

УДК 004

Д. Татаров, магістр гр. КН-24М,

*Центральноукраїнський національний технічний університет*

## ДОСЛІДЖЕННЯ ТА ПРИНЦИПИ ПОБУДОВИ СИСТЕМИ ВИЗНАЧЕННЯ КОНТЕКСТУ КОРИСТУВАЧА НА ОСНОВІ АРХІТЕКТУРИ CoDA

У статті розроблено програмне забезпечення, яке призначено для системи визначення контексту користувача на основі архітектури CoDA. Метою розробки є дослідження та принципи побудови системи визначення контексту користувача на основі архітектури CoDA. Об'єктом дослідження є процес визначення контексту користувача на основі архітектури CoDA. Предметом дослідження є методи визначення контексту користувача на основі архітектури CoDA. Методи дослідження базуються на методах теорії комп'ютерних мереж, теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи визначення контексту користувача на основі архітектури CoDA. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

### **визначення контексту користувача, CoDA**

**Постановка проблеми.** Gartner визначає CoDA. (Context delivery architecture, Архітектура доставки контексту) як архітектурний стиль, який спирається на стилі взаємодії та розбиття сервіс-орієнтовані архітектури (SOA) та архітектури, керованої подіями (EDA), і додає формальні механізми елементів програмного забезпечення, які відкривають та застосовують контекст користувача в реальному часі. CoDA. надає структуру архітекторам рішень, що дозволяє їм визначати та реалізовувати компоненти технології, інформації та процесів, які дозволяють службам використовувати контекстну інформацію для підвищення якості взаємодії з користувачем. Технології можуть включати контекстних брокерів, моніторів стану, датчиків, аналітичних движків та хмарних двигунів обробки транзакцій.

До того, як з'явився CoDA, існувала SOA (Service-Oriented Architecture), яка являє собою набір принципів та методологій для проектування та розробки програмного забезпечення у формі сумісних служб.

Реалізація SOA покладається на сітці неузгодженого програмного забезпечення, що є вільно пов'язаними одиницями функціональності, які не мають прямого зв'язку один з одним. Кожна служба керує конкретним та незалежним фрагментом функціональності та використовує визначені протоколи, які описують, як передавати інформацію між собою.

SOA покладається на сервісну орієнтацію як на основний принцип проектування. Якщо служба представляє собою простий інтерфейс, який вичерпує його основну складність, користувачі можуть отримати доступ до незалежних служб без знання реалізації платформи служби, включаючи основну операційну систему, мову програмування або апаратне забезпечення.

SOA допомагає підприємствам реагувати швидше та економічно ефективніше на мінливі ринкові умови шляхом заохочення повторного використання на рівні сервісу, а не на мікро (кодовому) рівні.

CoDA або Архітектура доставки контексту забезпечує контекстні послуги, які з'єднуються в рамках SOA. В результаті постачальники послуг, постачальники послуг мережі та інтегратори програмного забезпечення можуть розвивати контекст.

**Аналіз останніх досліджень і публікацій.** При аналізі останніх досліджень і публікацій [1-30] було виявлено певні прогалини у забезпеченні системи визначення контексту користувача на основі архітектури CoDA.

**Мета й завдання дослідження.** Метою роботи є дослідження та принципи побудови системи визначення контексту користувача на основі архітектури CoDA.

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

- Огляд існуючих систем визначення контексту користувача на основі архітектури CoDA.
- Дослідження системи визначення контексту користувача на основі архітектури CoDA.
- Програмна реалізація системи визначення контексту користувача на основі архітектури CoDA.

*Об'єктом дослідження* є процес визначення контексту користувача на основі архітектури CoDA.

*Предметом дослідження* є методи визначення контексту користувача на основі архітектури CoDA.

*Методи дослідження* базуються на методах теорії комп'ютерних мереж, теорії захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

#### **Виклад основного матеріалу.**

Архітектура доставки контексту (CoDA) забезпечує індивідуальний цифровий досвід у режимі реального часу, узгоджуючи дані, наміри та доставку по каналах. По суті, CoDA узгоджує те, чого хоче користувач – прямо зараз – з тим, як цей досвід доставляється через цифрові точки контакту. Замість того, щоб покладатися на статичний контент або заздалегідь визначені шляхи, ця архітектура динамічно адаптує відповіді на основі контексту користувача – місцезнаходження, пристрою, поведінки, уподобань – та передбачуваного наміру кожної взаємодії.

У сучасній гіперперсоналізованій екосистемі користувачі очікують більшого, ніж просто доступності. Вони вимагають актуальності, негайності та інтуїтивно оптимізованого досвіду в веб-, мобільних, Інтернеті речей та підключених середовищах. Від електронної комерції до платформ цифрової охорони здоров'я, здатність надавати персоналізований контент на основі намірів, контексту та стану мережі виділяє лідерів галузі.

У даній роботі розглядаються фундаментальні концепції, що формують CoDA, починаючи з визначення контексту та наміру користувача, переходячи до інтелектуальних механізмів доставки та завершуючи роллю якості обслуговування (QoS) у забезпеченні безперебійного виконання. Попутно розглядаються основні технології, такі як периферійні обчислення, оркестрація на основі штучного інтелекту, адаптивні механізми рендерингу, мережі доставки досвіду та брокери контексту. Готові побачити, як інфраструктура поєднується з інтелектом? Давайте зазирнемо всередину механізму.

