УДК 004

В.Ленецький, магістр гр. КН-23М
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ МАШИННОГО НАВЧАННЯ У BIG DATA

У статті розроблено програмне забезпечення, яке призначено для системи машинного навчання у Big Data. Метою розробки є дослідження та програмна реалізація системи машинного навчання у Big Data. Об'єктом дослідження є процес машинного навчання у Big Data. Предметом дослідження є методи машинного навчання у Big Data. Методи дослідження базуються на методах машинного навчання та обробки Big Data, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи машинного навчання у Big Data. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Постановка проблеми. Один з можливих сценаріїв застосування машинного навчання (МН) у системах зберігання даних (СЗД) – забезпечення якості обслуговування (QoS) на рівні додатків.

Однією з перспективних областей застосування машинного навчання можна назвати зберігання даних. Розумні пристрої зберігання усе ще залишаються концепцією майбутнього, тієї самої Next Big Thing, що так чекають експерти. Такі пристрої зможуть «розуміти» поводження додатків на основі специфічних факторів, визначати рівень вимог до доступності й надійності систем зберігання й автоматично підбудовуватися під необхідний клас рішень.

Говорячи про машинне навчання (МН), багато хто мають на увазі цілий ряд «забавних» сценаріїв, від яких дуже мало практичної користі. Як правило, мова йде про генератори текстів, які на перевірку виявляються нісенітницею, про автоматизовані сценарії для кіно, імітації стилю живопису старих майстрів і інших сумнівних проектів. Інформаційний шум навколо такої активності формує неправильне уявлення про машинне навчання й штучний інтелект, у те час як існують цілком конкретні області їхнього застосування в науці, техніку й в інформаційних технологіях.

У першу чергу машинне навчання допомагає інженерам автоматизувати ту або іншу інтелектуальну діяльність, адже комп'ютер здатний обробляти великий обсяг даних за менший час і швидше приймати конкретні рішення. Безумовно, скасувати посаду охоронця правопорядку й замінити її штучним інтелектом не вдасться, але цілком можливо застосувати потрібний алгоритм для точного розпізнавання осіб підозрюваних у великому потоці людей на жвавій вулиці.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи машинного навчання у Big Data.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи машинного навчання у Big Data.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем машинного навчання у Big Data.
- Дослідження системи машинного навчання у Big Data.
- Програмна реалізація системи машинного навчання у Big Data.

Об'єктом дослідження є процес машинного навчання у Big Data.

Предметом дослідження є методи машинного навчання у Big Data.

Методи дослідження базуються на методах машинного навчання та обробки Big Data, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Алгоритми машинного навчання усе ширше використовуються для збільшення продуктивності гібридних систем зберігання даних. Класифікації, регресії й аналіз часових рядів допомагають вибрати тип сховища й упорядкувати в ньому розміщення файлів.

У число чотирьох головних експериментів, проведених на Великому адронному коллайдері в ЦЕРН, входить LHCb. Його детектори й системи моделювання фізичних процесів щорічно генерують 15 екзабайт сирих даних, які після обробки містяться в розподілену систему зберігання на жорстких дисках і магнітних стрічках. Зрозуміло, що диски швидше стрічок, але дорожче, тому важливо оптимізувати розміщення файлів на різних носіях. Система керування дисковою пам'яттю для LHCb заснована на статистичному аналізі історії звертання до даних, що поряд з алгоритмами машинного навчання дає можливість прогнозувати популярність файлів, що дозволяє визначити, які файли можна видалити з диска. А застосовуючи алгоритми регресійного аналізу й аналізу часових рядів, вдається визначити оптимальне число копій файлів на диску для забезпечення їхнього надійного зберігання, економії дискового простору й зменшення часу доступу до даних.

Алгоритми машинного навчання вже давно застосовуються для поліпшення продуктивності алгоритмів роботи з кеш-пам'яттю, наприклад в [1] для прогнозування популярності даних використовувалися ланцюги Маркова, а опираючись на цей прогноз, удалося домогтися збільшення продуктивності гібридної (SSD + HDD) системи зберігання. В [2] прогноз популярності файлів виконувався за допомогою нейронної мережі й дозволив визначити оптимальне число копій файлів.

