant share of unfavorable contracts, which are features of "kick-backs" when concluding them).
4. The task of internal audit of receivables from suppliers of raw materials in terms of timing and amounts.

Clustering methods are used to audit the compliance of the sequence of payments for raw materials and the sequence of deliveries of raw materials at the stage of identifying characteristic properties.

Traditional clustering methods are:

1. Partition-based (partitioning-based) or center-based methods (e.g., methods k-means [3], PAM (k-medoids) [4], FCM [5]).
2. Mixture model or distribution-based or model-based methods (e.g., EM [5]).
3. Density-based methods (e.g., methods DBSCAN [6], OPTICS [7]).
4. Hierarchical methods:

– agglomerative or ascending (bottom up) (e.g. centroid communication methods, Vard, unit connection, full connection, group secondary) [8];

– divisive or descending (top down) (e.g., methods DIANA) [9].

Clustering methods can also be based on metaheuristics [10, 11] and artificial neural networks (e.g., SOM, ART) [12].

Object of study. Audit process for compliance with payment sequences and raw materials supply.

Subject of study. Spectral clustering method for auditing sequences of payment and supply of raw materials.

The aim of the work is to increase the effectiveness and efficiency of the audit by automating the analysis of data from sets of parallel-sequential operations of payment and supply of raw materials based on the spectral clustering method.

To achieve this goal, it is necessary to solve the following tasks:

1. Generate feature vectors for objects of sequences of payment and supply of raw materials.
2. Create a method for spectral clustering of sequences of payment and supply of raw materials.
3. Select characteristics for assessing the quality of spectral clustering.
4. Conduct a numerical study of the proposed spectral clustering method.

1 PROBLEM STATEMENT

The problem of increasing the efficiency of audit based on the method of spectral clustering of sequences of payment and supply of raw materials is presented as the problem of finding such a partition of the set of clustering

DOI 10.15588/1607-3274-2021-1-13

objects $A = \{a_1, ..., a_n\}$, represented by a set of feature vectors $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, per cluster $A_1, ..., A_c$ through a variety of indicator functions $\Lambda = \{\chi_{A_1}(\cdot), ..., \chi_{A_c}(\cdot)\}$, and with a set of cluster centroids $M = \{\mathbf{m}_1, ..., \mathbf{m}_c\}$, at which

$$\mathrm{F} = \sum_{i=1}^{n} \sum_{k=1}^{c} \chi_{A_k}(a_i) \left\| \mathbf{x}_i - \mathbf{m}_k \right\|^2 \to \min_{\Lambda, M}.$$

2 REVIEW OF THE LITERATURE

Existing clustering methods have one or more of the following disadvantages [6, 7]:

– have high computational complexity;

– do not allow the emission of noise and random emissions;

– clusters cannot have different shapes and sizes;

– require specifying the number of clusters;

– require the definition of parameter values.

In this regard, it is relevant to create a clustering method that will eliminate the indicated disadvantages.

One of these methods is spectral clustering [13, 14], which has already found application in the segmentation of signals of different physical nature [15]. Since initially the spectral clustering methods did not provide for the procedure for automating the determination of the parameters and the number of clusters, an attempt is being made to eliminate this drawback. [16], which will allow them to be used in IT audit of enterprises with different characteristics.

3 MATERIALS AND METHODS

Let's start by solving the first task formation of feature vectors for objects of sequences of payment and supply of raw materials.

The attributes for the objects of the sequence of payment and supply of raw materials are formed on the basis of the accounting variables of the lower level (Table 1), taking into account the possible options for generalizing their values at the average level for the periods of quantization. Clustering objects of payment (supply) for each supplier with which a long-term supply agreement is in force during the year for which the audit is carried out. Feature vector $\mathbf{x}_i = (x_{i1}, ..., x_{iq})$ objects of payment form indicators of the cost of paid raw materials δ_{s_d} by types $s_d \in \Theta_d$. Features vector $\mathbf{x}_i = (x_{i1}, ..., x_{iq})$ delivery objects form indicators of the amount of paid raw materials v_{s_k} by types $s_k \in \Theta_k$.

To assess the dimension of the vector of attributes and the number of objects of analysis, an analysis of the nomenclature of purchases of raw materials (components) of large engineering enterprises. So, based on this analysis, we can conclude that the sections of the nomenclature are on average from 8 to 12, the number of groups in each section is from 2–10. Analyzing the homogeneity of the procurement nomenclature, we can conclude that for continuous operation the plant can have long-term contracts with suppliers in the amount of 50 до 100.

Clustering will make it possible to form subsets of payment and supply operations that are similar in terms of the features highlighted above, which will allow analyzing the set of operations when comparing and reducing the computational complexity of solving the matching problem.

To form the rules for matching the sequences of payments and deliveries after clustering, it is necessary to select the rules of relationships. Based on the analysis of the terms of payment agreements and the supply of raw materials, the rules for recording these transactions in the system, the following rules were identified:

1) Delivery operations are carried out after payment, in accordance with the contractual policy of the enterprise in accordance with payment orders.

2) Delivery under a new payment order is not carried out until the previous one is closed.

3) Low-level delivery data that corresponds to one payment order is aggregated before clustering.

Let's move on to solving the second problem – a method ***creating*** for spectral clustering of sequences of payment and supply of raw materials (Fig. 1).

1. Specifying multiple clustering objects $A = \{a_i\}$, $i \in \overline{1,n}$. Specifying a set of feature vectors $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, and each object a_i corresponds to the feature vector $\mathbf{x}_i = (x_{i1}, ..., x_{iq})$. Setting the threshold for determining the number of clusters δ, $0 < \delta < 1$. Setting the number of nearest neighbors K.

2. Creation of a set of-nearest feature vectors $\breve{X} = \{\breve{\mathbf{x}}_i\}$, such that for each feature vector $\mathbf{x}_i$ K-nearest to it is the feature vector $\breve{\mathbf{x}}_i$.

3. Calculating scale options based on local scaling:

$$\sigma_i = \left\| \mathbf{x}_i - \breve{\mathbf{x}}_i \right\|, \ i \in \overline{1,n}.$$

4. Calculation of the symmetric similarity matrix

$$\mathbf{S} = [s_{ij}], \ i, j \in \overline{1,n},$$

$$s_{ij} = \begin{cases} \exp\left(-\dfrac{\left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2}{2\sigma_i \sigma_j} \right), & i \neq j \\ 0, & i = j \end{cases}.$$

5. Calculating the diagonal degree matrix:

$$\mathbf{D} = diag(d_1, ..., d_n),$$

$$d_i = \sum_{j=1}^{n} s_{ij}.$$

6. Calculation of the normalized symmetric Laplace matrix:

$$\mathbf{L} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{S})\mathbf{D}^{-1/2}.$$

DOI 10.15588/1607-3274-2021-1-13

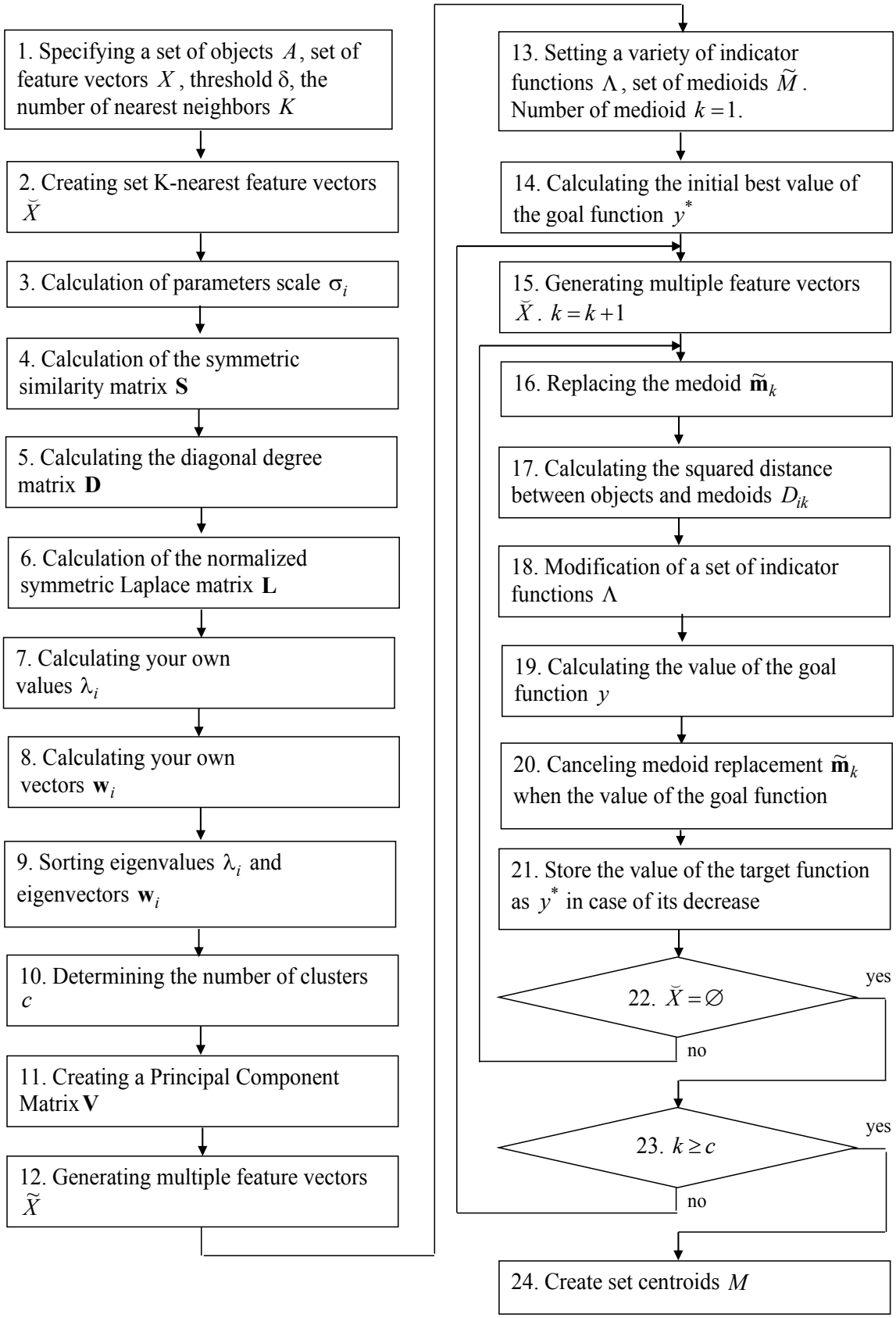

Figure 1 – The structure of spectral clustering of sequences of payment and supply of raw materials

DOI 10.15588/1607-3274-2021-1-13

7. Calculation of eigenvalues $\lambda_i, i \in \overline{1,n}$, matrix $\mathbf{L}$ as roots of the characteristic equation $\det(\mathbf{L}-\lambda\mathbf{I})=0$.