#### **Стратегічне значення контексту в сучасних цифрових застосунках**

##### **Аналіз контексту: що це насправді означає в дизайні застосунків**

Контекст стосується ситуативних факторів, які формують взаємодію користувача з цифровим продуктом. До них належать місцезнаходження користувача, характеристики пристрою (такі як розмір екрана, операційна система або обчислювальна потужність), часові сигнали, такі як час доби або день тижня, та моделі поведінки, такі як історія переглядів, попередні покупки або навігаційні шляхи в додатку. Додайте визначені користувачем налаштування, такі як мовні налаштування або налаштування сповіщень, і цифрове середовище отримує рівні значення та наміру.

Кожна змінна створює сигнал, який, будучи зареєстрованим у режимі реального часу та точно інтерпретованим, перетворює статичний цифровий досвід на адаптивне середовище. Користувач, який переглядає роздрібний додаток на телефоні під час поїздки на роботу та з

роботи, демонструє інші моделі взаємодії, ніж користувач, який використовує комп'ютер у робочий час. Контекст визначає різницю між наданням загального інтерфейсу та пропонуванням конкретної відповіді, побудованої на спостережуваній потребі.

#### **Від статичного до динамічного: чому контекст визначає релевантність**

Релевантність не виникає випадково. Системи повинні реагувати на зміну вхідних даних, щоб відповідати динамічним очікуванням. Цифрові додатки, оснащені контекстною усвідомленістю, коригують контент, функціональність та робочі процеси у відповідь на мінливе середовище.

– Push-сповіщення плануються на основі місцевого часу та вікон сприйняття поведінки.

– Рекомендації щодо контенту розвиваються синхронно з послідовностями перегляду та станами виконання завдань.

– Потоки автентифікації змінюються залежно від цілісності мережі або стану пристрою.

Кожен контекстний вхідний сигнал змінює ймовірність того, що користувач взаємодітиме, здійснить конверсію або повернеться. Замість того, щоб покладатися на статичні правила, додатки, що враховують контекст, використовують ці сигнали для прийняття рішень у режимі реального часу, що безпосередньо призводить до підвищення операційної ефективності та більш інтуїтивного взаємодії з користувачем.

#### **Контекст як основа персоналізації**

Ефективна персоналізація залежить від глибокого контексту. Хоча більшість систем покладаються на дані про ідентичність та вподобання, вони часто недостатньо використовують ефемерні сигнали – той вид миттєвих вхідних даних, які забезпечують ситуативну релевантність. Співвідносячи довгострокові профілі з поточними спостереженнями, сучасні архітектури забезпечують точний досвід.

Розглянемо різницю між пропозицією знижки всім користувачам та орієнтацією на користувачів у холоднішому кліматі за допомогою акцій на зимове спорядження. Пристрої можуть бути однаковими, але середовища хостингу відрізняються. Контекстно-залежна система розпізнає це та діє. Цей прямий зв'язок між екологічною обізнаністю та адаптацією досвіду переводить персоналізацію з реактивної на проактивну.

На практиці, персоналізація, що відповідає контексту, змінює правила гри для всього: від цифрових вітрин та медіаплатформ до інтерфейсів телемедицини та корпоративних панелей керування. Йдеться вже не про надання контенту, а про забезпечення правильної взаємодії, через правильний канал, за правильних умов.

#### **Розкриття основних принципів архітектури доставки контексту**

##### **Контекстна усвідомленість: сприйняття та адаптація в режимі реального часу**

В основі доставки контексту лежить здатність систем активно сприймати дані про навколишнє середовище та реагувати відповідно. Цей принцип перетворює статичні програми на динамічні, адаптивні до потреб користувачів. Використовуючи сигнали від пристроїв, поведінки користувача, місцезнаходження, часу доби та станів системи, контекстно-залежні архітектури ініціюють відповідні дії. Ці зміни відбуваються не через жорстке кодування, а через вивід під час виконання – системи навчаються на операційних даних та миттєво коригують свою поведінку.

Наприклад, навігаційний додаток, виявляючи раптове збільшення заторів, змінює маршрут користувача на основі поточних даних про дорожній рух, моделей водіння користувача та історичних тенденцій заторів. Це не просто реакція; це превентивна адаптація, що стала можливою завдяки інтегрованим даним датчиків та механізмам прогнозування.

##### **Модульність: багаторівневі компоненти з чіткою відповідальністю**

Системи доставки контексту працюють безперебійно, коли побудовані на модульній архітектурі. Кожен рівень – прийом даних, логічний висновок та доставка – функціонує незалежно, але взаємодіє через чітко визначені інтерфейси. Така модульність сприяє

швидшій ітерації, ефективнішому усуненню несправностей та безперешкодному оновленню окремих компонентів.

- Рівень захоплення: збирає дані з датчиків, взаємодії з користувачами та зовнішніх систем, таких як API або журнали бази даних.

- Рівень логічного висновку: обробляє необроблені дані, застосовує моделі машинного навчання або механізми правил для отримання корисних висновків.

- Рівень доставки: перетворює аналітичні дані на зміни інтерфейсу користувача, зміни поведінки системи або відповіді API у режимі реального часу.

Розв'язка також підтримує співпрацю між командами: фахівці з обробки даних покращують моделі логічного висновку, не торкаючись інтерфейсу користувача, тоді як фронтенд-інженери налаштовують механізми доставки зі стабільними вхідними даними.

### **Масштабованість та мінімізація затримки**

Контекстно-залежні системи взаємодіють з величезними потоками даних. Щоб надійно обробляти це, вони повинні масштабуватися горизонтально. Це означає, що архітектури повинні враховувати зростаючу кількість користувачів, пристроїв і точок даних без погіршення продуктивності.

Мінімізація затримки під час цього процесу масштабування не є обов'язковою – вона сприяє утриманню користувачів. Дослідження, проведене Google, показало, що мобільні користувачі залишають сторінки, які завантажуються довше 3 секунд. У контекстній доставці вікно допуску ще вужче. Для прогнозних систем, таких як системи рекомендацій, затримка, що перевищує 200 мілісекунд, може перервати потік користувачів і зменшити залученість.

Щоб відповідати цим вимогам, архітектури доставки контексту включають периферійні обчислення, сітки даних в пам'яті, асинхронні шаблони обміну повідомленнями та паралельні конвеєри обробки.

### **Безшовна інтеграція між пристроями та платформами**

Контекст не знаходиться в одному додатку чи пристрої. Він рухається разом з користувачем – від смартфонів до планшетів, від пристроїв Інтернету речей до носимих пристроїв. Архітектура повинна підтримувати безперервність між сеансами, місцями розташування та апаратними екосистемами. Це включає підтримку обізнаності про стан та уніфікованих моделей користувачів через розподілене управління ідентифікацією та рівні збереження контексту.

Уявіть собі роздрібного покупця, який починає переглядати товари на планшеті, додає товари до кошика через мобільний пристрій, а потім завершує покупку на комп'ютері. Щоб цей досвід був цілісним, архітектура серверної частини розпізнає контекст як потік першокласних даних, а не як розрізнені вхідні дані. Цей уніфікований дизайн дозволяє системам виявляти правильний контент, підтримувати релевантність і синхронізувати дії протягом усього шляху користувача.

### **Технології, що забезпечують архітектуру доставки контексту**

#### **Обробка даних у реальному часі**

Системи доставки контексту вимагають швидкої реакції. Обробка даних у режимі реального часу усуває затримки, постійно надаючи додаткам актуальну інформацію. Рішення, що залежать від навколишніх ситуаційних даних – поведінки користувача, місцезнаходження, стану пристрою – отримують точність і оперативність завдяки цій можливості.

Фреймворки для обробки потоків формують основу систем реального часу. Apache Kafka обробляє високопродуктивні конвеєри даних у розподілених системах, діючи як надійна основа для публікації та підписки на контекстні події. Apache Flink, з іншого боку, оптимізує аналітику з низькою затримкою для необмежених потоків, що робить її ідеальною для оцінки та оновлення контексту в режимі реального часу в міру надходження вхідних сигналів.

Механізми перетворення та збагачення переплітають необроблені сигнали із семантикою. Наприклад, перетворені телеметричні дані часових рядів можуть включати виведені типи активності – ходьба, водіння, сидіння – отримані з шаблонів акселерометра. Таке збагачення надає контекстним наборам даних їхню інтерпретованість, дозволяючи системам ефективніше реагувати на сигнальні тригери.

### **Периферійні обчислення для швидкої оцінки контексту**

Зміщення обчислень контексту ближче до джерела даних скорочує затримку та підвищує ситуаційну обізнаність. Саме тут вступають у гру периферійні обчислення. Обробка контекстних подій поблизу периферії зменшує час передачі даних до централізованих серверних частин, що дозволяє приймати рішення на міросекундному рівні. Такі програми, як автономна навігація, промислова робототехніка та персоналізовані дисплеї для роздрібно́ї торгівлі, залежать від цих надшвидких реакцій.

Периферійні пристрої – телефони, смарт-хаби, камери – стають не просто збирачами, а й інтерпретаторами контексту. Їхня вбудована логіка визначає шаблони локально. Наприклад, роздрібний дисплей може використовувати розпізнавання обличчя для адаптації рекламного контенту на основі вікової групи покупця або тривалості концентрації уваги, і все це обробляється на пристрої без залежності від хмари.

### **Інтернет речей та дані датчиків**

Датчики роблять фізичний світ контекстуально доступним. Вхідні дані з різних модальностей – руху, світла, звуку, температури чи біометрії – створюють навколишній інтелект. Пристрої збирають ці дані пасивно, дозволяючи цифровим системам «відчувати» своє оточення.

Носимі пристрої, такі як розумні годинники, безперервно подають фізіологічні сигнали, які можуть інформувати про рівень стресу або тип активності. Смартфони постійно відстежують місцезнаходження користувача та моделі взаємодії. Розумні будинки та підключені транспортні засоби – за допомогою вбудованих датчиків – генерують дані про навколишнє середовище, які безпосередньо надходять до контекстного механізму.

Разом ці розподілені джерела даних з реального світу утворюють динамічну карту поведінки користувачів та середовища. Цей фізичний контекст забезпечує різні варіанти використання – від моніторингу здоров'я до адаптивної міської інфраструктури.

### **Семантична інтеграція даних**

Інтеграція контексту означає стандартизацію того, як різні дані взаємодіють один з одним. Це вимагає семантичного рівня, який узгоджує різноманітні джерела. Онтології визначають поняття та зв'язки між ними – що становить «намір користувача» або значення «дім» проти «офіс» – тоді як метадані анують ці сигнали машинозчитуваним значенням.

Такі технології, як RDF (Resource Description Framework), структурують цей граф даних. SPARQL надає мову запитів для його проходження та вилучення з нього значення. У поєднанні з принципами пов'язаних даних системи можуть виявляти нові контекстні шляхи між доменами, навіть коли базові джерела є різноманітними та розподіленими.

### **Машинне навчання для контекстного виведення**

Необроблені дані рідко розповідають повну картину. Моделі логічного висновку перетворюють комбінації сигналів на прогнозне розуміння. Моделі з учителем навчаються на основі позначеної поведінки, щоб розпізнавати заздалегідь визначені контексти, такі як виявлення того, коли клієнт активно робить покупки, а коли просто переглядає веб-сторінки. Моделі без учителя групують та виявляють нові стани, наприклад, виявлення аномалій у робочих процесах користувачів.

Навчання з підкріпленням динамічно адаптує послуги. Спостерігаючи за зворотним зв'язком попередніх взаємодій, ці системи оптимізують результати, що залежать від контексту. Вони можуть вибирати, який діалог використовувати у віртуальному помічнику, або точно налаштовувати рейтинг рекомендацій на основі передбачуваних змін намірів.

Вилучення ознак перетворює необроблені вхідні дані, такі як аудіошаблони або кластери датчиків, на контекстні вбудовування. Ці конденсовані, високовимірні вектори

представляють ситуаційні нюанси у формі, придатній для сприйняття машиною. Контекстно-залежні системи, що використовують такі вбудовування, отримують здатність моделювати не лише те, що відбувається, але й чому, і що, ймовірно, буде далі.

### **Архітектура фундаменту: Будівельні блоки системи доставки контексту**

#### **Рівень збору даних**

Відправною точкою будь-якої контекстно-залежної архітектури є надійний та різноманітний надходження даних. Добре функціонуючий рівень збору даних інтегрує кілька джерел інформації для фіксації динамічного середовища, що оточує користувача, у режимі реального часу.

– Датчики: Захоплюють сигнали низького рівня – GPS для визначення місцезнаходження, акселерометри для руху, датчики навколишнього освітлення для визначення умов навколишнього середовища.

– Трекери подій: відстежують взаємодію користувачів на різних платформах і пристроях, збираючи детальні дані про поведінку.

– Налаштування користувача: Включають як явно зазначені опції, так і виведені з часом шаблони, що пропонують важливі дані для персоналізації.

– Зовнішні API: надсилає дані зі сторонніх сервісів, таких як метеорологічні служби, агрегатори трафіку, фінансові ринки тощо.

Разом ці канали створюють багатовимірний профіль контексту реального часу. Вони слугують вихідними даними для наступного операційного рівня – обробки та міркування.

#### **Рівень обробки контексту та міркування**

На цьому етапі необроблені дані перетворюються на практичну інформацію. Цей рівень інтерпретує непересічні метадані та перетворює їх на структурований контекст за допомогою передових обчислювальних інструментів.

– Контекстні механізми: центральні процесори, які узгоджують вхідні дані з моделями користувачів та контекстними фреймворками.

– Системи логічного висновку: інтерпретація на основі логіки керування. Наприклад, розпізнавання того, що користувач, який рухається, підключений до системи Bluetooth автомобіля, ймовірно, керує автомобілем.

– Графи знань: забезпечують структуровані семантичні зв'язки між контекстними сутностями, що дозволяє системам з більшою точністю міркувати про наміри користувача.

Співпраця між логічними механізмами та моделями навчання призводить до розуміння змін стану користувача та умов навколишнього середовища в режимі реального часу, що дозволяє проактивно реагувати на доставку.

#### **Адаптація та рівень надання послуг**

Після того, як контекст користувача інтерпретовано, система повинна відповідно адаптувати свої послуги. Саме тут аналітичні дані перетворюються на персоналізований досвід.

– Логіка на основі правил та прогнозування: визначає, як і коли коригувати послуги. Логіка на основі правил працює на детермінованих умовах, тоді як прогнозування використовує ймовірності, отримані з моделей машинного навчання.

– Логіка програми: Змінює поведінку функцій залежно від контексту, наприклад, вимикає сповіщення в соціальних програмах під час зустрічей, запланованих у календарі.

– Коригування інтерфейсу користувача: Включає зміни інтерфейсу в режимі реального часу, такі як збільшення розміру шрифту на сонці або спрощення інтерфейсу користувача під час керування автомобілем.

Цей рівень визначає, наскільки інтелектуально та витончено застосунок реагує на зміни контексту, забезпечуючи адаптивність без перебоїв.

#### **Цикл зворотного зв'язку**

Жодна система доставки контексту не працює у вакуумі, і жодна конфігурація не залишається оптимальною вічно. Увійдіть у цикл зворотного зв'язку – безперервний механізм удосконалення.

Дії користувача після адаптації сервісу відображаються в системі, допомагаючи перекалібрувати моделі, налаштувати правила та переглянути припущення. З часом цей цикл підвищує точність реагування, виявляє нові закономірності та відкриває можливості для глибшої персоналізації. Чим більше система слухає, тим розумнішою вона стає.

### **Проектування для користувача: адаптивні та інтелектуальні інтерфейси**

#### **Адаптивні інтерфейси користувача (AUI)**

Інтерфейси більше не функціонують у вакуумі. Адаптивні користувацькі інтерфейси (AUI) динамічно реагують на середовище, наміри, пристрій та історичні моделі взаємодії кожного користувача. Ця адаптивність виходить за рамки простих налаштувань чи вибору тем. AUI змінюють ієрархію макета, елементи навігації, зворотний зв'язок про взаємодію та навіть доступність функцій у відповідь на контекстні тригери.

Наприклад, додаток для продуктивності може надавати пріоритет інструментам для співпраці під час робочого часу на комп'ютері, але відображати швидкі нотатки та віджети календаря під час поїздки користувача на роботу через мобільний пристрій. Така поведінка не є жорстко запрограмованою – вона походить від контекстного механізму, який інтерпретує такі сигнали, як час, місцезнаходження, рух та тип пристрою.

#### **Елементи інтерфейсу користувача, які переналаштовуються в режимі реального часу**

Динамічні елементи інтерфейсу значною мірою спираються на контекстні сигнали. Вхідні дані в режимі реального часу, такі як рівень навколишнього шуму, пропускна здатність інтернету, виявлення руху або спосіб введення, активують правила конфігурації, які за потреби реструктуризують компоненти інтерфейсу. Картки інтерфейсу можуть згортатися або розгортатися, панелі інструментів можуть змінювати порядок, а переходи адаптуються за інтенсивністю або швидкістю анімації.

- Панель навігації може перемикатися з значків + підписів на значки лише у сценаріях низької пропускної здатності або обмеженого простору.
- Голосові підказки можуть з'являтися в середовищах, де використання hands-free передбачається.
- Підказки та мікровзаємодії можуть подовжити видимість, коли дані вказують на те, що користувач новачок у наборі функцій.

#### **Стратегії персоналізації**

Універсальний досвід створює тертя. Персоналізовані інтерфейси уникають цього, використовуючи моделі контекстних шарів, які відстежують сегменти користувачів, уподобання, мікроповедінку та історію подій. Послідовність контенту, модифікація тонів повідомлень та рекомендації функцій адаптуються до намірів та архетипів персонажів.

Розглянемо освітній додаток: студент першого курсу інженерного факультету отримує більше візуальних матеріалів та спрощену термінологію, тоді як студент старших курсів отримує глибшу аналітику та посилання на дослідження. Базова модель персоналізації використовує дерева рішень або класифікатори машинного навчання, навчені на когортах та мітках використання, для запуску варіацій інтерфейсу.

Послідовність контенту, тон повідомлень та рекомендації щодо функцій:

- Послідовність контенту: статті, модулі чи ресурси відображаються на основі попередньої споживчої поведінки та контекстуального часу, наприклад, з'являються навчальні матеріали ближче до екзаменаційних сесій.
- Тон повідомлення: Мова сповіщень пом'якшується або загострюється відповідно до моделювання стану користувача (наприклад, користувачі, що перебувають у стані стресу, отримують заспокійливі, інформативні оновлення, а не термінові пінги).
- Рекомендації щодо функцій: функції програми відображаються контекстуально, залежно від ймовірності релевантності, передбачуваних цілей або нещодавно випущених функцій, що відповідають історичним потребам.

### **Аналіз поведінки користувачів**

Успішне виконання контекстних інтерфейсів вимагає безперервної поведінкової телеметрії. Одних лише сирих даних недостатньо для оптимізації – системи доставки контексту аналізують їх на корисні шаблони за допомогою сегментації та відображення взаємодії.

Відстеження включає шляхи дотику, глибину прокручування, час виконання завдання, рівень перерв та вагання під час введення. Ці показники, коли вони агреговані та проаналізовані, впливають як на адаптацію інтерфейсу користувача, так і на пріоритезацію контенту.

Теплові карти, повтори сеансів та поведінкові сегменти:

- Теплові карти візуалізують місця концентрації уваги та взаємодії. Інтенсивні зони можуть свідчити про перенаселеність, тоді як ігноровані елементи сигналізують про неактуальність.

- Повтори сесій сприяють якісному аналізу, виявляючи вагання, помилки кліків або перешкоди під час адаптації, які не завжди очевидні в сукупних показниках.

- Поведінкові сегменти класифікують користувачів за такими параметрами, як орієнтація на ціль, цифрова грамотність або контекст сеансу, що дозволяє точніше налаштовувати інтерфейс.

Фіксуючи мікрвзаємодії в режимі реального часу та поєднуючи їх із шарами інтерпретації з високим контекстом, сучасні інтерфейси переходять від реактивних до справді випереджальних.

### **Сервісні шини та мікросервіси для модульної доставки**

Архітектури корпоративного рівня дотримуються модульності завдяки сервісним шинам та мікросервісам. Легка корпоративна сервісна шина (ESB) координує незалежні мікросервіси, призначені для аналізу рівнів контексту, таких як місцезнаходження, активність, ідентифікація та поведінка. Кожен мікросервіс розвивається незалежно, що дозволяє системі вибірково масштабуватися та підтримувати стримування відмов. Такі інструменти, як Istio та Linkerd, підтримують реалізацію сервісної мережі, забезпечуючи спостережуваність та стійкість в екосистемі доставки контексту.

### **Обробка рівня контекстного API Внутрішня та зовнішня інтеграція**

В основі лежить рівень контекстного API. Цей трансверсальний рівень приймає, обробляє та надає контекст як внутрішнім модулям, так і зовнішнім партнерам. Він використовує поєднання кінцевих точок RESTful та GraphQL, що дозволяє точно фільтрувати, змінювати та запитувати контекст у режимі реального часу. Зовнішні системи – будь то CRM, платформи Інтернету речей чи мобільні додатки – споживають контекст через токеновані канали доступу з керованими дозволами, забезпечуючи гнучкість та безпеку інтеграції.

### **Метрики якості досвіду (QoE)**

Надання контексту не обмежується технічною точністю – воно досягає кульмінації у сприйнятті користувацького досвіду. Три показники якості обслуговування (QoE) оцінюють продуктивність:

- Затримка: Мілісекунди визначають релевантність. Системи, оптимізовані для периферійних обчислень та попередньої вибірки на основі CDN, досягають часу доставки менше 50 мс для критичного контексту, такого як обробка переривань в автомобільних системах.

- Точність персоналізації: Вимірювана за показниками релевантності протягом сеансів користувачів, точність вище 85% у рекомендаціях або адаптації інтерфейсу користувача свідчить про надійне моделювання контексту.

- Рівень залученості користувачів: Очевидний вплив надання контексту надалі. Платформи зі зрілими контекстними фреймворками повідомляють про в 3–6 разів більшу тривалість сеансу та до 4-кратне збільшення щільності взаємодії порівняно із системами статичного контенту.

### **Збільшення бізнес-цінності за допомогою архітектури доставки контексту**

#### **Посилення залученості завдяки точному визначенню часу**

Загальне спілкування не привертає уваги. Архітектура доставки контексту усуває цю проблему, синхронізуючи контент з намірами, поведінкою та середовищем користувача. Аналізуючи сигнали в режимі реального часу, такі як місцезнаходження, тип пристрою, історичні взаємодії та навіть умови навколишнього середовища, система вибирає найрелевантніше повідомлення щомиті.

Цей підхід є основою для персоналізації кампаній у режимі реального часу. Користувач, який отримує push-сповіщення під час активного перегляду пов'язаного контенту, реагує набагато швидше, ніж той, кого переривають під час простою. Компанії, які використовують цю точність часу, повідомляють про покращення показників залученості. Наприклад, Gartner зазначив, що високоцільовий контекстний маркетинг може підвищити рівень відповідей до 20% порівняно зі статичними кампаніями.

#### **Оптимізація рентабельності інвестицій за допомогою інтелектуального розподілу ресурсів**

Контекстно-залежні системи доставки автоматизують значну частину процесу прийняття рішень, який раніше вимагав ручного налаштування та контролю. Замість впровадження універсальних стратегій, компанії можуть налаштовувати інтелектуальні потоки, які автоматично коригують контент, час та канал доставки на основі даних у реальному часі.

Це зменшує зайві зусилля, знижує вартість взаємодії з кожним клієнтом та покращує використання ресурсів. Наприклад:

- Маркетингові команди виграють від меншої кількості ітерацій контенту, водночас досягаючи вищих конверсій.
- Відділи продажів отримують ліди, сегментовані за намірами, що зменшує зусилля на холодний контакт.
- Інженерні команди уникають жорстко запрограмованої логіки персоналізації, зосереджуючись натомість на основних завданнях розробки.

За даними McKinsey, компанії, які застосовують передові стратегії персоналізації, що підтримуються контекстними технологіями, спостерігають зростання доходів на 5–15% та підвищення ефективності маркетингових витрат на 10–30%.

#### **Збільшення утримання клієнтів завдяки гіперперсоналізованому досвіду**

Коли інтерфейси відповідають контексту користувача – передбачаючи потреби, адаптуючи потік контенту та відображаючи поведінку користувача – залученість застосунку зростає. Клієнти залишаються довше, частіше взаємодіють та демонструють вищий рівень задоволеності.

Персоналізація на основі контекстної доставки також зменшує відтік користувачів. Стрімінгові платформи, що адаптують рекомендації до певних настроїв та умов перегляду, або фітнес-додатки, що змінюють пропозиції залежно від часу доби та нещодавньої активності, змушують користувачів повертатися.

На практиці це безпосередньо впливає на кінцевий результат. Adobe повідомляє, що компанії, які використовують розширену персоналізацію на основі контексту, спостерігають збільшення цінності життя клієнта (CLV) у 1,7 раза.

#### **Варіанти використання, розкриті завдяки наданню контексту**

Контекстна архітектура не обмежується теорією. Вона проявляється у відчутних бізнес-функціях у різних відділах:

- Маркетингова автоматизація: тригерні електронні листи та повідомлення реагують на поведінку користувачів у режимі реального часу – залишення кошика, паузи в перегляді або сплески інтересу. Контент розвивається з кожною взаємодією, щоб відображати, на якому етапі своєї подорожі знаходиться користувач.
- Оптимізація підтримки клієнтів: Чат-боти підтримки, наповнені контекстними сигналами, можуть випереджати запитання, посилаючись на нещодавні дії, характеристики

пристроїв або моделі використання послуг. Це скорочує час вирішення проблем і підвищує показники задоволеності клієнтів.

Реальні впровадження демонструють стабільні результати. Компанії, які використовують контекст для керування взаємодією зі службою підтримки, скоротили середній час обробки на 30–40% та покращили показники вирішення проблем з першого контакту більш ніж на 20%.

Балансування точності та захисту: безпека, конфіденційність та управління етичним контекстом

### **Конфіденційність за проектом у контекстній наданні**

Кожен рівень архітектури доставки контексту – збір даних, обробка, виведення та доставка – повинен вбудовувати засоби контролю конфіденційності з самого початку. Конфіденційність за принципом проектування (PbD) виходить за рамки дотримання вимог; вона інтегрує механізми конфіденційності в операційну логіку систем. Це означає, що конфіденційні вхідні дані ніколи не торкаються незахищеного сховища, контекстні висновки залишаються в межах визначених меж, а всі процеси за замовчуванням дотримуються принципу мінімального розкриття.

Підхід PbD вимагає проактивної, а не реактивної інженерії. Наприклад, моделі персоналізації на основі контексту повинні анонімізувати точки даних під час їх отримання, а не після нього. Вони також повинні створювати системні бар'єри, що запобігають надмірному контекстуалізму, коли системи роблять висновки, що виходять за рамки очікувань або корисності користувача.

### **Згода користувачів, мінімізація даних та прозорість політики**

В архітектурі контекстної доставки дані не просто чекають на запит – вони динамічно передаються, узгоджуючи їх із шаблонами використання, просторовими умовами та поведінкою користувачів у режимі реального часу. Цей потік має ґрунтуватися на чіткій та відкличній згоді користувача.

– Явні дозволи замінюють приховані прапорці; користувачі повинні розуміти, що вони дозволяють.

– Мінімізація даних означає збір лише інформації, що відповідає конкретному випадку використання. Якщо сервіс не потребує детальної геолокації, він не повинен мати до неї доступу.

– Чітко опубліковані політики, структуровані доступною мовою та підкріплені машинозчитуваними форматами, гарантують, що як люди, так і агенти штучного інтелекту дотримуються умов договорів використання.

Ця модель створює цифровий контракт, який є дієвим, відстежуваним та таким, що може бути виконаний, що є ключовим для масштабування контекстно-залежних послуг у різних юрисдикціях та екосистемах.

### **Забезпечення контексту на кожній точці взаємодії**

Контекстні дані не є статичними. Вони швидко передаються між рівнями – датчиками, посередниками, сервісами, механізмами прийняття рішень. Це робить їх захист на кожному етапі невід'ємним.

Шифрування на периферії має застосовуватися якомога ближче до джерела. Дані датчиків та сигнали користувача повинні бути зашифровані перед тим, як залишити пристрій, за допомогою стандартизованих бібліотек та захищених ключів. Це запобігає атакам типу «людина посередині» та не дозволяє платформам перехоплювати необроблений контекст.

Далі, моделі безпечного федеративного навчання дозволяють системам покращувати свої можливості контекстного виведення без агрегації персональних даних у централізовані бази даних. Замість надсилання даних до моделі, модель безпечно передається до даних, навчається локально та повертає лише зашифровані статистичні оновлення.

### **Автентифікація та довіра в екосистемах багатьох пристроїв**

Один користувач може взаємодіяти з десятком пристроїв протягом 24 годин – телефонами, годинниками, терміналами, розумними колонками, транспортними засобами. Ці кінцеві точки повинні координувати свою роботу, не порушуючи меж довіри.

– Безперервна автентифікація особи за допомогою мультимодальної біометрії гарантує, що контекст відповідає фактичному користувачеві, а не лише пристрою.

– Оцінка довіри між пристроями визначає, чи може вторинний пристрій отримувати доступ до контексту з первинного вузла, запобігаючи перевантаженню з менш захищених кінцевих точок.

– Принципи архітектури нульової довіри вимагають, щоб кожен запит або обмін даними був автентифікований, перевірений та зареєстрований, незалежно від зв'язку джерела з мережею.

Разом ці механізми створюють стійку структуру довіри, гарантуючи, що доставка контексту залишається узгодженою з ідентифікатором користувача, правами доступу та рівнем безпеки, незалежно від того, наскільки гнучкою стає екосистема.

### **Об'єднання інтелекту, досвіду та інфраструктури**

Архітектура доставки контексту (CoDA) – це не теоретична основа, що збирає цифровий пил, а операційна парадигма, яка диктує, як поведуться чутливі, релевантні та прогнозні цифрові системи. Кожен компонент, від датчиків, розгорнутих на периферії, до хмарних аналітичних механізмів, відповідає одній директиві: надавати точну, персоналізовану інформацію потрібному користувачеві за правильних умов.

Цей фреймворк процвітає завдяки поєднанню технологій, які вже вплетені в цифрову тканину – машинного навчання, обробки даних у реальному часі, CDN, адаптивних фреймворків інтерфейсу користувача та платформ оркестрації. Зі зближенням цих рівнів вони генерують більше, ніж просто підвищення продуктивності. Вони відкривають можливості, орієнтовані на аудиторію, де програми не чекають дій користувачів; вони передбачають та реагують динамічно, формуючись часом, місцем розташування, поведінкою та уподобаннями.

Подивіться, як адаптивний UX-дизайн сьогодні не спирається виключно на естетику інтерфейсу – він зчитує сигнали. Маршрутизація контекстних сигналів, таргетування збагаченого контенту та ідіоматичне моделювання поведінки стали міркуваннями дизайну, а не лише механізмами бекенду. Цей зсув відображає суть CoDA: об'єднання архітектури, інтерфейсу та обчислень у безшовний, інтелектуальний досвід.

#### **Розгортання плану**

– Виберіть одну сферу – охорона здоров'я, роздрібна торгівля, фінанси. Зробіть карту, як контекст може покращити шляхи доставки або переосмислити точки дотику.

– Проведіть опитування зацікавлених сторін щодо прогалин у точках контакту та неефективності. Де сьогодні ігнорується контекст?

– Визначте, де поточні конвеєри (від оркестрації API до пристрою користувача) розривають ланцюг між станом користувача та поведінкою програми.

– Прототипні сценарії, де система реагує на виведений контекст, а не на статичні вхідні дані чи підказки користувача.

Ви вже бачили функцію, яка змінювалася залежно від вашого місцезнаходження або нещодавнього шаблону використання? Це була взаємодія дизайну, архітектури та даних. Розбийте це – які системи це забезпечили? Кому належать ці контекстні моделі?

Вивчення реальних архітектур – не лише кодових баз, а й потоків доступу, карт затримки та ієрархій сигналів – пропонує більше розуміння, ніж статичні офіційні документи. CoDA переживає шаблони розгортання та спостережувані зміни у поведінці використання. Затримайтеся в цьому просторі, і системи, готові до майбутнього, не залишаться абстрактними ідеями, а операційними активами, готовими до впливу у великих масштабах.

### **Безпека продукту**

CoDA пропонує функції безпеки продуктів корпоративного рівня для більшого контролю, прозорості та гнучкості:

– Автентифікація: CoDA пропонує широкий спектр методів автентифікації, таких як SSO з SAML 2.0, вхід за допомогою Google, Microsoft, Apple, Magic Links та базова автентифікація з 2-факторною автентифікацією. Корпоративні клієнти можуть встановлювати політики автентифікації та керувати користувачами за допомогою SCIM.

– Шифрування: Дані клієнтів шифруються під час передачі та зберігання за замовчуванням.

– Авторизація: керування доступом до документів, папок, пакетів та робочих просторів. Спільний доступ із групами Google. Корпоративні клієнти також можуть ділитися файлами з групами SCIM та встановлювати розширені правила спільного доступу для форм, документів та пакетів.

– Аудит: Ми надаємо API аудиту для корпоративних клієнтів, щоб отримувати журнали аудиту за останні 12 місяців. CoDA також надає пакет адміністратора для перегляду подій аудиту.

– Політики підприємства: підприємства можуть встановлювати політики для керування користувачами, документами та пакетами. До них належать політики для автентифікації користувачів, зовнішнього та вхідного спільного доступу, публікації, створення спільних папок, експорту даних, завантаження файлів та тривалості сеансу.

– Панелі керування Enterprise: CoDA пропонує корпоративним клієнтам розширені адміністративні панелі керування для легкого керування їхніми робочими процесами в CoDA. До них належать панелі керування для перегляду та керування ліцензіями, загальнодоступними документами, активністю користувачів та документами, що належать користувачам з деініціалізацією.

– Конфігурації та схвалення пакетів: CoDA пропонує високорозвинені засоби контролю безпеки для інтеграцій сторонніх розробників. Підприємства мають повний контроль над тим, які дані можна вносити до CoDA, хто може їх вносити та хто може до них отримувати доступ.

### **Безпека застосунків**

Зобов'язання CoDA щодо безпеки починаються з процесів, інструментів та практик для безперервного проектування та розробки безпечного програмного забезпечення:

– Життєвий цикл розробки безпеки: Наша програма життєвого циклу розробки безпеки інтегрована в кожен етап процесу розробки програмного забезпечення для безперервного створення безпечного програмного забезпечення. Прикладами є щорічні тренінги з безпеки для всіх співробітників, моделювання загроз як частина процесу проектування та інструменти статичного аналізу коду.

– Щорічне тестування на проникнення: CoDA проводить щорічне тестування на проникнення, яке проводять авторитетні дослідницькі фірми з безпеки. Сфера цього тестування включає: веб-застосунки, інфраструктуру пакетів CoDA, хмарну інфраструктуру та мобільні застосунки.

– Програма публічної винагороди за виявлені помилки: CoDA запускає публічну програму винагороди за виявлені помилки через HackerOne.

### **Безпека інфраструктури**

CoDA створена з нуля з використанням найкращих практик безпеки AWS:

– Хмарна інфраструктура: CoDA побудована з використанням усталених принципів безпеки, включаючи глибоко ешелонований захист, мінімальні привілеї та зменшення площі поверхні атаки. CoDA дотримується найкращих практик AWS щодо мережевої безпеки, використовуючи такі сервіси, як AWS CloudFront, AWS WAF, групи безпеки AWS та VPC.

– Безпека операцій: Ми використовуємо багатофакторну автентифікацію, контроль доступу на основі ролей (RBAC) та гранти доступу «точно вчасно» для безпечного

керування нашим сервісом. Крім того, ми реєструємо події аудиту та інформацію про безпеку на кожному рівні нашої інфраструктури та відстежуємо їх на наявність підозрілої активності.



Рисунок 1 – Структурна схема системи

– Резервні копії: CoDA щодня створює резервні копії баз даних для всіх даних, що зберігаються в AWS. Резервні копії зазвичай зберігаються щонайменше 35 днів. Ми також проводимо регулярні перевірки цілісності резервних копій та проводимо навчання з аварійного відновлення на робочому столі щонайменше раз на рік.

– Синхронізація годинника: інформаційні ресурси CoDA використовують протокол NTP для синхронізації із затвердженими джерелами NTP.

– Безпека пакетів: пакети виконуються в ізольованих захищених пісочницях. Розробник пакетів ніколи не торкається облікових даних авторизації, які використовуються користувачами в пакетах. Ми гарантуємо, що пакети обмінюються даними лише з тими веб-сайтами, про які вони заявляють.

– Шифрування: CoDA використовує службу керування ключами Amazon (KMS) для створення, обслуговування та ротації ключів шифрування. CoDA також використовує протокол Transport Layer Security (TLS 1.2) для шифрування даних користувачів під час передачі та симетричний алгоритм шифрування AES-256 для шифрування даних клієнтів у стані спокою.

### Склад Комплексу

Комплекс складається з наступних компонентів:

1. Набір шлюзів кодування.
2. Центр генерації ключів.
3. