

ІНТЕЛЕКТУАЛЬНИЙ КОМП'ЮТИНГ В ПАМ'ЯТІ

Хаханов В. І. – д-р техн. наук, професор кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Абдуллаєв В. Х. – канд. техн. наук, доцент кафедри комп'ютерної інженерії, Азербайджанський державний університет нафти та промисловості, Азербайджан.

Чумаченко С. В. – д-р техн. наук, професор, завідувач кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Литвинова Є. І. – д-р техн. наук, професор кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Хаханова І. В. – д-р техн. наук, професор кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

АНОТАЦІЯ

Актуальність. Оброблені великі дані мають соціальне значення для розвитку суспільства та промисловості. Інтелектуальна обробка великих даних є умовою створення колективного розуму соціальної групи, компанії, держави та планети в цілому. При цьому економіка великих даних (Data Economy) виходить на перше місце в оцінці механізмів обробки, оскільки дуже важливими є два параметри: швидкість обробки даних та енерговитрати. Тому механізми, орієнтовані на паралельну обробку великих даних усередині центру зберігання даних, будуть завжди затребувані на IT-ринку.

Мета. Мета дослідження – підвищення економіки великих даних (Data Economy) завдяки аналізу даних як адрес таблиці істинності для ідентифікації патернів виробничих функціональностей на основі метрики подібності-відмінності.

Метод. Пропонуються архітектури Intelligent computing для управління кіберсоціальними процесами на основі моніторингу та аналізу великих даних. Пропонується обробка великих даних, як адрес таблиці істинності, для вирішення завдань ідентифікації, кластеризації, класифікації патернів соціальних та виробничих процесів. Пропонується сімейство автоматів для аналізу великих даних, як адрес. Розглядається таблиця істинності як розумна форма явних структур даних, що мають корисну константу – стандартний порядок прямування адрес. Мета обробки великих даних – зробити їх структурованими за допомогою таблиці істинності для подальшої ідентифікації до ухвалення актуаторних рішень. Таблиця істинності розглядається як механізм паралельної структуризації та пакування великих даних у її стовпці для визначення їх подібності-відмінності та еквівалентування даних за однаковими адресами. Подання даних, як адрес, пов'язане з унітарним кодуванням патернів двійковими векторами на знайденому універсумі примітивних даних. Механізм орієнтований на безпроцесорну обробку даних на основі read-write транзакцій за технологією in-memoгу комп'ютингу з суттєвою економією часу та енергії. Метрика обробки великих даних на таблиці істинності – це паралелізм, технологічна простота та лінійна обчислювальна складність. Платою за такі переваги є експоненційні витрати пам'яті зберігання явних структурованих даних.

Результати. Запропоновано паралельні алгоритми in-memoгу комп'ютингу для економічних механізмів перетворення великих неструктурованих даних, як адрес, корисні структуровані дані. Запропоновано архітектуру in-memoгу computing із глобальним зворотним зв'язком та алгоритм матричної паралельної обробки великих даних, як адрес. Вона включає структуру матричного аналізу великих даних для визначення подібності між векторами, які надходять на входи матричного сенсора. Векторний аналіз даних перетворюється на матричний комп'ютинг для обробки великих даних. Швидкість паралельного алгоритму аналізу великих даних на матриці MDV дедуктивних векторів ставиться в лінійну залежність від числа бітів вхідних векторів або потужності універсуму примітивів. Розроблено метод ідентифікації патернів ключовими словами. Він характеризується використанням унітарно-кодованих компонент даних для синтезу таблиці істинності бізнес-процесу. Це дозволяє застосовувати read-write транзакції для паралельної обробки великих даних, як адрес.

Висновки. Наукова новизна полягає у розробці наступних інноваційних рішень: 1) запропоновано нову векторно-матричну технологію паралельної обробки великих даних, як адрес, що характеризується використанням read-write транзакцій на матричній пам'яті без використання процесорної логіки; 2) запропоновано архітектуру in-memoгу computing з глобальним зворотним зв'язком та алгоритм матричної паралельної обробки великих даних, як адрес; 3) запропоновано метод ідентифікації патернів ключовими словами, який характеризується використанням унітарно-кодованих компонентів даних для синтезу таблиці істинності бізнес-процесу, що дає можливість використовувати транзакцію read-write для паралельної обробки великих даних, як адрес. Практична значимість дослідження полягає в тому, що будь-яке завдання штучного інтелекту (подібність-відмінність, класифікація-кластеризація та розпізнавання, ідентифікація образів) можна технологічно просто та ефективно вирішувати за допомогою таблиці істинності (або її похідних) та унітарно кодованих великих даних. Перспективи дослідження пов'язані з імплементацією цієї технології моделювання цифрових пристроїв на ринку EDA.

КЛЮЧОВІ СЛОВА: Intelligent Computing, Cloud, fog, and edge computing, Big data computing, In-memoгу computing, Cyber social computing, Hadoop Map-Reduce техніка, великі дані, як адреси, таблиця істинності, логічний вектор, подібність-відмінності, еквівалентність даних, універсум примітивів, патерни як двійковий вектор.

АБРЕВІАТУРИ

EDA – Electronic Design Automation;

VVV – volume, velocity, variety;

AI – Artificial Intelligence;

NN – Nueral Network;

QC – Quantum Computing;

OCR (optical character recognition) – оптичне розпізнавання символів;

GMM – модель гаусової суміші;
LR – логістична регресія;
RFC – класифікатор довільного лісу;
TF-IDF (TF – term frequency, IDF – inverse document frequency) – статистична міра, яка використовується для оцінки важливості слова в контексті документа;
HDFS – розподілена файлова система Hadoop;
MDV – матриця дедуктивних векторів;
MIV – матриця вхідних векторів;
MOV – матриця формування стовпців;
HPC (High-performance computing) – система високопродуктивних обчислень;
ТІ – таблиця істинності.

НОМЕНКЛАТУРА

D – дедуктивна матриця;
 $D1, D2$ – відмінності щодо кожного входу секвенсора;
 $S12$ – подібність між входами;
 T – матриця вхідних даних (Data as Address),
 F – матриця функціональності (Deductive Matrix L),
 M – матриця-результат аналізу (Result Matrix);
 U – універсум примітивів;
 Uj – рядок універсуму примітивів;
 Mj – стовпчик матриці аналізу примітивів;
 F – несправність,
 T – тест;
 L – логічний вектор;
 Z – якості розпізнавання кейсів;
 Qi – інтегральна оцінка якості розпізнавання кейсів
 Y – середня оцінка якості розпізнавання кейсів;
 Li – ідеальний кейс (патерн);
 Tj – актуальний кейс (патерн);
 n – кількість входів схеми або функціональності;
 $b1, b2$ – компоненти паттерна;
 X – кількість входів двигуна;
 Xi – екран;
 Yi – кількість виходів двигуна;
 k – потужність універсуму;
 P – число патернів чи екранів;
 $Q = kP/x$ – швидкодія векторної технології обробки великих даних, як адрес;
 $N = P/x$ – апаратні витрати (кількість восьмивходових елементів) на реалізацію каскадного секвенсора.

ВСТУП

Цікаві науково-практичні результати завжди виходять на стику наук шляхом інтеграції сучасних моделей, методів, алгоритмів та технологій. Так створюються нові наукові напрями, які вирішують практичні завдання для ІТ ринку. Очікуване об'єднання трьох просторів – соціального, фізичного, інформаційного – ініціювало створення нової парадигми Intelligent Computing (Інтелектуальний комп'ютинг) [1] для вирішення завдань моніторингу

© Хаханов В. І., Абдуллаєв В. Х., Чумаченко С. В., Литвинова С. І., Хаханова І. В., 2024
DOI 10.15588/1607-3274-2024-1-15

та управління кібер-соціально-фізичними процесами та створення мозку людства.

Intelligent Computing – галузь знань, що займається теорією, практикою та економікою детермінованого та ймовірнісного (AI) комп'ютингу за метрикою час-ресурси-якість для управління процесами та явищами на основі моніторингу оцифрованих та розумнопов'язаних між собою соціального, фізичного та інформаційного простору. Intelligent Computing [1] – нова парадигма обчислювальних архітектур, яка з'єднує людину з комп'ютером, традиційні обчислення з перцептивним, когнітивним та автономним інтелектом та сприяє цифровій революції в епоху великих даних, штучного інтелекту та Інтернету речей за допомогою нових обчислювальних теорій, архітектур, методів, систем та додатків. Intelligent Computing [1] включає в тому числі: Cloud, fog, and edge computing, Big data computing, In-memory computing, Design and test computing [24], Cyber social computing, Generic intelligence, Data intelligence, Analog computing, Graph computing, Artificial neural network, Fuzzy systems, Evolutionary computation, Perceptual intelligence, Cognitive intelligence, Natural language processing, Causal inference, Autonomous intelligence, Brain-computer Interface, High-performance computing, Quantum computing, Photonic computing, Biocomputing, Biocomputing computing, Intelligent computing for society, economy and governance.