Особливість системи керування дисковою пам'яттю для LHCb полягає в тому, що історія звертань до файлів сильно розріджена – часовий ряд, що відбиває факти звертань до дисків, має багато нульових значень. Більшість файлів використовуються рідко, але інтенсивно, тому такі методи, як штучні нейронні мережі [3], ARMA, ARIMA [4], демонструють погані результати через сильну розрідженість даних. Обчислення додаткових ознак для кожного файлу, що характеризують особливості їхньої історії звертань, дозволяє розділити файли на популярні й непопулярні.

Вхідними даними для алгоритму керування дисковою пам'яттю для LHCb є історія звертань до файлів і метадані, до яких ставляться: джерело даних, конфігурація детектора, тип файлу, тип даних (отримані по методу Монте Карло або реальні дані), тип події, час створення файлу, час першого/останнього обігу, розмір однієї копії, загальний розмір файлу на диску, число копій на диску й ін. Наприклад, можна використовувати історію звертання до файлів за два роки з розбивкою по тижнях і представити її у вигляді часового ряду з 104 точок, кожна з яких – це число звертань до файлу за один тиждень.

Для прогнозу популярності файлів використовується класифікатор – алгоритм машинного навчання із учителем, у результаті роботи якого кожний файл позначається як популярний або непопулярний. Як ми вже відзначали, часові ряди історії звертань до файлів сильно розріджені, тому показники за останні півроку (26 тижнів) історії звертань служать для розмітки даних – якщо файл протягом цього періоду не використовувався, то він позначається як непопулярний і одержує мітку «1», у противному випадку позначається як «0».

Метадані використовуються як вхідні ознаки для класифікатора, однак для уточнення алгоритму були обчислені нові ознаки, які використовувалися разом з існуючими. Ці нові ознаки описують форму часового ряду для історії звертань до файлу – наприклад, якщо останні півроку історії звертань використовуються для розмітки файлів, то для обчислення нових ознак беруться тільки перші 78 тижнів. Такий вибір часових інтервалів обумовлений тим, що дані зберігаються й використовуються роками, а між двома послідовними звертаннями до файлу можуть пройти тижні й місяці.

Один з ознак – кількість тижнів, протягом яких були звертання до файлу, іншої – число тижнів з тих пор, як до файлу зверталися востаннє. Також обчислювалися максимальне значення, середнє значення й стандартне відхилення числа тижнів між послідовними тижнями з ненульовим числом звертань і їхні відносини. Ще одна ознака – центр мас часового ряду файлу, у якому маса вказує на число звертань до файлу за кожній тиждень, а координата – на номер тижня. Всі ці ознаки істотно підняли якість навчання класифікатора.

У ролі класифікатора використовувався градієнтний бустинг над деревами (алгоритм машинного навчання, заснований на застосуванні ансамблю дерев рішень), що дозволяє без перенавчання швидко одержати високу якість класифікації. Для навчання використовувався метод перехресної перевірки з 10 частинами (k-fold cross-validation). Це означає, що всі файли розбиваються на 10 частин, при цьому класифікатор навчається на дев'яти частинах даних, а потім використовується для пророкування ймовірності одержання мітки «1» для десятої частини даних. Процедура повторюється 10 разів. У підсумку одержуємо прогноз імовірності мітки «1» для кожної з 10 частин. Передвіщена ймовірність перетвориться в популярність так, щоб популярність для класу з міткою «1» була розподілена рівномірно. Чим ближче популярність до 1, тим вище ймовірність того, що файл не буде використаний у майбутньому. Тобто обчислена популярність визначає антипопулярність файлу. Такий вибір зроблений для того, щоб файли, що є першими кандидатами на видалення, мали більшу величину метрики.

Популярність даних, рівна 1, відбиває ймовірність того факту, що файл буде марний у майбутньому, а другою важливою характеристикою є інтенсивність звертання до файлу. Існує безліч алгоритмів аналізу часових рядів для пророкування їхніх майбутніх значень, але знову, з огляду на, що більшість часових рядів сильно розріджено, складні параметричні методи, такі як поліноміальна регресія, авторегресія, ARMA, ARIMA, штучні нейронні мережі й інші, не підходять. Тому для прогнозу інтенсивностей звертань до файлів використовувалися дві непараметричні моделі.