8. Computing eigenvectors $\mathbf{w}_i, i \in \overline{1,n}$, dimensions n from the equation $(\mathbf{L}-\lambda_i\mathbf{I})\mathbf{w}_i=0$, which is obtained from the relation $\mathbf{L}\mathbf{w}_i=\lambda_i\mathbf{w}_i$.

9. Sorting eigenvalues λ_i and eigenvectors $\mathbf{w}_i$ in descending eigenvalues λ_i.

10. Determining the number of clusters c as the number of selected eigenvalues and eigenvectors by means of a rule based on the coefficient of determination

$$0 < R^2 < \delta,\ R^2 = \frac{\sum_{i=1}^{c}\lambda_i}{\sum_{i=1}^{n}\lambda_i},$$

at that $\sum_{i=1}^{c}\lambda_i$ interpreted as a fraction of the variance explained, and $\sum_{i=1}^{n}\lambda_i$ interpreted as a fraction of the total variance.

11. Creating a Principal Component Matrix $\mathbf{V}=[v_{ij}]$ dimensions $n \times c$, whose columns are selected eigenvectors $\mathbf{w}_i$, which have eigenvalues λ_i that are the greatest.

12. Generating multiple feature vectors $\tilde{X}=\{\tilde{\mathbf{x}}_1,...,\tilde{\mathbf{x}}_n\}$, and each object a_i corresponds to the feature vector $\tilde{\mathbf{x}}_i=(\tilde{x}_{i1},...,\tilde{x}_{ic})$

$$\tilde{x}_{ij}=\frac{v_{ij}}{\sqrt{\sum_{j=1}^{c}\left(v_{ij}\right)^2}},\ i \in \overline{1,n},\ j \in \overline{1,c}.$$

13. Setting randomly the initial partition of a set of clustering objects $A=\{a_1,...,a_n\}$, represented by a set of feature vectors $\tilde{X}=\{\tilde{\mathbf{x}}_1,...,\tilde{\mathbf{x}}_n\}$, per clusters $A_1,...,A_c$ through a variety of indicator functions $\Lambda=\{\chi_{A_1}(\cdot),...,\chi_{A_c}(\cdot)\}$ (return 1 or 0 depending on whether the object belongs to the *k*-th cluster). From set $\tilde{X}$ a set of medoids are randomly selected $\tilde{M}=\{\tilde{\mathbf{m}}_1,...,\tilde{\mathbf{m}}_c\}$. Medoid number is $k=0$.

14. Calculating the initial best value of the goal function

$$y^*=\sum_{i=1}^{n}\sum_{k=1}^{c}\chi_{A_k}(a_i)\left\|\tilde{\mathbf{x}}_i-\tilde{\mathbf{m}}_k\right\|^2.$$

15. Creation of a set of vectors of features that do not correspond to medoids $\breve{X}=\tilde{X}\setminus\tilde{M}$. Incrementing the medoid number, i.e. $k=k+1$

16. Replacing the medoid $\tilde{\mathbf{m}}_k$.

16.1. Saving the replaceable medoid $\tilde{\mathbf{m}}_k$, i.e. $\hat{\mathbf{m}}=\tilde{\mathbf{m}}_k$.

16.2. Extract from the set $\breve{X}$ next feature vector and assigning it the vector $\tilde{\mathbf{m}}_k$.

17. Calculating the squared distance between objects and medoids

$$D_{ik}=\left\|\tilde{\mathbf{x}}_i-\tilde{\mathbf{m}}_k\right\|^2,\ i \in \overline{1,n},\ k \in \overline{1,c}.$$

18. Modification of a set of indicator functions

$$\chi_{A_k}(a_i)=\begin{cases}1, & k=\arg\min\limits_{j\in\overline{1,c}} D_{ij}\\ 0, & k\neq\arg\min\limits_{j\in\overline{1,c}} D_{ij}\end{cases},\ i \in \overline{1,n},\ k \in \overline{1,c}.$$

The following conditions must be met for indicator functions

$$\sum_{k=1}^{c}\chi_{A_k}(a_i)=1,\ i \in \overline{1,n},$$

$$\sum_{i=1}^{n}\chi_{A_k}(a_i)>0,\ k \in \overline{1,c},$$

$$\chi_{A_k}(a_i) \in \{0,1\},\ k \in \overline{1,c},\ i \in \overline{1,n}.$$

19. Calculating the value of the goal function

$$y=\sum_{i=1}^{n}\sum_{k=1}^{c}\chi_{A_k}(a_i)\left\|\tilde{\mathbf{x}}_i-\tilde{\mathbf{m}}_k\right\|^2.$$

20. Cancellation of medoid replacement in case of increasing the value of the goal function

$$\text{if } y > y^*, \text{ then } \tilde{\mathbf{m}}_k=\hat{\mathbf{m}}.$$

21. Keeping the value of the target function as best as it increases.

$$\text{if } y < y^*, \text{ then } y^*=y.$$

22. If set $\breve{X}$ not empty then go to step 16.

23. If not all medoids are viewed, i.e. $k < c$, then go to step 15.

24. Creating multiple centroids

DOI 10.15588/1607-3274-2021-1-13

$$M = \{\mathbf{m}_1, ..., \mathbf{m}_c\},$$

$$m_{kj} = \frac{\sum_{i=1}^{n} \chi_{A_k}(a_i) x_{ij}}{\sum_{i=1}^{n} \chi_{A_k}(a_i)}, \; k \in \overline{1,c}, \; j \in \overline{1,q}.$$

The result of the method is a set of indicator functions $\Lambda = \{\chi_{A_1}(\cdot), ..., \chi_{A_c}(\cdot)\}$ and set of cluster centroids $M = \{\mathbf{m}_1, ..., \mathbf{m}_c\}$.

Fig. 1 shows the structure of spectral clustering of sequences of payment and supply of raw materials.

Let's move on to solving the third task – characteristics selecting for assessing the quality of spectral clustering. In the work, the following characteristics were chosen for assessing the quality of spectral clustering:

In the work, the following characteristics were chosen for assessing the quality of spectral clustering:

1. $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$,
2. $Precision = \frac{TP}{TP + FP}$,
3. $Recall = \frac{TP}{TP + FN}$,
4. Balanced F-measure

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}.$$

4 EXPERIMENTS

A numerical study of the proposed spectral clustering method was carried out in the package MATLAB.

The work used a standard database of handwritten numbers digit1000 (http://www.stat.washington.edu/spectral/datasets.html). There were 100 objects for each of the 10 digits, i.e. number of clustering objects $n = 1000$. For each object, the length of the feature vector was $q = 64$. Objects were noisy with additive Gaussian noise, i.e. added noise component $v^2 N(0,1)$, $v^2 = 0.05$. Threshold for determining the number of clusters $\delta = 0.05$, number of nearest neighbors $K = 7$.

5 RESULTS

The function reflecting the dependence of the determination coefficient on the number of clusters is presented in the form

$$R^2(c) = \frac{\sum_{i=1}^{c} \lambda_i}{\sum_{i=1}^{n} \lambda_i}.$$

Function part satisfying inequality $0 < R^2(c) < \delta$, shown in Fig. 2.

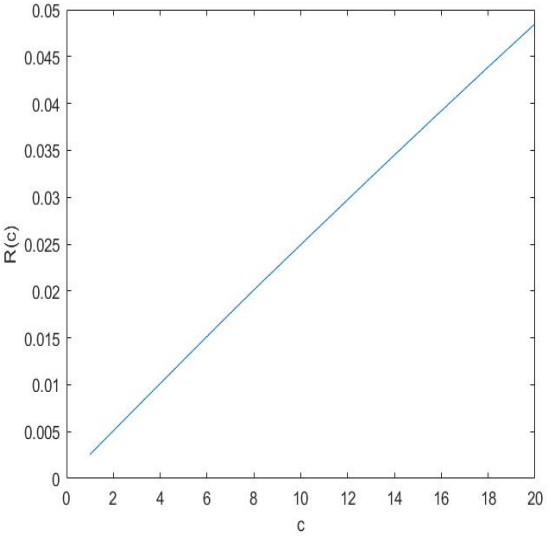

Figure 2 – Function reflecting the dependence of the coefficient of determination on the number of clusters

The dependence (Fig. 2) of the determination coefficient on the number of clusters shows that the determination coefficient increases with an increase in the number of clusters.

The results of comparison of the qualitative characteristics of the proposed method with the NJW method described in [13] are presented in Table 1.

Table 1 – Comparison of the qualitative characteristics of the proposed spectral clustering method with the existing NJW method

№	Method characteristics	Spectral clustering methods	
		This method	NJW
1	automatic determination of the number of clusters	+	–
2	automatic determination of the scale parameter	+	–
3	resistance to noise and accidental emissions	+	–
4	data can be sparse	+	+
5	clusters can have different shapes and sizes	+	+

The results of comparison of the quantitative characteristics of the proposed method with the NJW method described in [13] are presented in table 2.