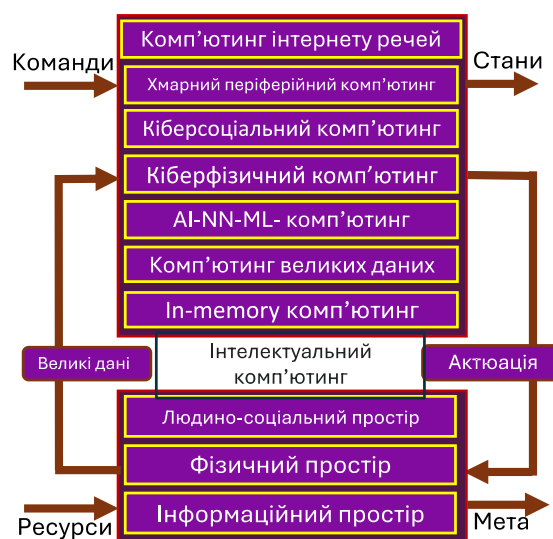


Рисунок 1 – Інтелектуальний комп'ютинг для трьох просторів

На рис. 2 позначено: Enterprise Systems – корпоративні системи; Smart IoT device – розумний пристрій IoT; Gateway – шлюз; CLOUD – ХМАРА: High computational resources – високо обчислювальні ресурси; Long-term storage – Тривале зберігання Global predictive models – глобальні прогнозні моделі; Global network management – управління глобальною мережею; FOG – ТУМАН: Distributed storage –

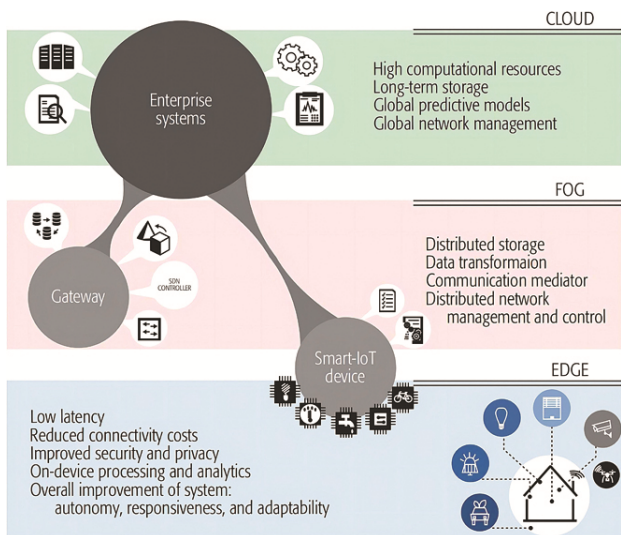


Рисунок 2 – Cloud, fog, and edge computing глобальна архітектура Intelligent Computing [1]

розподілене сховище; Data transformation – перетворення даних; Communication mediator – комунікаційний посередник; Distributed network management and control – управління та контроль розподіленої мережі; EDGE – КРАЙ: Low latency – низька затримка; Reduced connectivity costs – зменшені витрати на підключення; Improved security and privacy – покращена безпека та конфіденційність; On-device processing and analytics – обробка та аналітика на пристрої; Overall improvement of system (autonomy, responsiveness, and adaptability) – загальне покращення системи (автономність, відклик, адаптивність).

Big Data – великі дані – кількість та форма інформації, що важко сприймаються свідомістю людини. Метрика великих даних VVV: volume, velocity, variety (обсяг, швидкість генерування та різноманітність). Data Science – наука, що вивчає життєвий цикл та форми даних з метою отримання актуальної інформації для прийняття рішення. Використовуються підходи: дискретна математика та статистика, штучний інтелект та хмарні обчислення для аналізу великих обсягів даних під час вирішення завдань класифікації, регресії, кластеризації. Ключові інструменти: R, Python, Apache Hadoop, MapReduce, Apache Spark, NoSQL Databases, Cloud computing, GitHub.

Machine learning – алгоритми пошуку закономірностей у вхідних даних без програмування на основі розумних механізмів їх структуризації з метою розпізнавання патернів та прийняття рішень. «Ніхто машину не навчає». Висловлювання типу «я навчаю комп'ютер» або «ми навчили нейромережу», звучать щонайменше наївно, на думку авторитетів у цій галузі, Daniel Faggella [3]. Відношення ієрархічного порядку між механізмами AI-computing: Data Science \rightarrow Machine Learning \rightarrow (AI, NN, QC) \rightarrow Smart Data structure \rightarrow Big Data. Структурна формула дослідження поєднує такі компоненти Intelligent

Computing: Cloud-edge computing, Machine learning, Design and test computing [2], Cyber social computing, In-memory computing, Big data computing. Мета – зробити стійкими та моральними всі процеси та явища у кіберсоціальному просторі (рис. 3).

Архітектурно дослідження подано моделлю фон Неймана для хмарно-термінальної взаємодії механізмів федеративно-прискореного машинного навчання, що дозволяє керувати процесами на основі моніторингу даних, датчиків, людей та просторів. Щоб заощадити на часі і на витратах енергії, було запропоновано локально не використовувати архітектуру фон Неймана, а перейти на енергозберігаючу обробку великих даних у пам'яті, де вони зберігаються, використовуючи read-write транзакції, без потужної системи команд універсального процесора, які замінюють логічні вектори пам'яті.

Розумні (пов'язані) структури даних дозволяють без програмування вирішувати завдання структуризації корисної інформації шляхом суперпозиції таблиць та матриць, векторів. Ця ідея є основною в механізмах технічної діагностики, машинного навчання, включаючи нейронні мережі як тип розумних структур даних.

Добуток складності структур даних на обчислювальну складність алгоритму є постійною величиною (рис. 4).

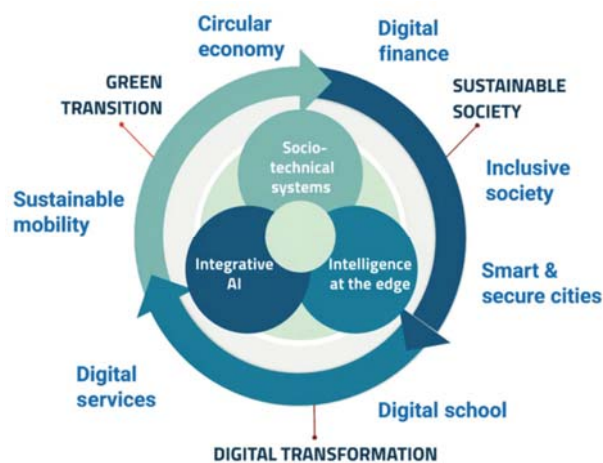


Рисунок 3 – Компоненти цифрового суспільства [1]: Socio-technical systems – соціо-технічна система; Intelligence at the edge – Інтелект на межі; Integrative AI – Інтегративний ІІІ; Green Transition – зелений перехід; Sustainable Society – стале суспільство; Digital Transformation – цифрова трансформація; Digital Finance – цифрові фінанси; Inclusive society – Інклюзивне суспільство; Smart & secure cities – Розумне та захищене суспільство; Digital school – цифрова школа; Digital services – цифрові послуги; Sustainable mobility – стійка мобільність; Circular economy – циркулярна економіка

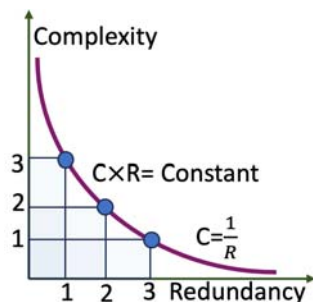


Рисунок 4 – Відношення між надмірністю структур даних та складністю алгоритму їх обробки: Complexity (C) – складність; Constant – константа; Redundancy (R) – надмірність

Тому платою за найпростіші алгоритми є експоненційні витрати пам'яті. Такі витрати будуть виправдані, якщо структура даних містить у явному вигляді рішення комбінаторних завдань. Найбільш універсальною моделлю комп'ютерного вирішення всіх комбінаторних завдань є таблиця істинності, якій понад сто років. Її явні структури даних адекватно описують такі цікаві об'єкти: логічну функціональність, графову структуру, діаграму Хасе, бінарні діаграми, двійкове дерево рішень. Таблиця істинності перетворює експоненційну складність алгоритму розв'язання комбінаторної задачі на лінійну. Крім того, таблиця істинності моделює життєвий цикл великих даних, які подорожують таблицею істинності від его (відмінності) до самості (подібності). Питання лише у тому, як підготувати великі дані їхньої обробки на таблиці істинності, як адрес.

Чому цього ніхто не робив раніше? 1. Таблиця істинності (ТІ) лякає дослідників своєю експоненційною розмірністю. 2. Мало хто звертав увагу на комбінаторику явних адрес ТІ для вирішення перебірних завдань. 3. Найголовніше – це представляти великі дані, як адреси ТІ, – простий хід у виконанні, але виявляється не дуже простим у розумінні. 4. Властивість адресації, закладена у ТІ, вважають дослідники, закриває ворота для паралельної обробки інформації, хоча це зовсім не так. 5. Важко користуватися ТІ, наприклад, на 20 змінних. 6. Мало хто асоціює логічний вектор з компактним записом ТІ, на якій вирішуються практично всі завдання синтезу та аналізу оцифрованих процесів та явищ, включаючи кіберсоціальні та кіберфізичні. 7. Важко було припустити ТІ як структуру, якою подорожують великі дані від різниці до подібності, від его до самості по Карлу Янгу і навпаки, ідентифікуючи життєвий цикл даних. 8. Незрозуміло, чому не використовувалися адреси ТІ для ідентифікації та моделювання комбінацій несправностей та помилок у кіберфізичних та кіберсоціальних системах.

Вже утворився клуб вчених та підприємців (14 осіб) у галузі ІТ-індустрії, які мають нагороди Global-IT, що є аналогом Нобелівської премії. Практично всі вони у своїй інавгураційній промові заявляли, що

майбутнє планети за комп'ютерними рішеннями в галузі моніторингу та управління суспільством. На цьому шляху людство має пройти такі кроки: 1) створення мозку людства до 2050; 2) безпілотне вирішення проблеми транспорту до 2030 року; 3) генерація запчастин людського організму до 2100; 4) вирішення проблеми харчування до 2040 року; 5) Intelligent Computing загального моніторингу та морального управління суспільством до 2080 року. Тут має бути добра воля політиків та керівників усіх країн та провідних компаній планети. Перший крок уже зроблено – IEEE-товариство легалізувало використання систем штучного інтелекту для рутинної роботи оформлення наукових ідей, досліджень та статей. Роботи ведуться зі створення стандарту ідентифікації оригінальних ідей у текстах та визначення валідності рівнів плагіаризму. Розумні контракти-програми, які пропонувалися в блокчейн комп'ютерного управління соціальними групами, будуть замінюватись розумними структурами даних без програмування, які формують ML-механізми для моніторингу та управління соціальними групами, університетами, компаніями та державними структурами.