Спочатку часові ряди згладжувалися по формулі ядерного згладжування Надарая-Ватсона на основі RBF-ядра (один з найпростіших видів непараметричної регресії), а ширина вікна згладжування була взята в 30 тижнів.

Найпростіший спосіб пророчити майбутнє значення часового ряду – це взяти в його якості значення з останньої точки спостереження (статична модель прогнозу в роботі).

Знаючи популярність і передвіщене значення інтенсивності звертань до файлів, можна визначити, які файли повинні зберігатися на жорстких дисках і скільки копій вони повинні мати. При цьому небажано видаляти з диска файли, які будуть затребувані в майбутньому. Більше того, має сенс зберігати найбільш затребувані файли з більшим числом копій, щоб зменшити середній час доступу до даних. Наявність додаткових копій дозволяє забезпечити декільком користувачам швидкий доступ до даних зі збереженням пропускної здатності, так як копії можуть бути розміщені фізично близько до місць, де в них бідують.

Для оптимізації розподілу даних використовувалася функція втрат, що дозволяє оцінити вартість зберігання всіх даних на дисках і стрічках, а також вартість відновлення даних з магнітних стрічок у випадку помилки.

Оптимальне число копій файлів визначається по передвіщеній інтенсивності звертань до файлів – чим вище інтенсивність звертань, тим більше копій має конкретний файл.

Якщо антипопулярність файлу вище граничного значення, то файл віддаляється з диска. Оптимізація функції втрат полягає в тому, щоб знайти такі граничні значення популярності даних і оптимальні значення копій файлів, які дають мінімум функції втрат.

Можна зрівняти запропонований алгоритм роботи рекомендаційної системи з алгоритмом Least Recently Used (LRU), що дивиться на останні спостереження в історії звертань до файлу й ухвалює рішення щодо видалення файлів з диска.

«Відношення часу доступу» – це відношення часу доступу (завантаження) до файлів, отримане після застосування алгоритму, до первісного часу доступу. Колонка «Число

помилок» показує, скільки файлів було вилучено з жорсткого диска, але потім знадобилося в роботі [2]. «Альфа» – коефіцієнт пропорційності між числом копій файлу й квадратним коренем інтенсивності звертання до нього. Як видно з таблиці, обидва алгоритми дозволяють оптимізувати обсяг дискового простору, однак LRU допускає більше помилок.

Як показує досвід, для багатьох сховищ характерно неоптимальне розміщення даних по різних рівнях. Дійсно, важко заздалегідь передбачити, як будуть використовуватися файли, у результаті затребувані дані можуть виявитися, наприклад, на стрічці. Запропонований алгоритм керування дисковою пам'яттю в гібридній системі зберігання, побудований на основі машинного навчання, допускає невелике число помилкових видалень файлів, дозволяє домогтися економії дискового простору й знизити середній час доступу до даних.

Розробка структурної схеми

На жаль або на щастя, про самостійну творчість машин у незапрограмованій заздалегідь ситуації мова не йде. Система виявляється марною, якщо наявних даних недостатньо або зафіксована подія не співвідноситься з якими-небудь попередніми аналогами. До того ж універсальних алгоритмів не існує – кожний з них вирішує одне певне завдання.

Розумні пристрої зберігання

Однієї з перспективних областей для застосування машинного навчання є зберігання даних. Перераховуючи потенційні напрямки нового технологічного прориву в сфері СЗД, експерти називають технології RAIN, SSD, NVMe, хмарні інфраструктури й програмно обумовлене зберігання (SDS), але навіть не згадують машинне навчання в СЗД. Імовірно, це пов'язане з тим, що відповідні методи ще мало вивчені й не до кінця освоєні лідерами ринку.

Поширення флеш-накопичувачів і розвиток програмно обумовленого зберігання – важливі фактори, але вони аж ніяк не новина. На відміну від названих технологій, розумні пристрої зберігання багато в чому залишаються концепцією майбутнього, тої самої Next Big Thing, що так чекають експерти.