Table 2 – Comparison of the quantitative characteristics of the proposed spectral clustering method with the existing NJW method

№ p/p	Method characteristics	Spectral clustering methods	
		This method	NJW
1	Accuracy	0.97	0.82
2	Precision	0.97	0.73
3	Completeness	0.97	0.82
4	Balanced F-measure	0.96	0.76

DOI 10.15588/1607-3274-2021-1-13

6 DISCUSSION

The selected values of the parameters of the proposed spectral clustering method provide high accuracy of clustering.

Traditional NJW Spectral Clustering Method [13]:

– requires specifying the number of clusters;

– scale parameter required;

– is not robust to noise and random outbursts (instead of the k-means method, a modified PAM method is used, i.e. centroid clustering is replaced by medoid clustering).

The proposed method eliminates the indicated disadvantages (table 2).

In terms of accuracy, precision, completeness, balanced F-measure, the proposed method is more effective than the NJW method (table 2).

CONCLUSIONS

The urgent task of increasing the effectiveness and efficiency of the audit was solved by creating a method of spectral clustering of sequences of payment and supply of raw materials.

The scientific novelty of obtained results is that the method of spectral clustering. It improves the quality of clustering due to:

– automatic determination of the number of clusters based on the explained and sample variance rule;

– automatic scaling parameter based on local scaling;

– resistance to noise and random outliers by replacing the k-means method with a modified PAM method, i.e. replacing centroid clustering with medoid clustering.

The practical significance of obtained results is that the proposed method makes possible to expand the scope of clustering methods based on spectral decomposition, which is confirmed by its adaptation for the audit task, and contributes to increasing the efficiency of intelligent computer systems for general and special purposes.

Prospects for further research are the study of the proposed method for a wide class of artificial intelligence tasks, as well as the creation of a method for matching payment and delivery sequences after clustering to solve audit problems.

ACKNOWLEDGEMENTS

The research was carried out in accordance with the priority direction of the development of science and technology in Ukraine “Information and communication technologies” and contain some results of research “Methods, models for the processing of intellectual, information technologies for highly efficient computational and local control systems in problem-based systems” (state registration number 0106U004501) and “Development of models and methods in biometric identification of people” (state registration number 0119U002860).

REFERENCES

1. Neskorodieva T., Fedorov E., Izonin I. Forecast Method for Audit Data Analysis by Modified Liquid State Machine, *The 1st International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2020), Khmelnytskyi, Ukraine, 10–12 June, 2020: proceedings,* 2020, CEUR-WS, Vol. 2623, pp. 25–35.
2. Neskorodieva T., Fedorov E. Method for Automatic Analysis of Compliance of Expenses Data and the Enterprise Income by Neural Network Model of Forecast, *The 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT&DS-2020), Lviv-Shatsk, Ukraine, 2–3 June, 2020: proceedings. CEUR-WS, Volume I: Main Conference.* 2020, Vol. 2631, pp. 145–158.
3. Brusco M. J., Shireman E , Steinley D. A Comparison of Latent Class, K-means, and K-median Methods for Clustering Dichotomous Data, *Psychological Methods*, 2017, Vol. 22 (3), pp. 563–580. DOI: 10.1037/met0000095.
4. Bezdek J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. New York, Plenum Press, 1981, 256 p. DOI: 10.1007/978-1-4757-0450-1.
5. Fu Z., Wang L. Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm, *Multimedia and Signal Processing,* 2012, pp. 61–66. DOI: 10.1007/978-3-642-35286-7_9.
6. Ester M., Kriegel H.-P., Sander J., Xu X. Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Second International Conference on Knowledge Discovery and Data Mining (KDD), Portland, Oregon, August 2–4, 1996: proceedings*. AAAI Press, pp. 226–231.
7. Ankerst M., Breunig M. M., Kriegel H.-P., Sander J. OPTICS: Ordering Points to Identify the Clustering Structure, *International Conference on Management of Data and Symposium on Principles of Database Systems. Philadelphia, Pennsylvania, USA, May, 1999: proceedings, Association for Computing Machinery*. New York, NY, United States, 1999, pp. 49–60.
8. Mirkin B. G. Clustering for Data Mining: A Data Recovery Approach. Boca Raton, FL, CRC Press, 2005, 277 p. DOI: 10.1201/9781420034912.
9. Aggarwal C. C., Reddy C. K. Data Clustering. Boca Raton, FL: CRC Press, 2014, 620 p. DOI:10.1201/9781315373515.
10. Subbotin S., Oliinyk A. , Levashenko V. , Zaitseva E. Diagnostic Rule Mining Based on Artificial Immune System for a Case of Uneven Distribution of Classes in Sample, *Communications*, 2016, Vol. 3, pp. 3–11.
11. Fedorov E., Utkina T., Nechyporenko O., Korpan Y. Development of technique for face detection in image based on binarization, scaling and segmentation methods, *Eastern-European Journal of Enterprise Technologies*, Vol. 1/9, 2020, pp. 23–31. DOI: 10.15587/1729-4061.2020.195369.
12. He J., Tan A.-H., Tan Ch.-L. Modified ART 2A Growing Network Capable of Generating a Fixed Number of Nodes, *IEEE Transactions on Neural Networks,* 2004, Vol. 15(3), pp. 728–37. DOI: 10.1109/TNN.2004.826220.
13. Andrew Y. Ng., Jordan I. M., Weiss Y. On spectral clustering: Analysis and an algorithm, *In Advances in neural information processing systems*, 2002, pp. 849–856.
14. Ulrike V. L. A tutorial on spectral clustering, *Statistics and computing*, Vol. 17(4): 2007, pp. 395–416. DOI: 10.1007/s11222-007-9033-z.
15. Fabien L., Schnörr C. Spectral clustering of linear subspaces for motion segmentation, *12th International Conference on Computer Vision (ICCV’09), Sep 2009, Kyoto*. Japan, proceedings, IEEE, pages to-appear, 2009. DOI: 10.1109/iccv.2009.5459173.
16. Tao Xiang, Shaogang G. Spectral clustering with eigenvector selection, *Pattern Recognition*, 2008, Vol. 41(3), pp. 1012–1029. DOI: 10.1016/j.patcog.2007.07.023.

DOI 10.15588/1607-3274-2021-1-13

17. Tao Xiang, Shaogang G. Spectral clustering with eigenvector selection, *Pattern Recognition,* 2008, Vol. 41(3), pp. 1012–1029. DOI: 10.1016/j.patcog.2007.07.023.
18. Feng Z., Licheng J., Hanqiang L., Xinbo G., Maoguo G. Spectral clustering with eigenvector selection based on entropy ranking, *Neurocomputing*, 2010, Vol. 73(10–12), pp. 1704–1717 DOI: 10.1016/j.neucom.2009.12.029.
19. Feng Zhao, Licheng J., Hanqiang L., Xinbo G., Gong M. Spectral clustering with eigenvector selection based on entropy ranking, *Neurocomputing*, 2010, 73(10–12):1704–1717. DOI: 10.1016/j.neucom.2009.12.029.

Received 17.11.2020.
Accepted 15.01.2021.

УДК 519.876.2:336

МЕТОД СПЕКТРАЛЬНОЇ КЛАСТЕРИЗАЦІЇ ПЛАТЕЖІВ І ПОСТАВКИ СИРОВИНИ ДЛЯ ПЛАНУВАННЯ АУДИТУ ВІДПОВІДНОСТІ

Нескородєва Т. В. – канд. техн. наук, доцент, Донецький національний університет імені Василя Стуса, Вінниця, зав. кафедри комп'ютерних наук та інформаційних технологій, Вінниця, Україна.

Федоров Є. Є. – д-р техн. наук, доцент, професор кафедри робототехніки та спеціалізованих комп'ютерних систем, Черкаський державний технологічний університет, Черкаси, Україна.

АНОТАЦІЯ

Актуальність. В даний час аналітичні процедури, які використовуються в ході аудиторської перевірки, базуються на методах інтелектуального аналізу даних. В роботі вирішується завдання підвищення результативності та ефективності аналітичних процедур аудиту шляхом кластеризації на основі спектрального розкладання. Об'єктом дослідження є процес аудиту відповідності послідовностей оплати і поставок сировини.

Мета. Метою роботи є підвищення результативності та ефективності аудиту за рахунок методу спектральної кластеризації послідовностей оплати і поставок сировини при автоматизації процедур перевірки їх відповідності.

Методи. Сформовано вектори ознак для об'єктів послідовностей оплати і поставок сировини, які потім використовуються в запропонованому методі. Створений метод вдосконалює традиційний метод спектральної кластеризації за рахунок автоматичного визначення кількості кластерів на основі правила поясненої і вибіркової дисперсії; автоматичного визначення параметра масштабу на основі локального масштабу (використовується правило K-найближчих сусідів); стійкості до шуму і випадковим викидів за рахунок заміни методу *k*-середніх модифікованим методом PAM, тобто заміни центроїдної кластеризації медоідною кластерізацією. Як і в традиційному підході дані можуть бути розріджені, а кластера можуть мати різну форму і розмір. Обрані характеристики оцінювання якості спектральної кластеризації.

Результати. Запропонований метод спектральної кластеризації був програмно реалізований в пакеті MATLAB. Отримані результати дозволили досліджувати залежність значень параметрів на якість кластеризації.

Висновки. Проведені експерименти підтвердили працездатність запропонованого методу і дозволяють рекомендувати його для використання на практиці при вирішенні завдань аудиту. Перспективи подальших досліджень можуть полягати в створенні інтелектуальних паралельних і розподілених комп'ютерних систем загального і спеціального призначення, які використовують запропонований метод для задач сегментації, машинного навчання та розпізнавання образів.