Об'єкт дослідження – in-memory intelligente комп'ютеринг, який знижує енергетичні та годинні витрати під час обробки великих даних.

Предмет дослідження – in-memory аналіз елементів або цифрових схем будь-якої розмірності за допомогою read-write транзакцій на логічних векторах.

Мета дослідження – підвищення економіки великих даних (Data Economy) завдяки аналізу даних, як адрес таблиці істинності, для ідентифікації патернів виробничих функціональностей на основі метрики подібності-відмінності.

1 ПОСТАНОВКА ЗАДАЧІ

Розглядається задача побудови логічного вектора бізнес-функціональності для моніторингу та управління бізнес-процесом. Модель синтезу логічного вектора L подана адресним відношенням між екранами T і таблицею істинності F : $L=F(T)$. Розв'язок задачі полягає в обробці великих даних, що знаходяться на екранах, як адресів таблиці істинності, що формує відповідну кількість ключових слів, з яких складається логічний вектор або ідентифікатор групи екранів для формування шаблону або логічної функціональності. Для представлення ключових слів адресами таблиці істинності необхідно їх унітарно закодувати на множині дозволених символів, які формують універсум. Задача побудови логічного вектора на таблиці істинності має лінійну складність, завдяки експоненціальному характеру надлишковості структури даних таблиці істинності. Формальне рівняння синтезу логічного вектора функціональності бізнес-процесу подано співвідношенням наступних параметрів: $L=F(X)$, L – логічний вектор

функціональності, F – таблиця істинності, X – кінцеве число змінних-екранів, на яких задана таблиця істинності та логічний вектор.

Нехай задано потік екранів $X = \{X1, X2, \dots, Xn\}$, які мають OCR-дані, представлені двома множинами: 1) headers – заголовки додатків та активних полів екрану; 2) keydata – дані, що заносяться в активні поля екрану. Практичне завдання побудови логічної моделі бізнес-процесу полягає у формуванні з ключових даних, що належать множині $\{\text{headers}, \text{keydata}\}$, ідентифікаторів екрану. Сукупність патернів формує логічну функцію бізнес-процесу. При цьому патерн – будь-яка просторово-часова послідовність елементів, що відтворюється, об'єднана смисловою метрикою. Патерн або кейс є реалізація бізнес-функціональності, яка залишає слід у вигляді даних. Ідентифікація патерну виходить шляхом стиснення великих даних у стовпцях таблиці істинності.

Ідея дослідження проста: для перетворення експоненційного алгоритму на лінійний, потрібно використовувати експоненційні структури явних даних, ідеальною формою яких є ТІ. Для обробки великих даних, як адрес ТІ, необхідно виконати унітарне кодування даних на універсумі примітивів. Отриману матрицю двійкових векторів перетворювати на адреси у вікні спостереження, що дорівнює кількості змінних у ТІ, та запакувати на ТІ. Комбінаторні властивості структури ТІ формують по

стовпцях еквівалентні множини даних при лінійних витратах часу. На ТІ вирішуються такі завдання аналізу великих даних: 1) еквівалентування даних по стовпцям адресам ТІ; 2) визначення подібності-відмінності за даними у патернах; 3) пошук та ідентифікація патернів у великих даних; 4) пошук даних за заданими шаблонами; 5) побудова логічної моделі процесу чи явища за допомогою унітарного кодування патернів двійковими векторами на універсумі примітивів; 6) валідація кіберсоціального процесу та пошук актуаторних станів та помилок у ньому; 7) побудова тесту моделі кіберсоціального процесу.

Поетапно схему синтезу комп'ютерного можна подати такими задачами (рис. 5): 1) використання властивостей ТІ для ідентифікації екранів, еквівалентування ключових даних; 2) знаходження патернів виробничих функціональностей на основі подібності-відмінності логічних векторів, що формують фрагменти кейсу; 3) кластеризація бізнес-потоків великих даних щодо виробничих функціональностей; 4) ідентифікація виробничих функціональностей та кейсів у бізнес-потокі даних; 5) побудова автоматів пошуку подібності-відмінності даних у патернах бізнес-потоків; 6) синтез логічного вектора виробничого процесу; 7) синтез логічних векторів для моніторингу та управління виробничим процесом.



Рисунок 5 – Схема синтезу комп'ютерного бізнес-процесу

2 ОГЛЯД ЛІТЕРАТУРИ

Цикл поліпшення структур даних, алгоритмів та механізмів становить інтерес з погляду економії ресурсів та часу на обробку великих даних. Підтвердженням сказаного може бути велика кількість публікацій у IEEE Xplore за останні 10 років, присвячених темі Data Economy. Спостерігається стійкий тренд зниження карт-вузлів в архітектурі Map Reduced під час обробки великих обсягів даних [4–8]. Великі дані прагнуть використання одного data-центру з великим обсягом пам'яті для зберігання інформації. Це насамперед необхідно для зниження витрат енергії і часу обробки даних рахунок зменшення транзакцій передачі між обчислювальними центрами.

Проведені експерименти показують, що скорочення карт має велику цінність при обробці величезних

обсягів даних у дуже великих кластерах. Використовуються також цікаві доповнення до Map Reduced, такі як Наївний Байєс, J48 та Random Forest, K-Means модель гаусової суміші (GMM), логістична регресія (LR) та класифікатор випадкового лісу (RFC) для ефективного майнінгу всередині величезних обсягів даних, які також економлять енергію. Показано, що Spark-технологія у 16 разів швидше, ніж обчислення Hadoop Map-Reduce для обробки великих даних і більш ефективна з точки зору економії енергії. Розглядаються механізми [9] прискорення ітеративних алгоритмів машинного навчання, реалізовані в Hadoop Map-Reduce (Stock) та Apache Spark на основі графічного процесора. Тут все добре зі швидкодією, але погано з енерговитратами. Розглянуто три алгоритми класифікації [10]: класифікатор k -NN, класифікатор центроїдів та

найвний байєсівський класифікатор. Емпіричні результати показують, що класифікатор Centroid є найбільш точним у цьому випадку з точністю до 95% порівняно з k -NN, точність якого становить 92%, і найвним класифікатором байєсівським з точністю 91,5%. Пропонується динамічне налаштування слотів [11] у процесі обробки даних – Dynamic Hadoop Slot Allocation, що підвищує завантаження кожного обчислювального вузла. Пропонується метод [12] передачі величезної бази даних від місця зберігання до вузла обчислення та індексування для розподілу запитів, масштабованості та продуктивності в гетерогенних середовищах. Результати свідчать, що запропонована робота скорочує час обробки даних на 30%. Наведено систему [13] обробки даних для автоматичної класифікації на основі важливих медичних термінів з використанням TF-IDF та тематичного моделювання для прийняття валідних діагнозів. Розглядається використання додаткової обчислювальної потужності [14] для зниження комунікаційного навантаження у розподілених обчисленнях, що позитивно впливає на швидкодію та зниження витрат енергії. Більш конкретно, розглядається загальна структура розподілених обчислень, заснована на структурах, що часто використовуються, таких як MapReduce, де загальні обчислення розбиваються на обчислення набору функцій «Map» і «Reduce», розподілених за сукупністю обчислювальних вузлів. Пропонується стратегія віртуального перетасовування [15], що дозволяє забезпечити ефективне переміщення даних та скорочення кількості операцій введення-виводу для перетасовування MapReduce, тим самим знижуючи енергоспоживання та зберігаючи енергію. Віртуальне перетасовування реалізується за допомогою комбінації трьох методів, включаючи трирівневу таблицю сегментів, злиття, близьке до вимоги, а також динамічне та збалансоване злиття піддерев. Економія енергоспоживання програм MapReduce склала 12%. Традиційна система високопродуктивних обчислень (HPC) є прикладом системи з обмеженою пам'яттю. Пропонується нова система MapReduce (Mammoth) [16], метою якої є підвищення продуктивності за рахунок глобального управління пам'яттю, що підвищує швидкодію обробки даних на 40%. Розвиток цифрової індустріалізації [17–21], цифрової економіки великих даних на місцях, досить віддалених від центру, представляє велику телекомунікаційну проблему, яку потрібно вирішувати для того, щоб мати цифрову індустрію інваріантну геопозиції будь-якого населеного пункту. Підкреслюється важливість великих даних у соціальній економіці, розкривається значення цифрової економіки промисловості великих даних, і

навіть аналізуються реформи, викликані великими даними та заходи, створені задля їх застосування [21].

Таким чином, розподілена комп'ютерна система складається з множини взаємопов'язаних вузлів. Вони можуть бути фізичними, віртуальними машинами чи контейнерами. Коли група вузлів надає клієнту послуги та програми, якби це була одна машина, її також називають кластером. Hadoop – це платформа для виконання завдань на кластерах комп'ютерів, яка забезпечує гарну якість базового обладнання та програмного забезпечення. Hadoop3 – це проект, написаний на Java з відкритим кодом Apache Software Foundation.