Що ж будуть робити розумні пристрої? Вони покликані не тільки зберігати інформацію, але й виконувати інтелектуальні аналітичні дії, забезпечуючи роботу мікросервісів і еластичну масштабованість. В ідеалі такі пристрої повинні мати здатність міняти свої ролі залежно від контексту й при необхідності поєднуватися для виконання спільного завдання.

Розумні пристрої зможуть «розуміти» поводження додатків на основі специфічних факторів, визначати рівень вимог до доступності й надійності систем зберігання й автоматично підбудовуватися під необхідний клас рішень. Пристрою нового типу допоможуть вирішити цілий ряд проблем, що виникають у світі розумних міст, підприємств і просторів.

Які основні сценарії використання машинного навчання вже зараз і розумних пристроїв зберігання в майбутньому? Зупинимося на них більш докладно.

Сценарії застосування

Перший і один з головних – предіктивна аналітика на основі аналізу поводження СЗД. Відповідний алгоритм покликаний здійснювати аналіз системних журналів і історії подій і попереджати про можливі проблеми с кластером зберігання. Рано або пізно буде досить простого голосового запиту до системи підприємства, щоб моментально одержати дані про стан інфраструктури, проаналізувати поточне навантаження, виділити й розподілити потрібні для співробітників ресурси.

Не виключено, що при збереженні поточних темпів розвитку мобільних технологій і автоматизації через п'ять-сім років засобу керування інфраструктурою навчаться розуміти природна мова. Уже сьогодні доступні аналоги Microsoft IFTTT, які адекватно реагують на завдання «Транслювати всі мої повідомлення з Twitter в Facebook». Якщо говорити про конкретні приклади, то на шляху до розумного центра обробки даних істотно просунулася компанія Nimble Storage із продуктом InfoSight.

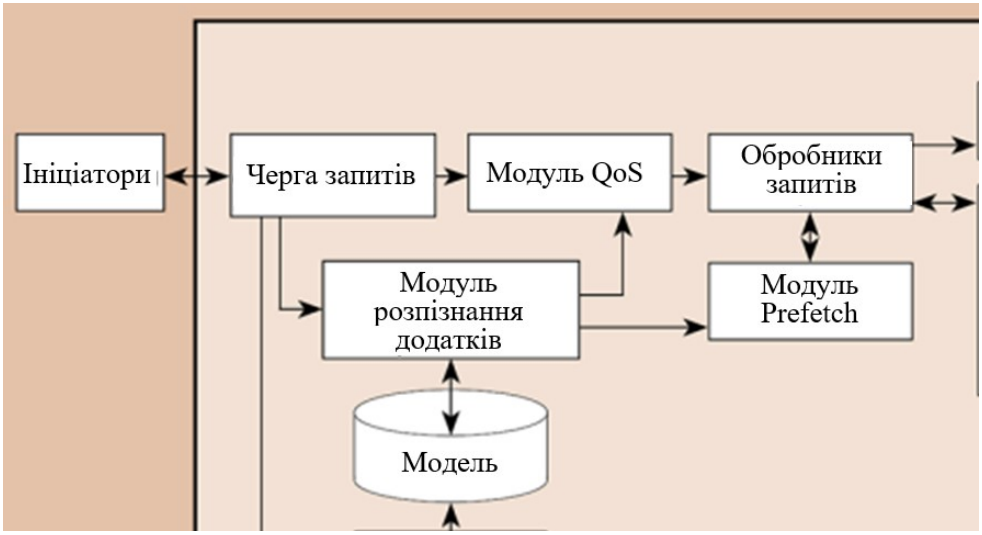

Рисунок 1 – Структурна схема системи

Другий важливий сценарій – зміна параметрів зберігання на лету з модифікацією настроювань СЗД у цілому й окремих компонентах зокрема. Багато хто вендори з різним успіхом розробляли свого роду «автопілот», що на основі аналізу наявних даних і поведінкових характеристик міг би автоматично змінювати параметри системи зберігання, наприклад, вибираючи оптимальні настроювання для переважного типу навантаження.