КЛЮЧОВІ СЛОВА: планування аудиту, кластеризація, спектральне розкладання, медоіди, послідовності оплати і поставок сировини.

УДК 519.876.2:336

МЕТОД СПЕКТРАЛЬНОЙ КЛАСТЕРИЗАЦИИ ПЛАТЕЖЕЙ И ПОСТАВКИ СЫРЬЯ ДЛЯ ПЛАНИРОВАНИЯ АУДИТА СООТВЕТСТВИЯ

Нескородева Т. В. – канд. техн. наук, доцент, Донецкий национальный университет имени Васыля Стуса, Винница, зав. кафедры компьютерных наук и информационных технологий, Винница, Украина.

Федоров Е. Е. – д-р техн. наук, доцент, профессор кафедры робототехники и специализированных компьютерных систем, Черкасский государственный технологический университет, Черкассы, Украина.

АННОТАЦИЯ

Актуальность. В настоящее время аналитические процедуры, используемые в ходе аудиторской проверки, базируются на методах интеллектуального анализа данных. В работе решается задача повышения результативности и эффективности аналитических процедур аудита путем кластеризации на основе спектрального разложения. Объектом исследования является процесс аудита соответствия последовательностей оплаты и поставок сырья.

Цель. Целью работы является повышение результативности и эффективности аудита за счет метода спектральной кластеризации последовательностей оплаты и поставок сырья при автоматизации процедур проверки их соответствия.

Методы. Сформированы вектора признаков для объектов последовательностей оплаты и поставок сырья, которые затем используются в предложенном методе. Созданный метод усовершенствует традиционный метод спектральной кластеризации за счет автоматического определения количества кластеров на основе правила объясненной и выборочной дисперсии; автоматического определения параметра масштаба на основе локального масштабирования (используется правило K-ближайших соседей); устойчивости к шуму и случайным выбросам за счет замены метода k-средних модифицированным методом PAM, т.е. замены центроидной кластеризации медоидной кластеризацией. Как и в традиционном подходе данные могут быть разрежены, а кластера могут иметь разные форму и размер. Выбраны характеристики оценивания качества спектральной кластеризации.

DOI 10.15588/1607-3274-2021-1-13

Результаты. Предложенный метод спектральной кластеризации был программно реализован в пакете MATLAB. Полученные результаты позволили исследовать зависимость значений параметров на качество кластеризации.

Выводы. Проведенные эксперименты подтвердили работоспособность предложенного метода и позволяют рекомендовать его для использования на практике при решении задач аудита. Перспективы дальнейших исследований могут заключаться в создании интеллектуальных параллельных и распределенных компьютерных систем общего и специального назначения, которые используют предложенный метод для задач сегментации, машинного обучения и распознавания образов.

КЛЮЧЕВЫЕ СЛОВА: планирование аудита, кластеризация, спектральное разложение, медоиды, последовательности оплаты и поставок сырья.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Neskorodieva T. Forecast Method for Audit Data Analysis by Modified Liquid State Machine / T. Neskorodieva, E. Fedorov, I. Izonin // The 1st International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2020), Khmelnytskyi, Ukraine, 10–12 June, 2020: proceedings. – CEUR-WS. – 2020 – Vol. 2623. – P. 25–35.
2. Neskorodieva T. Method for Automatic Analysis of Compliance of Expenses Data and the Enterprise Income by Neural Network Model of Forecast. / T. Neskorodieva, E. Fedorov // The 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT&DS-2020), Lviv-Shatsk, Ukraine, 2–3 June, 2020: proceedings. – CEUR-WS, Volume I: Main Conference. – 2020. – Vol. 2631. – P. 145–158.
3. Brusco M. J. A Comparison of Latent Class, K-means, and K-median Methods for Clustering Dichotomous Data. / M. J. Brusco, E Shireman, D. Steinley // Psychological Methods. – 2017. – Vol. 22 (3). – P. 563–580. DOI: 10.1037/met0000095.
4. Bezdek J. C. Pattern Recognition with Fuzzy Objective Function Algorithms / J. C. Bezdek. – New York : Plenum Press, 1981. – 256 p. DOI: 10.1007/978-1-4757-0450-1.
5. Fu Z. Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm. / Z. Fu, L. Wang // Multimedia and Signal Processing, 2012. – P. 61–66. DOI: 10.1007/978-3-642-35286-7_9.
6. Ester M. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise / M. Ester, H.-P. Kriegel, J. Sander, X. Xu // Second International Conference on Knowledge Discovery and Data Mining (KDD), Portland, Oregon, August 2–4, 1996: proceedings. – AAAI Press. – P. 226–231.
7. OPTICS: Ordering Points to Identify the Clustering Structure / [M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander] // International Conference on Management of Data and Symposium on Principles of Database Systems. Philadelphia, Pennsylvania, USA, May, 1999: proceedings. – Association for Computing Machinery : New York, NY, United States, 1999. – P. 49–60.
8. Mirkin B. G. Clustering for Data Mining: A Data Recovery Approach / B. G. Mirkin // Boca Raton, FL: CRC Press, 2005. – 277 p. DOI: 10.1201/9781420034912.
9. Aggarwal C. C. Data Clustering: / C. C.Aggarwal, C. K. Reddy // Boca Raton, FL: CRC Press , 2014. – 620 p. DOI: 10.1201/9781315373515.
10. Diagnostic Rule Mining Based on Artificial Immune System for a Case of Uneven Distribution of Classes in Sample / [S. Subbotin, A. Oliinyk, V. Levashenko, E. Zaitseva] // Communications. – 2016. – Vol. 3. – P. 3–11.
11. Fedorov E. Development of technique for face detection in image based on binarization, scaling and segmentation methods. /[E. Fedorov, T. Utkina, O. Nechyporenko, Y. Korpan] // Eastern-European Journal of Enterprise Technologies, Vol. 1/9, 2020. – P. 23–31. DOI: 10.15587/1729-4061.2020.195369.
12. He J. Modified ART 2A Growing Network Capable of Generating a Fixed Number of Nodes / J. He, A.-H. Tan, Ch.-L. Tan // IEEE Transactions on Neural Networks. 2004. Vol. 15(3). – P. 728–37. DOI: 10.1109/TNN.2004.826220.
13. Andrew Y. Ng. On spectral clustering: Analysis and an algorithm / Y Ng. Andrew, Jordan I. M., Y. Weiss // In Advances in neural information processing systems. – 2002. – P. 849–856.
14. Ulrike V. L. A tutorial on spectral clustering. / V. L. Ulrike// Statistics and computing. – 2007. – Vol. 17(4). – P. 395–416. DOI: 10.1007/s11222-007-9033-z.
15. Fabien L. Spectral clustering of linear subspaces for motion segmentation / L. Fabien, C Schnörr // 12th International Conference on Computer Vision (ICCV'09), Sep 2009, Kyoto, Japan: proceedings. – IEEE, pages to-appear, 2009. DOI: 10.1109/iccv.2009.5459173.
16. Tao Xiang. Spectral clustering with eigenvector selection. / X. Tao, G. Shaogang // Pattern Recognition. – 2008. – Vol. 41(3). – P. 1012–1029. DOI: 10.1016/j.patcog.2007.07.023.
17. Tao Xiang. Spectral clustering with eigenvector selection / X. Tao, G. Shaogang // Pattern Recognition. – 2008. – Vol. 41(3). – P. 1012–1029. DOI: 10.1016/j.patcog.2007.07.023.
18. Feng Z. Spectral clustering with eigenvector selection based on entropy ranking / [Z. Feng, J. Licheng, L. Hanqiang et all] // Neurocomputing. – 2010. – Vol. 73(10–12). – P. 1704–1717 DOI: 10.1016/j.neucom.2009.12.029.
19. Feng Zhao. Spectral clustering with eigenvector selection based on entropy ranking. / [Feng Zhao, J. Licheng, L. Hanqiang et al.] // Neurocomputing. – 2010. – 73(10–12). – P. 1704–1717. DOI: 10.1016/j.neucom.2009.12.029.

DOI 10.15588/1607-3274-2021-1-13

e-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2021. № 1
p-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2021. № 1

UDC 004.942

DEEP REINFORCEMENT LEARNING WITH SPARSE DISTRIBUTED MEMORY FOR "WATER WORLD" PROBLEM SOLVING

Novotarskyi M. A. – Dr.Sc, Professor of Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Stirenko S. G. – Dr.Sc, Professor, Head of Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Gordienko Y. G. – Dr.Sc, Professor of Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Kuzmych V. A. – Post-graduate student of the Department of Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

ABSTRACT

Context. Machine learning is one of the actively developing areas of data processing. Reinforcement learning is a class of machine learning methods where the problem involves mapping the sequence of environmental states to agent's actions. Significant progress in this area has been achieved using DQN-algorithms, which became one of the first classes of stable algorithms for learning using deep neural networks. The main disadvantage of this approach is the rapid growth of RAM in real-world tasks. The approach proposed in this paper can partially solve this problem.

Objective. The aim is to develop a method of forming the structure and nature of access to the sparse distributed memory with increased information content to improve reinforcement learning without additional memory.

Method. A method of forming the structure and modification of sparse distributed memory for storing previous transitions of the actor in the form of prototypes is proposed. The method allows increasing the informativeness of the stored data and, as a result, to improve the process of creating a model of the studied process by intensifying the learning of the deep neural network. Increasing the informativeness of the stored data is the result of this sequence of actions. First, we compare the new transition and the last saved transition. To perform this comparison, this method introduces a rate estimate for the distance between transitions. If the distance between the new transition and the last saved transition is smaller than the specified threshold, the new transition is written in place of the previous one without increasing the amount of memory. Otherwise, we create a new prototype in memory while deleting the prototype that has been stored in memory the longest.

Results. The work of the proposed method was studied during the solution of the popular "Water World" test problem. The results showed a 1.5-times increase in the actor's survival time in a hostile environment. This result was achieved by increasing the informativeness of the stored data without increasing the amount of RAM.

Conclusions. The proposed method of forming and modifying the structure of sparse distributed memory allowed to increase the informativeness of the stored data. As a result of this approach, improved reinforcement learning parameters on the example of the "Water World" problem by increasing the accuracy of the model of the physical process represented by a deep neural network.

KEYWORDS: Deep Reinforcement Learning, DQN-algorithm, Sparse Distributed Memory, "Water World" problem.

ABBREVIATIONS

DQN is a Deep Q Network;
RAM is a random access memory;
SDM is a Sparse Distributed Memory.

NOMENCLATURE

Φ is a reinforcement learning method;
S is a state space;
A is a set of permissible actions;
R is a reward function;
Q is a state-action function;
γ is a discount rate;
B is a logical similarity threshold of prototypes;
D is a minibach for deep neural network training;
t is a step of the algorithm;
s_t is a state of the environment at the arbitrary t;
a_t is an action that the agent implements at t;
r_t is a reward that the agent receives at t;
s_{t+1} is a next state of the environment in accordance with t;
f_t is a logical variable that determines whether the t iteration is the final iteration in the current episode;
π is a policy of the deep reinforcement learning;
K is a maximum number of transitions in the SDM;
k is an index of the transition in the SDM;
d_k is a transition tuple that is a prototype in the SDM;
M is a set of minibach indices;
m is an index of the minibach element;
norm is a parameter that determines the norm of similarity of the previous and next states of the environment;
b is a threshold of similarity of the previous and next states of the environment;
N is a coefficient of similarity of prototypes;
n is a dimension of the state vector;
δ is a threshold of the vector element similarities for the previous and next of the environment states;
i is an index of the vector elements of the environment state;
μ_t^i is a similarity of the i-th elements of the current and previous vectors of the environment states;

DOI 10.15588/1607-3274-2021-1-14

P is a SDM queue;
η is a training step size;
J is number of training sessions;
j is a current training session index;
E is a maximum number of episodes;
e is a current episode index.

INTRODUCTION

Reinforcement learning today is a broad class of methods that includes learning how to map the sequence of environmental states to agent's actions. The purpose of the actions of an agent operating in this environment is to maximize the reward it can receive. In turn, the actions of the agent can affect the environment. Therefore, the reinforcement learning methods implement a closed cycle, which significantly distinguishes them from supervised methods of machine learning. Other important features of reinforcement learning are the inability to predict the consequences of the agent's actions accurately and the need to take into account previous actions to increase future rewards. Based on these approaches, a large set of methods has been formed that use linear functions to map the space of the states of the environment to the space of actions of the agent. Linear methods have certain shortcomings, which reduce their efficiency in real-world tasks. One of the most prominent shortcomings is a significant increase in the number of elements of the state space in such tasks. As a result, in most cases, the agent is in a situation for which it has no experience of correct behavior. The only way to solve this problem is to generalize the prior experience gained by the agent and extrapolate it to future states. Algorithms based on deep learning are one of the modern approaches to generalization and have opened a new stage in development of reinforcement learning algorithms. For a long time, these algorithms were considered unstable. The DQN-algorithm has become an important step forward. It was created and tested on the "Arcade" game platform, where DQN showed high training stability. The basic idea behind modern DQN-algorithms is that the global reinforcement learning task is divided into a sequence of local supervised learning tasks. It is implemented by combining the concept of the target network with the experience replay. The data is stored in memory, which is represented by a FIFO-structure of fixed length. The deep learning network uses a training set randomly selected from memory. Ways to improve these methods are mainly based on modifications of data structures that provide the experience replay.

This paper also focuses on improving the data structures of the DQN-algorithm by applying to it the data storage principles used by SDM. This approach aims to reduce the amount of memory used and increase the stability training.

The object of study is the reinforcement learning process of a deep neural network with sparse distributed memory in solving the problem of "Water World".

The subject of the study is a method of improving the deep reinforcement learning without significantly increasing the size of RAM.

The purpose of the work is to develop a method of forming the structure and nature of access to the sparse distributed memory with increased information content to improve reinforcement learning without additional memory.

1 PROBLEM STATEMENT

Reinforcement learning method in general can be represented by a tuple,

$$\Phi = (S, A, Q, R, \gamma),$$

where $Q: S \times A \to A$, $R: S \times A \times S \to \mathbb{R}$, $\gamma \in (0,1)$.

Fig. 1 shows a generalized diagram with the main connections for interaction between the elements of the Φ tuple.

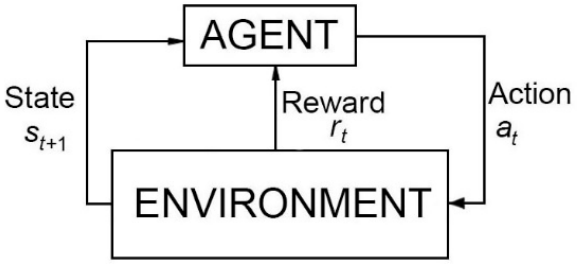

Figure 1 – The generalized scheme of interaction in reinforcement learning

The agent obtains $s_t \in S$ from the environment at an arbitrary t. As a result, the agent implements the $a_t \in A$ and receives a $r_t \in R$ after performing the a_t. The agent also perceives $s_{t+1} \in S$. In this state the environment will be at a $t+1$ time. The purpose of the training system for strengthening is to maximize the total reward, which can be represented as a mathematical expectation of the sum of all rewards with the appropriate discount rates:

$$\mathbf{R} = \mathrm{E}\left[\sum_{t=1}^{T} \gamma^t r_t\right].$$

To do this, in the case of using a DQN-agent, it is necessary to minimize the loss function, the expression of which depends on the chosen method of training.

2 REVIEW OF THE LITERATURE

Unlike other areas of machine learning, in the field of reinforcement learning there is a textbook [1], which most experts consider a basic textbook. This is very important for the unification of terminology and systematization of approaches to the development of new methods. The authors regularly update the content of this textbook. Because of this, it remains relevant for many years. Well-known machine learning algorithms use such mathematical approaches as Finite Markov Decision processes [2],

DOI 10.15588/1607-3274-2021-1-14

Multi-arm bandits [3], gradient descent methods [4], Monte Carlo methods [5], Temporal-Difference Learning [6, 7] and others. When implementing these methods, the exploration-exploitation dilemma always remains relevant [8, 9]. The most commonly used research strategies are: greedy approach, ε -greedy approach, softmax approach and Bayesian Approach [10]. All of these methods have one thing in common, as they require a large amount of memory to store a tabular representation of a value function. Approximate value functions were originally used [11] to overcome this deficiency. Modern approaches to memory reduction involve using SDM [12, 13].

One of the first stable algorithms that uses deep learning for nonlinear approximation of a value function is presented in [14]. This approach is designed to further enhance the capabilities of reinforcement learning [15]. The main advantage of this approach is that it avoided the explicit mapping of the state space to the action space. Combining the benefits of deep learning with the already known machine learning methods, DeepMind has made significant progress in training agent for the game AlphaGo [16]. The created technologies are widely used now for reinforcement learning. This paper uses a modern approach called DQN [17]. Experience replay and target network were used to increase the stability and productivity of learning on small data sets [18]. Features of the application of these technologies for the "Water World" task will be discussed below.

3 MATERIALS AND METHODS

Fig. 2 shows a framework of the deep reinforcement learning system that uses SDM.

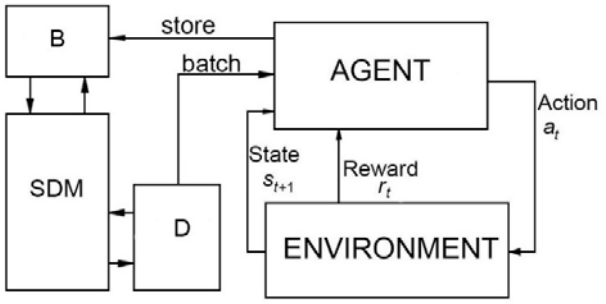

Figure 2 – Deep reinforcement learning system with SDM

As can be seen from Fig. 2, the agent affects the environment by performing the next action. The environment returns a reward for the action performed and notifies its next state, which helps the agent select the next action. This classic scheme of agent-environment interaction is complemented by the fact that each transition is stored by the agent in external memory to implement Experience Replay technology. Information about the transition is represented as the $d_k = (s_t, s_{t+1}, a_t, r_t, f_t)$ tuple, where $0 \le k \le K-1$.

A feature of the new approach we are proposing is the method of storing and reading transitions from SDM. The learning parameters are improved due to the selective storage of only "important" transitions.

Experience Replay technology often uses queued memory. This means that each new transition is always saved at the end of the queue. As soon as the specified amount of memory is full, each addition of a new transition displaces the oldest transition from SDM.

We offer a modified algorithm for storing transitions in SDM. The first step of the algorithm is to compare the new transition with the last transition in the queue. This comparison is as follows:

$$B = norm \;\&\; [a_t = a_{t-1}] \;\&\; [r_t = r_{t-1}] \;\&\; [f_t = f_{t-1}], \quad (1)$$

It should be noted that in formula (1) the expression [.] always returns a certain logical value. The *norm* parameter in expression (1) is determined from formula (2)

$$norm = \left[\min\left(\|s_t - s_{t-1}\|, \|s_{t+1} - s_t\| \right) \ge b \right]. \quad (2)$$

The b value is determined from the expression

$$b = \begin{cases} 1 - \dfrac{1}{N-1}, & N \ge 3, \\ 0.5, & N < 3. \end{cases} \quad (3)$$

The smaller the value of N is, the more states are skipped from storing to SDM due to their similarity to the previously stored states. The use of expression (3) aims to simplify the tuning process.

In the next step of the algorithm, we use the value of B variable to select the method of data storage in the SDM. If the B variable is true, then new transition is written to the place of the last transition in the queue, which corresponds to the time reference $(t-1)$. However, if the B variable is false, then a new transition is added to the end of the queue, and all other transitions are moved in the queue to one position.

Below we define the rule for determining the difference norm for the vectors of states $\|s_t - s_{t-1}\|$ and $\|s_{t+1} - s_t\|$. Let the state vector at time t be represented by elements $s_t = \left(s_t^0, s_t^1, ..., s_t^i, ..., s_t^n\right)$. To calculate the difference norm, first determine the element-by-element similarity of states using the expression:

$$\mu_t^i = \begin{cases} 1 - \dfrac{\left|s_t^i - s_{t-1}^i\right|}{\delta}, & \left|s_t^i - s_{t-1}^i\right| \le \delta, \\ 0, & \left|s_t^i - s_{t-1}^i\right| > \delta, \end{cases} \quad (4)$$

where δ is in the $[0,1]$ range.

Then the difference norm of states s_t and s_{t-1} is $\|s_t - s_{t-1}\| = \min\left(\mu_t^0, \mu_t^1, ..., \mu_t^i, ..., \mu_t^n\right)$. Similarly, using ex-

DOI 10.15588/1607-3274-2021-1-14

pression (4), define the difference norm of s_{t+1} and s_t states as $\|s_{t+1} - s_t\| = \min\left(\mu_{t+1}^0, \mu_{t+1}^1, ..., \mu_{t+1}^i, ..., \mu_{t+1}^n\right)$.

Therefore, a new transition is saved as a new SDM prototype only if the distance between the respective states exceeds the b value or there is a difference between rewards or actions at times t and $(t-1)$. The new prototype is also saved if the new state is the last state in the episode, which is determined by the f_t parameter.

Note, that the optimal definition of difference norm of states is, generally, problem-specific. We show in Section 4 that the proposed difference norm works well for the "Water World" problem.

The memory structure formed by the described algorithm evolves from episode to episode and is used by the learning agent. The agent is an off-policy agent and uses SDM to create a D training set. In the t training step the agent uses a $D_t = \{d_m\}_{m \in M}$ set of transitions. It is obvious that all indices of the selected elements in the M set are in the range $[0, K-1]$.

Algorithm 1	DQN with SDM
1	**Initialization:** P, η, K, J, γ, β, E, T, M
2	**for** $e = 1$ **to** E **do**
	environment.reset()
3	$s \leftarrow$ observe(environment)
4	**while** $t < T$ **do**
5	$a \leftarrow \pi(s)$
6	$t = t + 1$
7	$(s, r, f) \leftarrow$ environment(a)
8	$s_{next} \leftarrow$ observe(environment)
9	update_prototypes(s, s_{next}, r, a, f)
10	**end while**
11	**for** $j = 1$ **to** J **do**
12	minibatch=random.sample(prototypes,M)
13	**for** m **in** M do
14	$s^{array}.append\left(s^m\right)$
15	$s_{next}^{array}.append\left(s_{next}^m\right)$
16	$a^{array}.append\left(a^m\right)$
17	$r^{array}.append\left(r^m\right)$
18	$e^{array}.append\left(e^m\right)$
19	$\delta_j = r_j + \gamma Q\left(s_j, \arg\max_a Q(s_j, a)\right) - Q\left(s_{j-1}, a_{j-1}\right)$
20	$\theta \leftarrow \theta + w_j \delta_j \nabla_\theta Q\left(s_{j-1}, a_{j-1}\right)$ weight change
21	**end for**
22	$\theta \leftarrow \theta + \eta\Delta$ update weights
23	**end for**

Algorithm 2 for writing to SDM and creating new prototypes.

Algorithm 2	update_prototypes(prototype)
1	$(s_t, s_{t+1}, a_t, r_t, f_t)$ = prototype
2	$(s_t, s_{t-1}, a_{t-1}, r_{t-1}, f_{t-1})$ =queue[last]
3	if $N \geq 3$ then $b = 1 - \frac{1}{N-1}$ else $b = 0.5$
4	for $i = 0$ to n do
5	if $\left\|s_t^i - s_{t-1}^i\right\| \leq \delta$ then $\mu_t^i = 1 - \frac{\left\|s_t^i - s_{t-1}^i\right\|}{\delta}$
6	else $\mu_t^i = 0$
7	if $\left\|s_{t+1}^i - s_t^i\right\| \leq \delta$ then $\mu_{t+1}^i = 1 - \frac{\left\|s_{t+1}^i - s_t^i\right\|}{\delta}$
8	else $\mu_{t+1}^i = 0$
9	$\|s_t - s_{t-1}\| = \min\left(\mu_t^0, \mu_t^1, ..., \mu_t^i, ..., \mu_t^n\right)$
10	$\|s_{t+1} - s_t\| = \min\left(\mu_{t+1}^0, \mu_{t+1}^1, ..., \mu_{t+1}^i, ..., \mu_{t+1}^n\right)$
11	$norm = \left[\min\left(\|s_t - s_{t-1}\|, \|s_{t+1} - s_t\|\right) \geq b\right]$
12	if $norm \,\&\, [a_t = a_{t-1}] \,\&\, [r_t = r_{t-1}] \,\&\, [e_t = e_{t-1}]$ then
13	queue[last]=prototype
14	else
15	queue.append(prototype)

The allowable number of episodes, *E,* limits the operation of algorithm 1. In each episode, the algorithm can perform a maximum of *T* iterations if there is no terminal state, which leads to premature termination of the episode. Iteration involves the actor's interaction with the environment. The choice of the next a_t action is based on the current s_t state of the environment in accordance with the π policy. The environment returns a tuple, which contains its new s_t state, the r_t reward for the action and the e_t sign of the terminal state. This tuple, together with the previous s_{t-1} state and a_{t-1} action forms the next prototype for storage in SDM. The method of updating SDM prototypes is implemented in the *update_prototypes*() function, the pseudocode of which is shown in algorithm 2. The essence of this algorithm corresponds to the previous theoretical description. The new prototype is passed to the function as a parameter. The first stage begins with reading the last saved prototype. When fine-tuning the mode of operation of this algorithm, we provide the choice of the constant *N*, which indirectly determines the distance between two successive changes in the vectors of the states of the environment. These norms are calculated in algorithm 2 in accordance with (3) and (4). The resulting step in the analysis of the similarity of the prototypes is the calculation of the logical expression (1). If this expression returns True, then we assume that the new prototype is close to the previous saved prototype. In this case, we replace the previous prototype with a new prototype without increasing the size of the SDM. If expression (1) returns False, it indicates that the actor's action caused

DOI 10.15588/1607-3274-2021-1-14

significant changes in the environment. Therefore, such a prototype is additionally added to the SDM. If the memory is full, we delete the prototype that has been in the SDM for the longest time, as the one that has the least relevance among all saved prototypes. In each episode, we perform *J* sessions of deep neural network training by sequential modification of its θ weights. For each training session, we form a minibatch by randomly selecting prototypes from SDM with a linear probability distribution. Modification of our model allows to define TD-error and to form the graph of the loss function.

This approach has improved reinforcement learning efficiency by making the prototypes stored in SDM more informative without increasing the size of memory to store new information. The results of the study on the example of the "Water World" problem, which confirm the conclusions, are given below.

4 EXPERIMENTS

Consider the work of the described approach on the example of the problem "Water World", which was first proposed in [20, 21] and is considered a popular problem that allows you to explore the reinforcement learning algorithms. The essence of this problem is that round objects float in a square two-dimensional space. The actor is also among those objects that are hostile to him. The purpose of training an actor is to ensure the maximum possible time of the actor's existence, which can be done by avoiding collisions with enemy objects. In order to avoid such collisions, the actor has sensors that are evenly spaced in a circle. The data set created by the set of sensors forms a vector of states of the environment. The actions of the agent in this environment are reduced to the choice of direction and speed modulus in order to avoid collisions. It is important that the actor not only react to the current situation, but also gain experience that would allow him to accept a short-term deterioration of the current situation in order to avoid a catastrophe in the future. An example is a situation where an actor is surrounded by enemy objects and there is a narrow way out of that environment. In this case, the actor must temporarily move closer to enemy objects in order to escape from the environment. The actions of the actor in a similar situation are shown in Fig. 3.

The study was carried out with variation of the parameters of this task to identify their impact on the learning speed of the actor in the proposed method of increasing the informativeness of SDM. Table 1 summarizes the values or ranges of changes in the parameters of the problem.

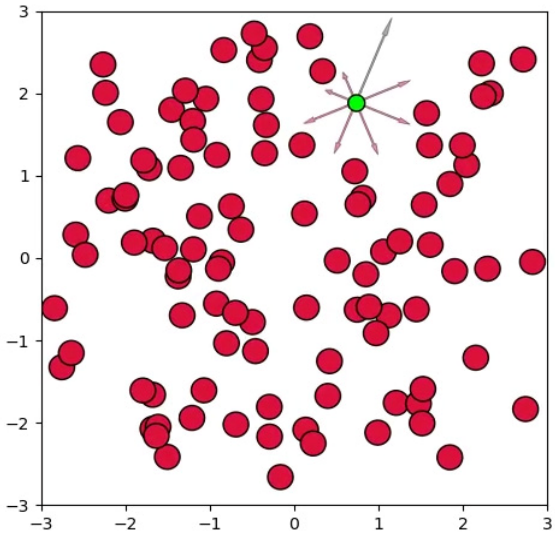

Figure 3 – The actor in a situation of leaving the environment with hostile objects

These studies have confirmed in practice the effectiveness of the proposed approach to increase the informativeness of SDM, as it allowed obtaining improved reinforcement learning when using the same amount of memory. At the same time, when it is necessary to increase the accuracy of learning, the corresponding increase in additional memory is much less than when using the FIFO-structure for memory. The specific values of this reduction require further research.

Fig. 4 shows the number of successful iterations that determine the growth of the "life expectancy" of the actor with the increase in the number of episodes.

On average, with the use of informative SDM, life expectancy increased 1.5 times compared to "Water World", which used a simple queue to store transitions.

Good results can be observed with improved learning parameters due to reduced reinforcement learning error from episode to episode. This fact confirms the graphs of the loss function value decrease (Fig. 5) in comparison with the basic solution of the problem.

Table 1 – Parameters of experiments

№	Parameter name	Range	Unit of measurement
1	Number of sensors	4–64	pcs.
2	Number of enemy objects	20–200	pcs.
3	Number of episodes	200–1000	qty.
4	Number of iterations	2000–4000	qty.
5	SDM size	10 000–100 000	number of prototypes
6	Coefficient of similarity of prototypes, *N*	1–10	qty.
7	Minibatch size	200–400	number of prototypes
8	Survival reward	+1	scores
9	Penalty for collision	–10	scores
10	Discount parameter, γ	0.99	qty.

DOI 10.15588/1607-3274-2021-1-14

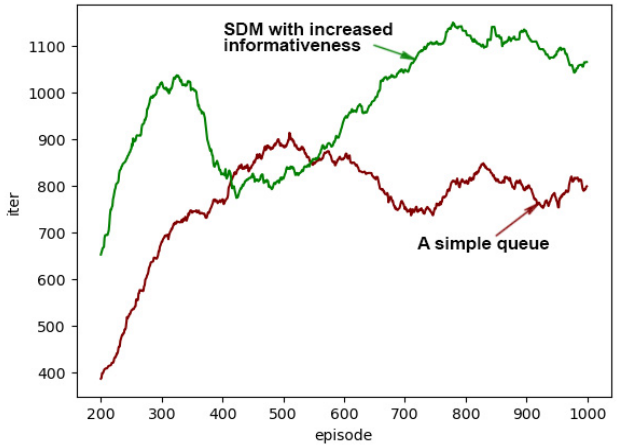

Figure 4 – Dependence of the number of iterations on the number of episodes

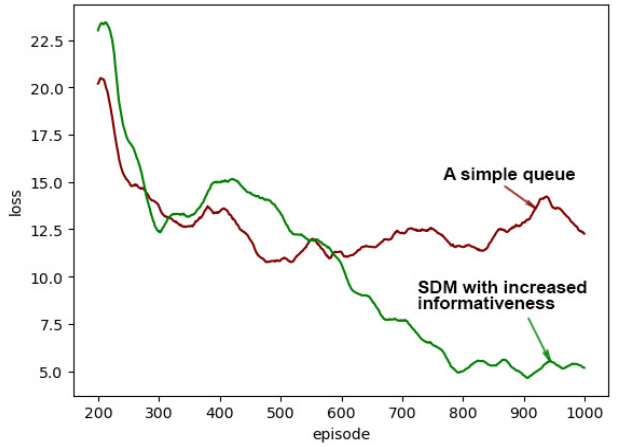

Figure 5 – Comparison of loss functions for the basic approach and method using SDM

In Fig. 6 we see the growth rate of the number of the most successful iterations, that is such iterations in which the actor received a reward of more than 3000 points, provided that the maximum reward for the iteration reaches 4000 points.

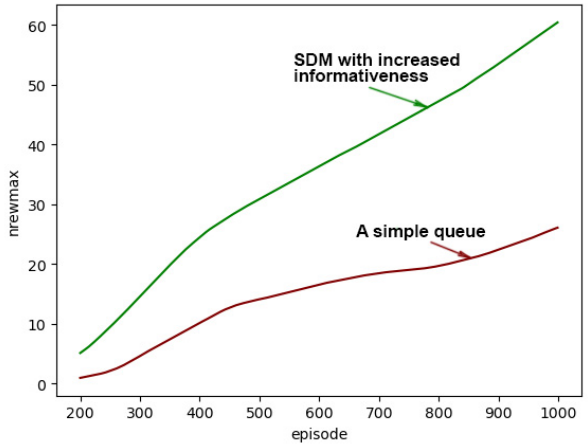

Figure 6 – Comparison of the number of successful iterations for the basic approach and method using SDM

6 DISCUSSION

The paper is devoted to one of the current approaches to reinforcement learning, which uses the deep neural network learning to build a model of the studied process. We used a DQN-algorithm based on two technologies, namely Experience replay and Target network, which allow us to create a sustainable learning process. The peculiarity of this approach is to use a certain buffer with data that reflects the previous experience of the actor, and the use of random samples from this buffer for the deep neural network. The disadvantage of this approach is the need to use large amounts of memory to ensure a successful reinforcement learning process. The main idea of this work is to use SDM to increase the informativeness of the data stored in the buffer. A method of modification and formation of SDM prototypes has also been developed, which provides increased informativeness without increasing its size. Further research, in our opinion, should be aimed at studying the hierarchical structure of SDM, provided that such a hierarchy should be formed on the basis of the hierarchy of features that are elements of the prototypes.

CONCLUSIONS

The article considers a new approach to the creation of DQN – reinforcement learning algorithms. The relevance of these studies is due to the fact that in recent years, algorithms of this type have provided tectonic shifts in the field of learning with reinforcement by allowing the use of neural network neural learning to significantly reduce the amount of RAM used to store previous experience compared to tabular methods reinforced training. Nevertheless, the need for RAM is still significant for real-world problems.

The article proposes an approach that improves the deep reinforcement of learning without a significant increase in memory, which is an urgent problem.

The scientific novelty of the obtained results is that a new method of forming the structure of memory for deep learning is proposed. This method uses a sparse distributed memory structure to store prototypes, each of which is one of the past states of the environment. The main difference of this method from existing analogues is the original principle of adding new prototypes, which allows to increase the informativeness of the stored data. Using this principle, we were able to increase the speed of deep learning by 1.5 times without increasing the size of memory.

The practical significance of the obtained results is that on the basis of this method the software system of training of deep reinforcement is developed. This software system is used to solve an important navigation problem, which is presented in the form of a well-known test task “Water World”. The task is to teach the actor to survive in a dynamically changing environment among hostile objects.

Prospects for further research are to study the hierarchical structure of sparse distributed memory, provided that such a hierarchy should be formed on the basis of a hierarchy of features as elements of prototypes.

DOI 10.15588/1607-3274-2021-1-14

ACKNOWLEDGEMENTS

The work is supported by the state budget scientific research project of National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". The title of the project is "Artificial Intelligence Platform for Distant Computer-Aided Detection (CADe) and Computer-Aided Diagnosis (CADx) of Human Diseases" (state registration number 0120U105463).

REFERENCES

1. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. Cambridge: The MIT Press, 2018, 548 p.
2. Puterman M. L. Markov decision processes: discrete stochastic dynamic programming. New Jersey, John Wiley & Sons, 2014, 684 p.
3. Zhao Q. Multi-Armed Bandits: Theory and Applications to Online Learning in Networks. NY, Morgan & Claypool, 2019, 147 p. DOI: 10.2200/S00941ED2V01Y201907CNT022.
4. Theodoridis S. Machine learning: a Bayesian and optimization perspective. Elsevier, 2020, 1160 p. DOI: 10.1016/C2019-0-03772-7.
5. Doucent A., de Freitas N., Gordon N. Sequential Monte Carlo methods in practice. NY, Springer, 2001, 616 p.
6. Hester T. TEXPLORE: Temporal Difference Reinforcement Learning for Robots and Time-Constrained Domains, NY, Springer, 2013, 179 p.
7. Sutton R. Learning to Predict by the Method of Temporal Differences, *Machine Learning*, 1988, Vol. 3, pp. 9–44. DOI: 10.1007/BF00115009.
8. Laureiro-Martinez D., Brusoni S., Canessa N., Zollo M. Understanding the exploration-exploitation dilemma: an fMRI study of attention control and decision-making performance, *Strategic Management Journal*, 2015, Vol. 36, pp. 319–338. DOI: 10.1002/smj.2221.
9. Rejeb L., Guessoum Z., Hallah R. M. An adaptive approach for the exploration-exploitation dilemma for learning agents. Berlin, Springer, 2005, pp. 316–325.
10. Mersmann O., Bischl B., Trautmann H., Preuss M., Weihs C., Rudolph G. Exploratory landscape analysis. *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 829–836.
11. Melo F. S., Meyn S. P., Ribeiro M. I. An analysis of reinforcement learning with function approximation, *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 664–671.
12. Kanerva P. Sparse distributed memory. Cambridge:The MIT Press, 1990, 155 p.
13. Wu Ch. Novel Function Approximation Techniques forLarge-scale Reinforcement Learning. PhD dissertation, 2010, 130 p.
14. Mnih V., Kavukcuoglu K., Silver D., Graves A., Antonoglou I., Wierstra D., Riedmiller M. A. Playing Atari with Deep Reinforcement Learning, *arXiv*: 1312.5602v1 [cs. LG], 2013, 9 p.
15. Ollero J., Child C.H.T. Performance Enhancement of Deep Reinforcement Learning Networks using Feature Extraction, *Lecture Notes in Computer Science*, 2018, Vol. 10878. pp. 208–218. DOI: 10.1007/978-3-319-92537-0_25.
16. Holcomb S. D. Porter W.K., Ault Sh. V., Mao G., Wang J. Overview on DeepMind and its AlphaGo Zero AI, *Proceedings of 2018 International Conference on Big Data and Education*, 2018, pp. 67–71. DOI: 10.1145/3206157.3206174.
17. Sewak M. Deep Q Network (DQN), Double DQN, and Dueling DQN, *Deep Reinforcement Learning*. Springer, pp. 95–108. DOI: 10.1007/978-981-13-8285-7_8.
18. Gao J., Shen Y., Liu J., Ito M., Shiratori N. Adaptive traffic signal control : deep reinforcement learning algorithm with experience replay and target network, arXiv 1705.02755v1 [cs.N], 2017, 10 p.

Received 15.12.2020.
Accepted 25.01.2021.

УДК 004.942

ГЛИБОКЕ НАВЧАННЯ З ПІДКРІПЛЕННЯМ З ПАМ'ЯТТЮ З ПРОРІДЖЕНИМИ ДАНИМИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ «ВОДНИЙ СВІТ»

Новотарський М. А. – д-р техн. наук, професор кафедри обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Стіренко С. Г. – д-р техн. наук, професор, завідувач кафедри обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Гордієнко Ю. Г. – д-р техн. наук, професор кафедри обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Кузьмич В. А. – аспірант кафедри обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

АНОТАЦІЯ

Актуальність. Машинне навчання це одна з галузей обробки даних, яка активно розвивається. Значних успіхів у цій сфері вдалося досягти завдяки використанню DQN-алгоритмів, які стали одними з перших стійких алгоритмів навчання при використанні глибоких нейронних мереж. Основним недоліком такого підходу є стрімке зростання оперативної пам'яті при реалізації задач реального світу. Запропонований в роботі підхід дозволяє частково вирішити цю проблему.

Мета. Метою роботи є розробка методу формування структури та характеру доступу до розрідженої розподіленої пам'яті з підвищеною інформативністю для покращення навчання з підкріпленням без залучення додаткової пам'яті.

Метод. Запропоновано метод формування структури та модифікації пам'яті з проріженими даними для зберігання попередніх переходів актора у вигляді прототипів. Метод дозволяє підвищити інформативність збережених даних і, як результат, покращити процес створення моделі досліджуваного процесу шляхом інтенсифікації навчання глибокої нейронної мережі. Підвищення інформативності збережених даних є результатом такої послідовності дій. Спочатку виконуємо порівняння нового переходу та останнього збереженого переходу. Для виконання такого порівняння, в рамках даного методу, введено норму оцінки відстані між переходами. Якщо відстань між новим переходом та останнім збереженим переходом є меншою за заданий поріг, то новий перехід записується на місце попереднього без збільшення обсягу пам'яті. У протилежному випадку створюємо новий прототип в пам'яті з одночасним видаленням прототипу, який зберігався у пам'яті найдовше.

Результати. Роботу запропонованого методу було досліджено під час вирішення популярної тестової задачі "Водний світ". Результати показали збільшення часу виживання актора у ворожому середовищі в 1,5 рази. Такий результат був досягнутий за рахунок підвищення інформативності збережених даних без збільшення обсягу оперативної пам'яті.

Висновки. Запропонований метод формування та модифікації структури пам'яті з проріженими даними дозволив підвищити інформативність збережених даних. В результаті такого підходу було одержано покращені параметри навчання з підкріпленням на прикладі задачі «Водний світ» за рахунок підвищення точності моделі фізичного процесу, представленого глибокою нейронною мережею.

КЛЮЧОВІ СЛОВА: глибоке навчання з підкріпленням, DQN-алгоритм, розріджена розподілена пам'ять, задача «Водний світ».

DOI 10.15588/1607-3274-2021-1-14

УДК 004.942

ГЛУБОКОЕ ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ С ПАМЯТЬЮ С ПРОРЕЖЕННЫМИ ДАННЫМИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ «ВОДНЫЙ МИР»

Новотарский М. А. – д-р техн. наук, профессор кафедры вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Киев, Украина.

Стиренко С. Г. – д-р техн. наук, профессор, заведующий кафедрой вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Киев, Украина.

Гордиенко Ю. Г. – д-р техн. наук, профессор кафедры вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Киев, Украина.

Кузьмич В. А. – аспирант кафедры вычислительной техники, Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», Киев, Украина.

АННОТАЦИЯ

Актуальность. Машинное обучение это одна из активно развивающихся отраслей обработки данных. Значительных успехов в этой сфере удалось достичь благодаря использованию DQN-алгоритмов, которые стали одними из первых устойчивых алгоритмов обучения с использованием глубоких нейронных сетей. Основным недостатком такого подхода является стремительный рост оперативной памяти при реализации задач реального мира. Предложенный в работе подход позволяет частично решить эту проблему.

Цель. Целью работы является разработка метода формирования структуры и характера доступа к разреженной распределенной памяти с повышенной информативностью для улучшения обучения с подкреплением без привлечения дополнительной памяти.

Метод. Предложен метод формирования структуры и модификации памяти с прореженными данными для хранения предыдущих переходов актера в виде прототипов. Метод позволяет повысить информативность хранимых данных и, как результат, улучшить процесс создания модели изучаемого процесса путем интенсификации обучения глубокой нейронной сети. Повышение информативности хранимых данных является результатом такой последовательности действий. Сначала выполняем сравнение нового перехода и последнего сохраненного перехода. Для выполнения такого сравнения, в рамках данного метода, введена норма оценки расстояния между переходами. Если расстояние между новым переходом и последним сохраненным переходом меньше заданного порога, то новый переход записывается на место предыдущего без увеличения объема памяти. В противном случае создаем новый прототип в памяти с одновременным удалением того прототипа, который хранился в памяти дольше.

Результаты. Работа предложенного метода была исследована в ходе решения популярной тестовой задачи «Водный мир». Результаты показали увеличение времени выживания актера во враждебной среде в 1,5 раза. Такой результат был достигнут за счет повышения информативности хранимых данных без увеличения объема оперативной памяти.

Выводы. Предложенный метод формирования и модификации памяти с прореженными данными позволил повысить информативность хранимых данных. В результате такого подхода были получены улучшенные параметры обучения с подкреплением на примере задачи «Водный мир» за счет повышения точности модели физического процесса, представленного глубокой нейронной сетью.

КЛЮЧЕВЫЕ СЛОВА: глубокое обучение с подкреплением, DQN-алгоритм, разреженная распределенная память, задача «Водный мир».

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Sutton R. S. Reinforcement Learning: An Introduction / R. S. Sutton, A. G. Barto. – Cambridge : The MIT Press, 2018. – 548 p.
2. Puterman M.L. Markov decision processes: discrete stochastic dynamic programming / M. L. Puterman. – New Jersey : John Wiley & Sons, 2014. – 684 p.
3. Zhao Q. Multi-Armed Bandits: Theory and Applications to Online Learning in Networks / Q. Zhao. – NY : Morgan & Claypool, 2019. – 147 p. DOI: 10.2200/S00941ED2V01Y201907CNT022.
4. Theodoridis S. Machine learning: a Bayesian and optimization perspective / S. Theodoridis. – Elsevier, 2020. – 1160 p. DOI: 10.1016/C2019-0-03772-7.
5. Doucent A. Sequential Monte Carlo methods in practice / A. Doucent, N. de Freitas, N. Gordon. – NY : Springer, 2001. – 616 p.
6. Hester T. TEXPLORE: Temporal Difference Reinforcement Learning for Robots and Time-Constrained Domains / T. Hester. – NY : Springer, 2013. – 179 p.
7. Sutton R. Learning to Predict by the Method of Temporal Differences / R. Sutton // Machine Learning. – 1988. – Vol. 3. – P. 9–44. DOI: 10.1007/BF00115009.
8. Understanding the exploration-exploitation dilemma: an fMRI study of attention control and decision-making performance / [D. Laureiro-Martinez, S. Brusoni, N. Canessa, M. Zollo] // Strategic Management Journal. – 2015. – Vol. 36. – P. 319–338. DOI: 10.1002/smj.2221.
9. Rejeb L. An adaptive approach for the exploration-exploitation dilemma for learning agents / L. Rejeb, Z. Guessoum, R. M'Hallah. – Berlin : Springer, 2005. – P. 316–325.
10. Mersmann O. Exploratory landscape analysis / [O. Mersmann, B. Bischl, H. Trautmann et al] // Proceedings of the 13th annual conference on Genetic and evolutionary computation. – 2011. – P. 829–836.
11. Melo F. S. An analysis of reinforcement learning with function approximation / F. S. Melo, S. P. Meyn, M. I. Ribeiro // Proceedings of the 25th international conference on Machine learning. – 2008. – P. 664–671.
12. Kanerva P. Sparse distributed memory / P. Kanerva. – Cambridge : The MIT Press, 1990. – 155 p.
13. Wu Ch. Novel Function Approximation Techniques forLarge-scale Reinforcement Learning / Ch. Wu. – PhD dissertation, 2010. – 130 p.
14. Playing Atari with Deep Reinforcement Learning / [V. Mnih, K. Kavukcuoglu, D. Silver et al] // arXiv: 1312.5602v1 [cs. LG], 2013. – 9 p.
15. Ollero J. Performance Enhancement of Deep Reinforcement Learning Networks using Feature Extraction / J. Ollero, C.H.T. Child // Lecture Notes in Computer Science. – 2018. – Vol. 10878. – P. 208–218. DOI: 10.1007/978-3-319-92537-0_25
16. Overview on DeepMind and its AlphaGo Zero AI / [S. D. Holcomb, W. K. Porter, Sh. V. Ault et al] // Proceedings of 2018 International Conference on Big Data and Education. – 2018. – P. 67–71. DOI: 10.1145/3206157.3206174.
17. Sewak M. Deep Q Network (DQN), Double DQN, and Dueling DQN / M. Sewak // Deep Reinforcement Learning. – Springer – P. 95–108. DOI: 10.1007/978-981-13-8285-7_8.
18. Gao J. Adaptive traffic signal control : deep reinforcement learning algorithm with experience replay and target network / [J. Gao, Y. Shen, J. Liu et al] // arXiv 1705.02755v1 [cs.N] . – 2017. – 10 p.

DOI 10.15588/1607-3274-2021-1-14