MapReduce – це парадигма програмування, що забезпечує масштабування сотень або тисяч серверів у кластері Hadoop. Файлова система Google використовує методи розподілу, обробки та агрегування великого обсягу даних для пошукової системи Google. Версія MapReduce з відкритим вихідним кодом пізніше була випущена в рамках проекту Apache Hadoop з орієнтацією на функціональне програмування, яке орієнтоване на знання hardware архітектури. Фази виконання завдання MapReduce: поділ даних на кількох комп'ютерних вузлах, застосування функції карти до кожного фрагмента даних. Сортування та перемішування даних, та розподіл за редукторами. Скорочення даних із видачею результату. Чотири етапи: розподіл, зіставлення, сортування та перемішування, скорочення (split, map, sort & shuffle, reduce) (рис. 6, а). Архітектура Hadoop є пакетом файлової системи, механізму MapReduce і HDFS (розподіленої файлової системи Hadoop). Кластер Hadoop складається з одного головного та кількох підлеглих вузлів. Головний вузол включає Job Tracker, Task Tracker, NameNode і DataNode, тоді як підлеглий вузол включає DataNode і TaskTracker. Проблемне місце тут – блок сортування та перемішування даних, який у кожній компанії є секретним ключем, який формує успіх компанії та позитивну економіку обробки великих даних. MapReduce – це макро-виконання механізму глибоких нейронних мереж, де від шару до шару працюють процедури сортування та перемішування даних.

У метриці Hadoop Map-Reduce техніка та архітектура аналізу даних за допомогою ТІ представлена на рис. 6,б. Переваги запропонованого розв'язку: 1) відсутні канали зв'язку між пам'яттю та процесором між процесорами-вузлами кластера; 2) відсутній блок перемішування та переплутування інформації, який замінює розумна структура рядків-адрес таблиці істинності; 3) блок розподілу даних за картками-вузлами замінюється поділом унітарно кодованих даних за адресами, що набагато ефективніше за швидкодією та енерговитратами.

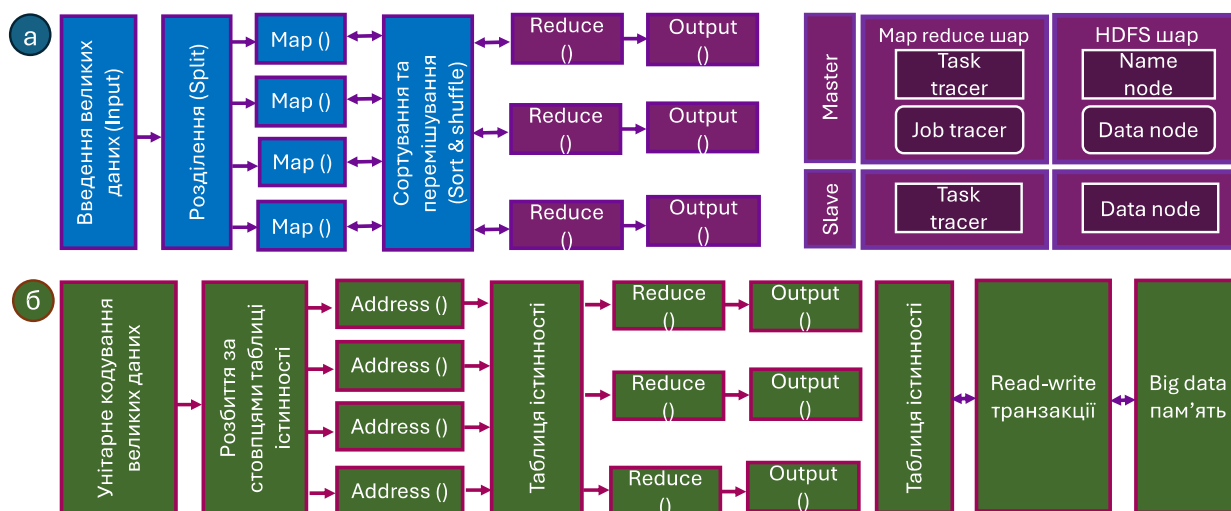


Рисунок 6 – Схема технік та архітектур для обробки великих даних Hadoop Map-Reduce

Оброблені великі дані мають соціальне значення для розвитку суспільства та промисловості. Інтелектуальна обробка великих даних є умовою створення колективного розуму соціальної групи, компанії, держави та планети в цілому. При цьому економіка великих даних (Data Economy) виходить на перше місце в оцінці механізмів обробки, оскільки дуже важливими є два параметри: швидкість обробки даних та енерговитрати. Тому механізми, орієнтовані на паралельну обробку великих даних усередині центру зберігання даних, будуть завжди потрібні на IT-ринку. Великі дані – кількість інформації, що важко сприймається свідомістю людини. Метрика великих даних VVV: Volume, Velocity, Variety (обсяг, швидкість генерування та різноманітність). 1. Об'єм означає, що Big Data є великі масиви інформації. Сьогодні в кіберпросторі знаходиться 165 зета-байт інформації, з них використовується лише 4% структурованих даних. 2. Швидкість створення даних. Один автомобіль без драйвера генерує сьогодні 4 терабайти даних за день. 3. Різноманітність – це різні джерела інформації та різноманітність форматів (аудіо, відео, фото, тексти, символи) даних. Економіка великих даних – це довести 4 корисні відсотки до 20 з метою отримання штучного розуму або мозку людства з мінімальними витратами за часом та енергією. Далі розглядаються економічні механізми перетворення великих неструктурованих даних, як адрес, в корисні структуровані дані, за рахунок розробки паралельних алгоритмів in-memory комп'ютерингу.

3 МАТЕРІАЛИ І МЕТОДИ

Розглянемо online класифікацію великих даних за метрикою подібності-відмінності патернів у бізнес-потокі екранів. Вихідні дані представлені потоком OCR-розпізнаних текстових фрагментів на сотнях зображень екранів комп'ютера. Математична модель представлена у вигляді дедуктивних векторів, які виконують роль фільтра для визначення подібності-

відмінності екранів бізнес-потокі даних. При цьому дані на екранах подані двійковими векторами унітарно кодованих примітивів універсуму, які розглядаються як адреси для вилучення інформації з дедуктивного вектора.

Паралельний аналіз двійкової вхідної інформації на матриці трьох векторів (0100 1101 0001) для отримання відмінностей по кожному входу (D_1 , D_2) і подібності між ними (S_{12}) представлений на рис. 7. Адреси формуються комбінацією сигналів з першого та другого вхідного вектора output vectors, які складають стовпець-код для зчитування відповідного стовпця (00 01 10 11) векторної матриці, що формує паралельно стовпці output vectors. Природно, що логічне множення (&) трьох векторів ($D_1 \& D_2 \& S_{12}$) = 000000000000 дає нуль вектор, та їх диз'юнкція (+) вхідних векторів: ($D_1 + D_2 + S_{12}$) = 111001111110. Векторний аналіз даних перетворюється на матричний комп'ютеринг для обробки великих даних. Тому нижня частина рис. 7, б присвячена структурі матричного аналізу великих даних для визначення подібності-відмінності між векторами, яка надходить на входи матричного секвенсора. Кількість входів такого секвенсора – число рядків у матриці D – завжди дорівнює кількості потоків даних, які необхідно обробляти паралельно. Кожен рядок матриці L своїми одиничними значеннями формує деяку функціональність (схожість, відмінність) обробки вхідних даних, як адрес. Сукупність рядків дедуктивної матриці L формує метрику властивостей, кількість яких дорівнює виходам секвенсора (матриці M). Інакше, кількість векторів матриць L і M дорівнює між собою.

Формально можна розглядати запропонований комп'ютеринг як хог-відношення трьох матриць, що дорівнює нулю, де T – вхідні дані (Data as Address), F – функціональність (Deductive Matrix L), M – результат аналізу (Result Matrix).

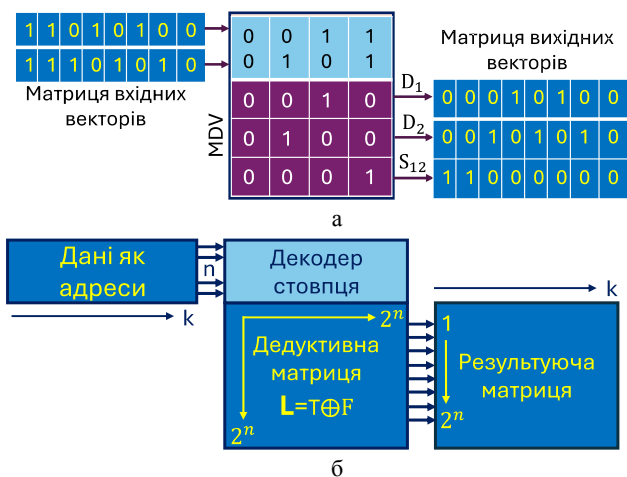


Рисунок 7 – Матричний паралельний аналіз великих даних, як адрес

Швидкодія паралельного алгоритму аналізу великих даних на матриці MDV дедуктивних векторів ставиться в лінійну залежність від числа бітів вхідних векторів або потужності універсуму примітивів. Універсум (словник) – впорядкована множина (вектор) неповторних даних на вибраній кількості екранів. Універсум знаходиться шляхом обчислення множини всіх примітивних символів (слів) на основі оператора збору оригінальних текстів всіх аналізованих екранів (рис. 8).

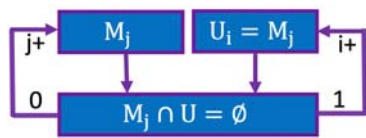


Рисунок 8 – Алгоритм отримання універсуму примітивів

У процесі створення універсуму як метрики бізнес потоку виконується кодування кожного символу десятковим номером (рис. 9). Після цього на отриманій метриці виконується унітарне кодування символів кожного екрана.

1	2	3	4	5	6	7	8	9	10	11	12
ma	he	ye	in	on	no	do	go	di	ha	qu	me
1	1	0	0	0	1	0	0	0	1	0	0
1	1	1	0	0	0	1	1	1	0	1	0

Рисунок 9 – Вектор універсуму примітивів та унітарне кодування символів екрану

Потім кожна множина символів екрану кодується двійковим вектором, де одиницею відзначається позиція, що відповідає символу, що присутня на екрані, що кодується. Отримані вхідні дані, що становлять вхідну матрицю вхідних векторів (MIV), подаються на матрицю векторів дедуктивних MDV для подальшого аналізу, де стовпці вхідної матриці розглядаються як адреси для зчитування відповідних стовпців MDV для формування стовпців MOV. Якщо необхідно проаналізувати $n=8$ екранів одночасно, тоді синтезується наступна матриця для елемента на вісім

входів і кожен дедуктивний вектор буде розмірності $2^n=256$ біт.

Архітектура векторної технології обробки великих даних має структуру (рис. 10), яка виконує функції: 1) Preprocessing – підготовка текстів до моделювання шляхом пошуку універсуму текстових примітивів та унітарного кодування екранів двійковими векторами; 2) M-Computing – memory-based computing – моделювання двійкових векторів даних на пам'яті-матриці дедуктивних векторів, що виділяють подібність-відмінність; 3) Postprocessing – виконується перетворення двійкових вихідних векторів у текстові примітиви.

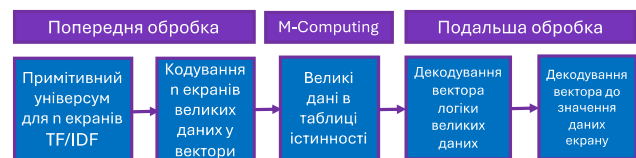


Рисунок 10 – Векторна технологія обробки великих даних

Розглянемо аналіз даних, як адрес ТІ, для ідентифікації екранів та патернів-кейсів бізнес-функціональності на основі метрики даних, впорядкованих за їх значимістю. Завдання можна зобразити схемою векторно-логічного синтезу моделі бізнес-процесу (рис. 11): 1) використання властивостей ТІ для ідентифікації екранів, еквівалентування ключових даних-слів; 2) знаходження патернів-кейсів на основі подібності логічних векторів, що формують кейси; 3) розкладання бізнес-поточку великих даних щодо суттєвих функціональностей; 4) розпізнавання функціональностей та кейсів у бізнес-поточці даних та їх ідентифікація; 5) побудова online-автомата управління функціональностями бізнес-процесів; 6) щоб відновити функціональність потрібні два компоненти: питання-headers або вхідні умови, а також відповіді-дані або стан виходів, які слід автоматично сконтактенувати: $F=TxorL$ – питання-відповідь. Наявність вихідних даних не гарантує відновлення функціональності, для цього потрібно знати ще й вхідні дані, тобто питання, які задає система для отримання, наприклад, кредиту. Аналіз питань та відповідей в автоматичному режимі дає можливість отримати функціональність бізнес-процесу, представлену header-data ідентифікаторами екранів.

Для вимірювання відношень між процесами та явищами застосовується метрика Similarity-Difference. Найбільш адекватною і простою цифровою моделлю для опису відношень Similarity-Difference є ТІ. Вона поєднує властивості Similarity-Difference у повному комбінаторному виконанні для вирішення завдань класифікації, кластеризації та ідентифікації патернів (екранів). Вирішення завдання ідентифікації екранів пов'язане з виконанням наступних процедур: 1) для потоку екранів необхідно формувати універсум примітивів із ключових слів; 2) виконується кодування потоку екранів двійковими векторами у метриці універ-

суму ключових слів; 3) потік двійкових даних у форматі-вікні з 8-екранів-змінних стискається в ТІ по стовпцях, які є упорядкованими адресами ТІ. Вона розставляє ключові слова згідно з адресами на вісімці екранів. Кожен стовпець ТІ формує еквівалентну множину ключових слів.

Якість розпізнавання кейсів Z , інтегральна оцінка якості розпізнавання кейсів Q_i ; – середня оцінка

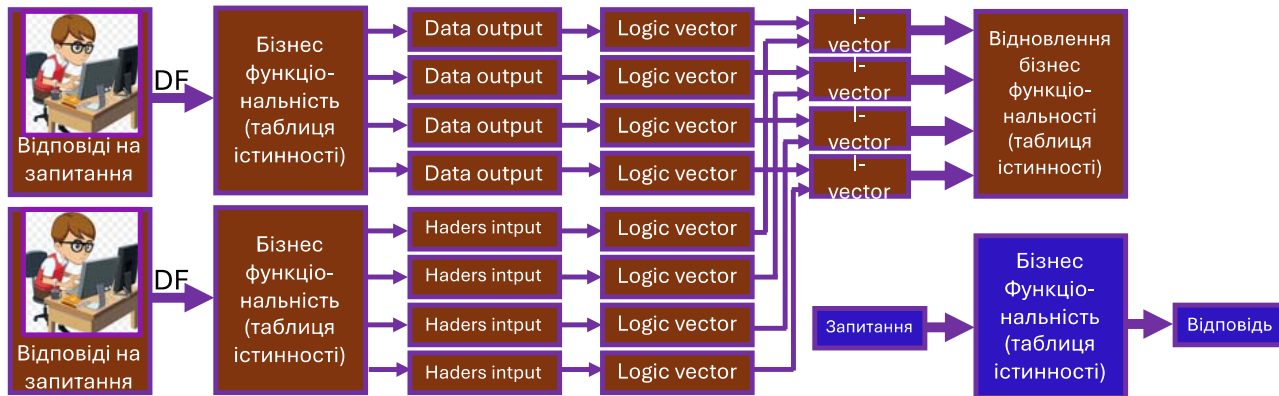


Рисунок 11 – Схема векторно-логічного синтезу моделі бізнес-процесу

4 ЕКСПЕРИМЕНТИ

Взаємодія двигуна з великими даними для отримання результату аналізу показано на рис. 12. Усі три компоненти знаходяться у пам'яті. При цьому двигун Deductive Matrix Engine (автомат з обробки великих даних із зворотними зв'язками) становить computational memory, а вхідні Data as address та вихідні Result Matrix дані – conventional memory. Змінні глобальних зворотних зв'язків враховують передісторію подібності-відмінності вже оброблених патернів визначення метричних властивостей поточних даних. На першому кроці алгоритму псевдозмінні набувають значення $L=00...00$ на метриці знайденого універсуму. Алгоритм закінчує свою роботу після того, як до двигуна будуть завантажені всі дані на шляху його руху зверху вниз. Слід зазначити, що матриця вхідних даних та результат аналізу повністю ідентичні за розмірами. В даному випадку, за розмірами збігається кількість входів X і виходів двигуна Y .

Продуктивність Q даного двигуна з обробки великих даних визначається добутком потужності універсуму та найближчого цілого, що не менше за відношення кількості патернів P до кількості активних входів X двигуна. Метрика або рівень паралелізму визначається кількістю актуальних входів X . Продуктивність обробки великих даних залежить від розмірів матриці вхідних даних, поділеної на кількість актуальних входів X Deductive Matrix Engine. Для підвищення швидкодії аналізу даних необхідно збільшувати число вхідних логічних змінних X дедуктивної матриці. У цьому слід пам'ятати, що функціональні можливості дедуктивної двійкової матриці обмежуються простором $2^n \times 2^n$, n – число вхідних змінних, у якому вирішуються будь-які комбінаторні завдання з класифікації та ідентифікації будь-яких унітарно кодованих процесів і явищ, включаючи біометричні [23, 24, 26].

© Хаханов В. І., Абдуллаєв В. Х., Чумаченко С. В., Литвинова С. І., Хаханова І. В., 2024
DOI 10.15588/1607-3274-2024-1-15

якості розпізнавання кейсів Y визначаються відповідним чином через ідеальний кейс/патерн Li ; актуальний кейс/патерн Tj ; кількість входів схеми або функціональності n та компоненти патерна $a1, a2, b1, b2$.

Потрібно будувати оцінку покриття еталонних кейсів, або оцінку максимальних покриттів еталонних кейсів відрізками знайдених кейсів.

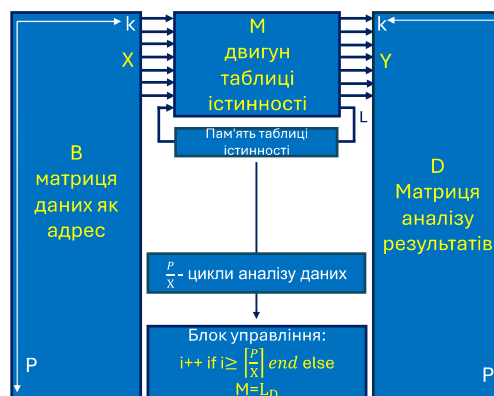


Рисунок 12 – Архітектура in-memory автомата та sliding алгоритм матричної обробки даних, де k – потужність універсуму, P – число патернів або екранів, що підлягають аналізу, $X = 7$ – кількість активних входів двигуна

Стовпці між собою створюють порядок Similarity-Difference (рис. 13), оцінки яких зменшуються від країв до центру і далі таблиці істинності; 4) тому для ідентифікації екранів необхідно завжди вибирати ключові слова, що відповідають крайнім правим та лівим стовпцям таблиці істинності, що мають максимальні значення difference та similarity. Similarity-стовпець формує клас, Difference-стовпець формує унікальний ідентифікатор екрану. Метрика TF/IDF підтверджує значущість 1-координат стовпців таблиці істинності для ідентифікації екранів за допомогою її комбінаторної структури метрики Similarity-Difference. Стовпці таблиці істинності мають однакову інтегральну оцінку TF/IDF. Це означає, що Difference-координати ($n=8$) у структурі таблиці істинності мають максимальну оцінку значимості будь-якої функції. Для верифікації логічної функціональності ці точки обов'язково мають бути покриті тестами (testbench).

U-set	a	b	c	d	k	e	y	j	z	x,2	s,3	r,4	g,5	m,6	w,7	q,1
No	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1		$\frac{14}{81}$		1		1		1		$\frac{14}{82}$		1		1		$\frac{14}{84}$
x_2			$\frac{14}{81}$	1											$\frac{14}{83}$	$\frac{14}{84}$
x_3					$\frac{14}{81}$	1							$\frac{14}{82}$	1	$\frac{14}{83}$	$\frac{14}{84}$
x_4									$\frac{14}{81}$	$\frac{14}{82}$	1	1	$\frac{14}{82}$	1	$\frac{14}{83}$	$\frac{14}{84}$
D		1	1	1					1	1			1		1	1

Рисунок 13 – Метрика TF/IDF координат таблиці істинності TF/IDF [28] = [(number of times term t appears in a document) / (total number of terms in the document)] * log10 (Total number of documents / Number of documents with term t in it)

Перевірка несправностей цих точок гарантує перевірку інших несправностей ліній цифрової структури на 90%. Ці точки вказані як критичні для дедуктивної матриці, побудованої для верифікації логічної функціональності. Критичні точки (стану) у будь-якій логічній функціональності кейсів збирають максимальну кількість ключових даних.

5 РЕЗУЛЬТАТИ

Таблиця істинності якісно та метрично визначає еквівалентну множину екранів на координатах стовпців [22, 27]. Логічний вектор визначає сигнатуру функціональності у кадрі. Щоб розрізнити еквівалентні екрани у кадрі, необхідно зробити наступне: 1) виключити екрани, що повторюються; 2) укласти у дужки еквівалентні ключові слова по стовпцях таблиці істинності; 3) similarity зменшується по стовпцях праворуч ліворуч, difference зростає по стовпцях зліва направо. Ідентифікатор екрана складається з крайніх ключових слів.

Локальні та інтегральні оцінки якості розпізнавання кейсів наведено на рис. 14. Потрібно будувати оцінку покриття еталонних кейсів відрізками знайдених кейсів, або оцінку максимальних покриттів еталонних кейсів відрізками знайдених кейсів.

Швидкодія паралельного алгоритму аналізу великих даних на матриці MDV дедуктивних векторів ставиться в лінійну залежність від числа бітів вхідних векторів або потужності універсуму примітивів.

Таким чином, запропоновано паралельні алгоритми in-memoгу комп'ютингу для економічних механізмів перетворення великих неструктурованих даних, як адрес, у корисні структуровані дані.

Запропоновано архітектуру in-memory computing із глобальним зворотним зв'язком та алгоритм матричної паралельної обробки великих даних, як адрес. Вона включає структуру матричного аналізу великих даних для визначення подібності між векторами, які надходять на входи матричного секвенсора. Векторний аналіз даних перетворюється на матричний комп'ютинг для обробки великих даних.

Розроблено метод ідентифікації патернів ключовими словами. Він характеризується використанням унітарно-кодovаних компонент даних для синтезу таблиці істинності бізнес-процесу. Це дозволяє використовувати read-write транзакції для паралельної обробки великих даних, як адрес.

	1	2	3	4	5	6	7	8	9
Gold	1-7	8-26	27-34	35-45	47-65	67-88	89-96	98-101	112-121
Real	1-8	9-25	27-33	34-46	48-64	65-89	90-97	99-105	107-123
Yield	7/9	17/18	6/7	10/12	17/18	21/24	6/8	2/6	9/16
$Q = \frac{1}{n} \sum_{i=1}^n Y_i$	0,76	0,77	0,94	0,85	0,83	0,94	0,88	0,75	0,33

Gold	1	1	1	1	2	2	2	2	2	3	3	3	4	4	4	4	4	5	5	5	5	5	
Real	1	1	1	1	1	2	2	2	3	3	3	3	4	4	4	4	5	5	5	6	6	6	6
x_3	4/5		3/5		2/5		1/3		2/3		2/5		3/5		5/5								
Y	4,8/5=0,96																						

Рисунок 14 – Локальні та інтегральні оцінки якості розпізнавання

6 ОБГОВОРЕННЯ

Завдяки тому, що матриця векторного аналізу визначена в метричному булевому просторі відношень n змінних $2^n \times 2^n$, така модель дозволяє вирішувати практично будь-яке комбінаторне завдання з ідентифікації, мінімізації, класифікації або кластеризації великих

даних. Паралельність формування всіх рішень для заданих великих даних робить запропонований механізм на пам'яті привабливим для ринку електронних технологій.

Швидкодія векторної технології обробки великих даних, як адрес, за допомогою запропонованого сек-

венсора визначається потужністю універсуму примітивів k (сумарна розрядність вхідних векторів) або числа екранів P (загальна кількість входів секвенсора), що підлягають обробці: $Q=kP/x$. При цьому апаратні витрати (кількість восьмивхідних елементів) на реалізацію каскадного секвенсора визначається числом екранів, що підлягають обробці: $N=P/x$, $X=6$, нижче наступній схемі. Ця оцінка також визначає кількість ітерацій движка, необхідних для паралельної обробки кадрів (8 патернів) вхідних даних.

Оцінка значимості координат таблиці істинності свідчить про те, що найзначніші координати визначаються стовпцях відмінності символів. Незначні координати перебувають у шпальтах подібності символів. Значимість координат стовпців зменшується зліва направо (рис. 15).

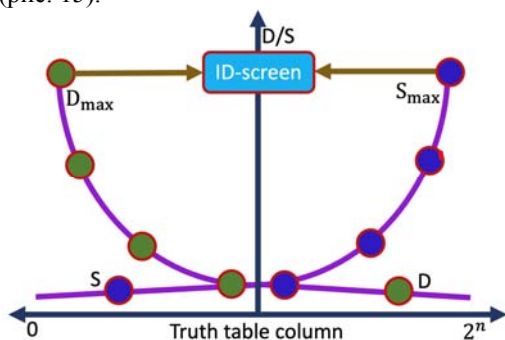


Рисунок 15 – Графіки подібності-відмінності по стовпцях таблиці істинності

У таблицях (див. рис. 14) наведено локальні та інтегральні оцінки автоматичного розпізнавання фрагментів та кейсів у бізнес-потоці даних. Насправді, оцінка покриття більше, ніж $Y=90\%$ вважається прийнятним результатом ідентифікації знайдених у потоці патернів.

Властивості програмного коду, що реалізує in-memory computing на логічних векторах за допомогою read-write транзакцій: точність ідентифікації патернів-екранів – 95%, точність ідентифікації класів еквівалентностей ключових даних – 97%, точність пошуку помилок оператора при виконанні картки функціональності – виконанні карти функціональності. Вхідний потік великих даних містить 10 тисяч екранів.

ВИСНОВКИ

Зроблено крок на шляху створення векторно-логічного in-memory комп'ютингу, який використовує лише read-write транзакції на адресній пам'яті [23, 26, 29, 30]. Показано переваги векторної моделі для компактного опису процесів, явищ, функцій та структур.

Наукова новизна полягає у розробці наступних інноваційних рішень:

1) запропоновано нову векторно-матричну технологію паралельної обробки великих даних, як адрес, яка характеризується використанням read-write

транзакцій на матричній пам'яті без використання процесорної логіки;

2) запропоновано архітектуру in-memory computing з глобальним зворотним зв'язком та алгоритм матричної паралельної обробки великих даних, як адрес;

3) запропоновано метод ідентифікації патернів ключовими словами, який характеризується використанням унітарно-кодованих компонентів даних для синтезу таблиці істинності бізнес-процесу, що дає можливість використовувати транзакцію read-write для паралельної обробки великих даних, як адрес.

Практична значимість дослідження полягає в тому, що будь-яку задачу штучного інтелекту (подібність-відмінність, класифікація-кластеризація та розпізнавання, ідентифікація образів) можна технологічно просто та ефективно вирішувати за допомогою таблиці істинності (або її похідних) та унітарно кодованих великих даних.

Перспективи дослідження пов'язані з імплементацією зехнології технології моделювання цифрових пристроїв на ринку EDA.

ЛІТЕРАТУРА

- Shiqiang Zhu. Intelligent Computing: The Latest Advances, Challenges, and Future [Electronic resource] / [Shiqiang Zhu, Ting Yu, Tao Xu et al.] // Intelligent Computing. – 30 Jan 2023. – Vol 2. – 45 p. – Article ID: 0006 – DOI: 10.34133/icomputing.0006. – Access mode: <https://spj.science.org/doi/10.34133/icomputing.0006>
- Hahanova A. Vector-Deductive Faults-As-Address Simulation / A. Hahanova // International Journal of Computing. – 2023. – №22(3). – P. 328–334. – <https://doi.org/10.47839/ijc.22.3.3227>
- Faggella Daniel What is Machine Learning? [Electronic resource] / Daniel Faggella. – February 26, 2020. – Access mode: <https://emerj.com/ai-glossary-terms/what-is-machine-learning/>.
- Bhardwaj R. Data analyzing using Map-Join-Reduce in cloud storage / R. Bhardwaj, N. Mishra, and R. Kumar // 2014 International Conference on Parallel, Distributed and Grid Computing. – Solan, India. – 2014. – P. 370–373. DOI: 10.1109/PDGC.2014.7030773.
- Hadoop Map Reduce Techniques: Simplified Data Processing on Large Clusters with Data Mining / [S. Suresh, T. Rajesh Kumar, M. Nagalakshmi et al] // 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). – Dharan, Nepal. – 2022. – P. 420–423. DOI: 10.1109/I-SMAC55078.2022.9986501.
- Friend Recommendation System Using Map-Reduce and Spark: A Comparison Study / A. M. A. Sai et al // 2023 4th International Conference on Innovative Trends in Information Technology (ICITIT). – Kottayam, India. – 2023. – P. 1–6. DOI: 10.1109/ICITIT57246.2023.10068723.
- Agarwal S. and Sinha R. MR-KClust: An efficient Map Reduce based clustering Technique / S. Agarwal and R. Sinha // 2022 6th International Conference on Electronics, Communication and Aerospace Technology – Coimbatore, India. – 2022. – P. 1460–1464. DOI: 10.1109/ICECA55336.2022.10009369.

8. Comparative analysis of Gaussian mixture model, logistic regression and random forest for big data classification using map reduce / [V. Singh, R. K. Gupta, R. K. Sevakula and N. K. Verma] // 2016 11th International Conference on Industrial and Information Systems (ICIIS). – Roorkee, India. – 2016. – P. 333–338. DOI: 10.1109/ICIINFS.2016.8262961.
9. Adaptively Accelerating Map-Reduce/Spark with GPUs: A Case Study / [K. R. Jayaram, A. Gandhi, H. Xin, and S. Tao] // 2019 IEEE International Conference on Autonomic Computing (ICAC). – Umea, Sweden. – 2019 – P. 105–114. DOI: 10.1109/ICAC.2019.00022.
10. Besimi N. Overview of data mining classification techniques: Traditional vs. parallel/distributed programming models / N. Besimi, B. Çiço and A. Besimi // 2017 6th Mediterranean Conference on Embedded Computing (MECO). – Bar, Montenegro. – 2017. – P. 1–4. DOI: 10.1109/MECO.2017.7977126.
11. Tang S. DynamicMR: A Dynamic Slot Allocation Optimization Framework for MapReduce Clusters / S. Tang, B.-S. Lee and B. He // IEEE Transactions on Cloud Computing. – 1 July-Sept. 2014. – Vol. 2, no. 3. – P. 333–347. DOI: 10.1109/TCC.2014.2329299.
12. Replication-Based Query Management for Resource Allocation Using Hadoop and MapReduce over Big Data / [A. Kumar, N. Varshney, S. Bhatiya and K. U. Singh] // Big Data Mining and Analytics. – December 2023. – Vol. 6, No. 4. – P. 465–477. DOI: 10.26599/BDMA.2022.9020026.
13. Usharani A. V. Secure EMR Classification and Deduplication Using MapReduce / A. V. Usharani and G. Attigeri // IEEE Access. – 2022. – Vol. 10. – P. 34404–34414. DOI: 10.1109/ACCESS.2022.3161439.
14. A Fundamental Tradeoff Between Computation and Communication in Distributed Computing / [S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr] // IEEE Transactions on Information Theory. – Jan. 2018. – Vol. 64, no. 1. – P. 109–128. DOI: 10.1109/TIT.2017.2756959.
15. Virtual Shuffling for Efficient Data Movement in MapReduce / [W. Yu, Y. Wang, X. Que, and C. Xu] // IEEE Transactions on Computers. – Feb. 2015. – Vol. 64, No. 2. – P. 556–568. DOI: 10.1109/TC.2013.216.
16. Shi X. Mammoth: Gearing Hadoop Towards Memory-Intensive MapReduce Applications / X. Shi et al. // IEEE Transactions on Parallel and Distributed Systems. – Aug. 2015. – Vol. 26, No. 8. – P. 2300–2315. DOI: 10.1109/TPDS.2014.2345068.
17. Luo W. Enterprise data economy: A hadoop-driven model and strategy / W. Luo // 2013 IEEE International Conference on Big Data. – Silicon Valley, CA, USA. – 2013. – P. 65–70. DOI: 10.1109/BigData.2013.6691690.
18. Wu J. Empirical Research on the Relationship between Logistics and Regional Economy Based on Big Data / J. Wu, Y. Chen, and Y. Chen // 2022 International Conference on Big Data, Information and Computer Network (BDICN). – Sanya, China. – 2022. – P. 12–15. DOI: 10.1109/BDICN55575.2022.00010.
19. Dai G. Research on Digital Economy Information System through Cloud Computing and Big Data Technology / G. Dai // 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA). – Dalian, China. – 2021. – P. 734–737. DOI: 10.1109/ICDSCA53499.2021.9650173.
20. Liu R. Placement of High Availability Geo-Distributed Data Centers in Emerging Economies / R. Liu, W. Sun, and W. Hu // IEEE Transactions on Cloud Computing. – July-Sept. 2023. – Vol. 11, No. 3 – P. 3274–3288. DOI: 10.1109/TCC.2023.3280983.
21. Approximate Clustering Ensemble Method for Big Data / [M. S. Mahmud, J. Z. Huang, R. Ruby et al], // IEEE Transactions on Big Data. – Aug. 2023. – Vol. 9, No. 4. – P. 1142–1155. DOI: 10.1109/TBDATA.2023.3255003.
22. Davis M. Emil Post's contributions to computer science / M. Davis // Proceedings of the Fourth Annual Symposium on Logic in Computer Science. – 1989. – P. 134–136.
23. Devadze D. Vector-Deductive Memory-Based Transactions for Fault-As-Address Simulation / D. Devadze, Z. Davitadze and A. Hahanova // 2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT). – 2022. – Athens, Greece. – P. 1–6. DOI: 10.1109/DESSERT58054.2022.10018769.
24. Kovalev I. S. Development of Computer System Components in Critical Applications: Problems, Their Origins and Solutions / [I. S. Kovalev, O. V Drozd, A. Rucinski et al.] // Herald of Advanced Information Technology. Publ. Nauka i Tekhnika – 2020. – Vol. 3, No. 4. – P. 252–262. DOI: 10.15276/hait.04.2020.4.
25. Hahanov V. Cyber Physical Computing for IoT-driven Services / V. Hahanov – New York : Springer, 2018. – 279 p. <https://doi.org/10.1007/978-3-319-54825-8>
26. Vector-Logic Synthesis Of Deductive Matrices For Fault Simulation / [W. Gharibi, A. Hahanova, V. Hahanov et al.] // Èlektronik modeling. – 2023. – № 45(2). – P. 16–33. doi.org/10.15407/emodel.45.02.016
27. Wroblewski F. J. Undecidability in the completion of truth-function logic / F. J. Wroblewski // Proceedings of the Twenty-First International Symposium on Multiple-Valued Logic. – 1991. – P. 225–229. DOI: 10.1109/ISMVL.1991.130734.
28. Liu H. A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis / H. Liu, X. Chen, and X. Liu // IEEE Access. 2022. – Vol. 10. – P. 32280–32289. DOI: 10.1109/ACCESS.2022.316017.
29. Векторно-логічне моделювання несправностей // [В. І. Хаханов, С. В. Чумаченко, С. І. Литвинова та ін.] // Радіоелектроніка, інформатика, управління. – 2023. – №2. – P. 37–51. DOI: 10.15588/1607-3274-2023-2-5.
30. Векторні моделі логіки і структури для тестування та моделювання цифрових схем / [Г. В. Хаханова, В. І. Хаханов, С. В. Чумаченко та ін.] // Радіоелектроніка, інформатика, управління. – 2021. – №3. – P. 69–85. Doi: 10.15588/1607-3274-2021-3-7.

Received 04.01.2024.
Accepted 28.02.2024.

UDC 681.326

IN-MEMORY INTELLIGENT COMPUTING

Hahanov V. I. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Abdullayev V. H. – PhD, Associate Professor of the Computer Engineering Department, Azerbaijan State University of Oil and Industry, Baku, Azerbaijan.

© Хаханов В. І., Абдуллаєв В. Х., Чумаченко С. В., Литвинова С. І., Хаханова І. В., 2024
DOI 10.15588/1607-3274-2024-1-15



Chumachenko S. V. – Dr. Sc., Professor, Head of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Lytvynova E. I. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Hahanova I. V. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. Processed big data has social significance for the development of society and industry. Intelligent processing of big data is a condition for creating a collective mind of a social group, company, state and the planet as a whole. At the same time, the economy of big data (Data Economy) takes first place in the evaluation of processing mechanisms, since two parameters are very important: speed of data processing and energy consumption. Therefore, mechanisms focused on parallel processing of large data within the data storage center will always be in demand on the IT market.

Objective. The goal of the investigation is to increase the economy of big data (Data Economy) thanks to the analysis of data as truth table addresses for the identification of patterns of production functionalities based on the similarity-difference metric.

Method. Intelligent computing architectures are proposed for managing cyber-social processes based on monitoring and analysis of big data. It is proposed to process big data as truth table addresses to solve the problems of identification, clustering, and classification of patterns of social and production processes. A family of automata is offered for the analysis of big data, such as addresses. The truth table is considered as a reasonable form of explicit data structures that have a useful constant – a standard address routing order. The goal of processing big data is to make it structured using a truth table for further identification before making actuator decisions. The truth table is considered as a mechanism for parallel structuring and packing of large data in its column to determine their similarity-difference and to equate data at the same addresses. Representation of data as addresses is associated with unitary encoding of patterns by binary vectors on the found universe of primitive data. The mechanism is focused on processorless data processing based on read-write transactions using in-memory computing technology with significant time and energy savings. The metric of truth table big data processing is parallelism, technological simplicity, and linear computational complexity. The price for such advantages is the exponential memory costs of storing explicit structured data.

Results. Parallel algorithms of in-memory computing are proposed for economic mechanisms of transformation of large unstructured data, such as addresses, into useful structured data. An in-memory computing architecture with global feedback and an algorithm for matrix parallel processing of large data such as addresses are proposed. It includes a framework for matrix analysis of big data to determine the similarity between vectors that are input to the matrix sequencer. Vector data analysis is transformed into matrix computing for big data processing. The speed of the parallel algorithm for the analysis of big data on the MDV matrix of deductive vectors is linearly dependent on the number of bits of the input vectors or the power of the universe of primitives. A method of identifying patterns using key words has been developed. It is characterized by the use of unitary coded data components for the synthesis of the truth table of the business process. This allows you to use read-write transactions for parallel processing of large data such as addresses.

Conclusions. The scientific novelty consists in the development of the following innovative solutions: 1) a new vector-matrix technology for parallel processing of large data, such as addresses, is proposed, characterized by the use of read-write transactions on matrix memory without the use of processor logic; 2) an in-memory computing architecture with global feedback and an algorithm for matrix parallel processing of large data such as addresses are proposed; 3) a method of identifying patterns using keywords is proposed, which is characterized by the use of unitary coded data components for the synthesis of the truth table of the business process, which makes it possible to use the read-write transaction for parallel processing of large data such as addresses. The practical significance of the study is that any task of artificial intelligence (similarity-difference, classification-clustering and recognition, pattern identification) can be solved technologically simply and efficiently with the help of a truth table (or its derivatives) and unitarily coded big data. Research prospects are related to the implementation of this digital modeling technology devices on the EDA market.

KEYWORDS: Intelligent Computing, Cloud, fog, and edge computing, Big data computing, In-memory computing, Cyber social competing, Hadoop Map-Reduce technique, big data as addresses, truth table, logical vector, similarities-differences, equivalence data, universe of primitives, patterns as a binary vector.

REFERENCES

1. Shiqiang Zhu, Ting Yu, Tao Xu, et al Intelligent Computing: The Latest Advances, Challenges, and Future [Electronic resource], *Intelligent Computing*, 30 Jan 2023, Vol. 2, 45 p. Article ID: 0006 DOI: 10.34133/icomputing.0006. Access mode: <https://spj.science.org/doi/10.34133/icomputing.0006>
2. Hahanova A. Vector-Deductive Faults-As-Address Simulation, *International Journal of Computing*, 2023, №22(3), pp. 328–334. <https://doi.org/10.47839/ijc.22.3.3227>
3. Daniel Faggella. What is Machine Learning? [Electronic resource], February 26, 2020. Access mode: <https://emerj.com/ai-glossary-terms/what-is-machine-learning/>.
4. Bhardwaj R., Mishra N., and Kumar R. Data analyzing using Map-Join-Reduce in cloud storage, *2014 International Conference on Parallel, Distributed and Grid Computing*. Solan, India, 2014, pp. 370–373. DOI: 10.1109/PDGC.2014.7030773.
5. Suresh S., Rajesh Kumar T., Nagalakshmi M., Bennilo Fernandes J. and Kavitha S. Hadoop Map Reduce Techniques: Simplified Data Processing on Large Clusters with Data Mining, *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. Dharan, Nepal, 2022, pp. 420–423. DOI: 10.1109/I-SMAC55078.2022.9986501.
6. Sai A. M. A. et al. Friend Recommendation System Using Map-Reduce and Spark: A Comparison Study, *2023 4th International Conference on Innovative Trends in Information Technology (ICITIT)*. Kottayam, India, 2023, pp. 1–6. DOI: 10.1109/ICITIT57246.2023.10068723.
7. Agarwal S. and Sinha R. MR-KClust: An efficient Map Reduce based clustering Technique, *2022 6th International*

- Conference on Electronics, Communication and Aerospace Technology*. Coimbatore, India, 2022, pp. 1460–1464. DOI: 10.1109/ICECA55336.2022.10009369.
8. Singh V., Gupta R. K., Sevakula R. K. and Verma N. K. Comparative analysis of Gaussian mixture model, logistic regression and random forest for big data classification using map reduce, *2016 11th International Conference on Industrial and Information Systems (ICIIS)*. Roorkee, India, 2016, pp. 333–338. DOI: 10.1109/ICIINFS.2016.8262961.
 9. Jayaram K. R., Gandhi A., Xin H., and Tao S. Adaptively Accelerating Map-Reduce/Spark with GPUs: A Case Study, *2019 IEEE International Conference on Autonomic Computing (ICAC)*. Umea, Sweden, 2019, pp. 105–114. DOI: 10.1109/ICAC.2019.00022.
 10. Besimi N., Çiço B. and Besimi A. Overview of data mining classification techniques: Traditional vs. parallel/distributed programming models, *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. Bar, Montenegro, 2017, pp. 1–4. DOI: 10.1109/MECO.2017.7977126.
 11. Tang S., Lee B.-S. and He B. DynamicMR: A Dynamic Slot Allocation Optimization Framework for MapReduce Clusters, *IEEE Transactions on Cloud Computing*, 1 July-Sept. 2014, Vol. 2, No. 3, pp. 333–347. DOI: 10.1109/TCC.2014.2329299.
 12. Kumar A., Varshney N., Bhatiya S. and Singh K. U. Replication-Based Query Management for Resource Allocation Using Hadoop and MapReduce over Big Data, *Big Data Mining and Analytics*, December 2023, Vol. 6, No. 4, pp. 465–477. DOI: 10.26599/BDMA.2022.9020026.
 13. Usharani A. V. and Attigeri G. Secure EMR Classification and Deduplication Using MapReduce, *IEEE Access*, 2022, Vol. 10, pp. 34404–34414. DOI: 10.1109/ACCESS.2022.3161439.
 14. Li S., Maddah-Ali M. A., Yu Q., and Avestimehr A. S. A Fundamental Tradeoff Between Computation and Communication in Distributed Computing, *IEEE Transactions on Information Theory*, Jan. 2018, Vol. 64, No. 1, pp. 109–128. DOI: 10.1109/TIT.2017.2756959.
 15. Yu W., Wang Y., Que X., and Xu C. Virtual Shuffling for Efficient Data Movement in MapReduce, *IEEE Transactions on Computers*, Feb. 2015, Vol. 64, No. 2, pp. 556–568. DOI: 10.1109/TC.2013.216.
 16. Shi X. et al. Mammoth: Gearing Hadoop Towards Memory-Intensive MapReduce Applications, *IEEE Transactions on Parallel and Distributed Systems*, Aug. 2015, Vol. 26, No. 8, pp. 2300–2315. DOI: 10.1109/TPDS.2014.2345068.
 17. Luo W. Enterprise data economy: A hadoop-driven model and strategy, *2013 IEEE International Conference on Big Data*. Silicon Valley, CA, USA, 2013, pp. 65–70. DOI: 10.1109/BigData.2013.6691690.
 18. Wu J., Chen Y., and Chen Y. Empirical Research on the Relationship between Logistics and Regional Economy Based on Big Data, *2022 International Conference on Big Data, Information and Computer Network (BDICN)*. Sanya, China, 2022, pp. 12–15, DOI: 10.1109/BDICN55575.2022.00010.
 19. Dai G. Research on Digital Economy Information System through Cloud Computing and Big Data Technology, *2021 IEEE International Conference on Data Science and Computer Application (ICDSCA)*. Dalian, China, 2021, pp. 734–737. DOI: 10.1109/ICDSCA53499.2021.9650173.
 20. Liu R., Sun W., and Hu W. Placement of High Availability Geo-Distributed Data Centers in Emerging Economies, *IEEE Transactions on Cloud Computing*, July-Sept. 2023, Vol. 11, No. 3, pp. 3274–3288. DOI: 10.1109/TCC.2023.3280983.
 21. Mahmud M. S., Huang J. Z., Ruby R., Ngueilbaye A. and Wu K. Approximate Clustering Ensemble Method for Big Data, *IEEE Transactions on Big Data*, Aug. 2023, Vol. 9, No. 4, pp. 1142–1155. DOI: 10.1109/TBDDATA.2023.3255003.
 22. Davis M. Emil Post's contributions to computer science, *Proceedings of the Fourth Annual Symposium on Logic in Computer Science*, 1989, pp. 134–136.
 23. Devadze D., Davitadze Z. and Hahanova A. Vector-Deductive Memory-Based Transactions for Fault-As-Address Simulation, *2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 2022. Athens, Greece, pp. 1–6. DOI: 10.1109/DESSERT58054.2022.10018769.
 24. Kovalev I. S., Drozd O. V., Rucinski A., Drozd M. O., Antoniuk V. V., Sulima Y. Y. Development of Computer System Components in Critical Applications: Problems, Their Origins and Solutions, *Herald of Advanced Information Technology. Publ. Nauka i Tekhnika*, 2020, Vol. 3, No.4, pp. 252–262. DOI: 10.15276/hait.04.2020.4.
 25. Hahanov V. Cyber Physical Computing for IoT-driven Services. New York, Springer, 2018, 279 p. <https://doi.org/10.1007/978-3-319-54825-8>
 26. Gharibi W., Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Hahanov I. Vector-Logic Synthesis Of Deductive Matrices For Fault Simulation, *Elektronik modeling*, 2023, № 45(2), pp. 16–33. doi.org/10.15407/emodel.45.02.016
 27. Wroblewski F. J. Undecidability in the completion of truth-functional logic, *Proceedings of the Twenty-First International Symposium on Multiple-Valued Logic*, 1991, pp. 225–229. DOI: 10.1109/ISMVL.1991.130734.
 28. Liu H., Chen X., and Liu X. A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis, *IEEE Access*, 2022, Vol. 10, pp. 32280–32289. DOI: 10.1109/ACCESS.2022.316017.
 29. Hahanov V., Chumachenko S., Litvinova Y., Hahanova I., Khakhanova A., Shkil A., Rakhlis D., Hahanov I., Shevchenko O. Vector-Logical Fault Simulation, *Radio Electronics, Computer Science, Control*, 2023, №2, pp. 37–51. <https://doi.org/10.15588/1607-3274-2023-2-5>
 30. Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Rakhlis D. Vector-Driven Logic And Structure For Testing And Deductive Fault Simulation, *Radio Electronics, Computer Science, Control*, 2021, No. 3, pp. 69–85. <https://doi.org/10.15588/1607-3274-2021-3-7>.