Полем для експериментів по реалізації такого алгоритму стали флеш-накопичувачі. Через специфіку NAND Flash, при розробці рішень для SSD в одному такому рішенні неможливо сполучити великий обсяг, високу надійність і максимальне число перезаписів, через що доводиться йти на певні компроміси. Настроювання пристрою і його поводження можна змінити шляхом завдання параметрів регістрів (регістрів може бути більше 1000 в 3D NAND і до 100 у планарній пам'яті). Удалий приклад машинного навчання для автоматичної модифікації параметрів різних регістрів – рішення технологічного стартапа NVMdurance.

Третій приклад застосування МН в СЗД – забезпечення якості обслуговування (QoS) на рівні додатків. Усе більше розроблювачів доходять висновку, що традиційний підхід «один додаток для одного LUN» не оптимальний і вимагає перегляду. Дійсно, нерідкі випадки, коли з одним томом з одного хосту взаємодіють багато додатків, різні по функції й навантаженню. Левова частина цих додатків не критична для діяльності підприємства, тобто ресурси системи витрачаються нераціонально.

Як не заблудитися в «випадковому лісі»

Для рішення цієї проблеми були реалізовані проекти, у рамках яких розпізнавання додатків і їхнього навантаження здійснюється по характерних операціях уведення-виводу (IO). У сфері Data Mining є кілька алгоритмів, придатних для подібного завдання ідентифікації – наприклад, Random Forest, або «випадковий ліс». Цей метод заснований на побудові великого числа (ансамблю) «дерев» рішень (їхня кількість є параметром методу). У порівнянні з іншими аналогічними алгоритмами Random Forest має наступні переваги:

– висока швидкість навчання;
– неітеративне навчання (алгоритм завершується за фіксоване число операцій);
– масштабованість (здатність обробляти більші обсяги даних);
– висока якість одержуваних моделей (порівнянне з нейронними мережами й ансамблями нейронних мереж);
– мала кількість параметрів, що налаштовуються.

Random Forest – імовірнісний алгоритм. Коли до системи зберігання звертається невідомий додаток, він оцінює ймовірність збігу даного додатка з раніше виявленими в системі. При цьому в алгоритму є й недоліки. Один з них – схильність до перенавчання на

зашумлених даних. Втім, ця проблема вирішується за допомогою застосування спеціалізованих фільтрів на зразок FCBF.

Реалізацію функції QoS зроблено в модулю QoSmic. Принцип його функціонування досить простий. Всі запити до СЗД проходять через модуль розпізнавання додатків. Модуль працює у двох режимах:

– Навчання: система зберігання вивчає характеристики нового додатка, що планується розпізнавати з метою забезпечення QoS або читання, що попереджає.

– Розпізнавання: додатки ідентифікуються в режимі реального часу, а інформація про їх використовується модулями QoS і Prefetch.

Протягом певного періоду часу (наприклад, 20 с) запити накопичуються в модулі розпізнавання, далі журнал аналізується й на підставі зібраних відомостей будуються сигнатури вводу-виводу. У режимі навчання сигнатури позначаються ім'ям додатка, а в режимі розпізнавання передаються у відповідний модуль для ідентифікації.

Для ідентифікації додатка досить знати чотири характеристики: довжину запиту, тип запиту (читання або запис), зсув (адресний простір), час надходження запиту. На підставі цих параметрів і будуються сигнатури I/O.

Споконвічно при ідентифікації додатків акцент робився на пошуку відомих «паттернів» (послідовності довжин запитів, що йдуть один за іншим). Даний підхід відмінно зарекомендував себе раніше у випадку порівняльних тестів (benchmark), ПЗ для резервного копіювання й антивірусів. Тестувались різні алгоритми машинних навчань, спочатку вдавалося ідентифікувати додатка з низькою ймовірністю, але з додаванням у сигнатуру уведення-виводу спеціальних атрибутів точність ідентифікації вдалося довести до 99,9%.

У деяких системах головними критеріями для ідентифікації є адресний простір. У випадку QoSmic передбачається, що з одним простором можуть працювати кілька додатків; таким чином, розпізнавання по місцю розташування не здійснюється. Ключовим же критерієм для ідентифікації виявився розподіл довжин запитів, що надходять від конкретного додатка.

Розроблений метод ідентифікації має високу точність і швидкістю, що дозволяє використовувати його для автоматичного завдання пріоритету критично важливим додаткам і гарантувати їм необхідну пропускну здатність поза залежністю від навантаження з боку інших працюючих додатків. Високий рівень точності досягається завдяки певним атрибутам, тобто параметрам у сигнатурі уведення-виводу, що дозволяє формувати досить точний статистичний профіль різних додатків, з високою точністю виявляти працююче й відрізняти його від низькопріоритетного.

Безумовно, адміністратор повинен брати до уваги необхідність періодичного перенавчання системи. Особливо цим не варто зневажати, якщо часто видається повідомлення «не вдалося визначити» або додатки визначаються неправильно. У процесі навчання сам алгоритм може підказати, що отриманих сигнатур недостатньо для точної ідентифікації.

Від навчання до роботи

Автоматичне визначення працюючого додатка на ініціаторі дозволяє застосовувати технологію QoSmic у будь-якій системі зберігання, де є модуль QoS, відповідальний за різні рівні якості обслуговування. Сам же спосіб забезпечення QoS може бути реалізований по-різному. Наприклад, у рішеннях компаній Fujitsu або Oracle рівень QoS визначається адміністратором вручну. Ці СЗД створюють профілі на основі співвідношення операцій читання-запису й типів навантаження, при цьому розпізнавання конкретних додатків не відбувається.

Схожа ідея застосування машинного навчання викладена в патенті US8762583 компанії EMC. Вона припускає використання нейронної мережі, для того щоб визначити оптимальний спосіб обробки вхідних запитів вводу-виводу. Даний підхід може бути використаний і в модулі QoS для балансування навантаження на систему.

Таким чином, методи машинного навчання можуть із успіхом застосовуватися в СЗД, не викликаючи додаткових затримок і втрат продуктивності. У результаті вузьким місцем у системі зберігання виявляється не обчислювальний модуль, а швидкість читання із самих дисків, тобто обмеження встаткування.

Завершуючи огляд прикладів впровадження машинного навчання в системах зберігання даних, відзначимо, що роль «науки про даний» і самих фахівців з аналізу даних (data scientist) в адмініструванні IT-інфраструктури може згодом істотно зменшитися. Можливо, у майбутньому не залишиться адміністраторів як таких, оскільки через подальший розвиток алгоритмів машинного навчання й штучного інтелекту залученість людини в керування інфраструктурою буде зведена до мінімуму.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів машинного навчання у Big Data.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем машинного навчання у Big Data.

– Досліджена система машинного навчання у Big Data.

– На основі отриманих результатів досліджень створена програмна реалізація системи машинного навчання у Big Data.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання машинного навчання у Big Data. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». 8th International Symposium on Intelligent Informatics, ISI 2023, 2025. vol 389. pp 377-389. Springer, Singapore.
2. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». CEUR Workshop Proceedings, 2023, 3628, pp. 106-115.
3. Smirnov O., Fedorov E., Neskorodieva A., Neskorodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». CEUR Workshop Proceedings Volume 3664, 2024, Pages 11-23.
4. Malyukov V., Bebeshko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». Lecture Notes in Networks and Systems, 2023, 729 LNNS, pp. 104–112.
5. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». Advanced Information Systems, 2023, 7(2), pp. 49-56.
6. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». CEUR Workshop Proceedings, Volume 3530, 2023, pp. 256-265.
7. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». CEUR Workshop Proceedings, Volume 3312, 2022, pp. 47-58.
8. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12.
9. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.
10. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34.

11. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., Smirnova, T. «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems. Volume 152, 2021, Pages 85-98.
12. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w.
13. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207.
14. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.
15. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.
16. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
17. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.
18. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.
19. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.
20. Smirnov O. Kuznetsov A., Zaicenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.
21. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.
22. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.
23. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019.
24. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.
25. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.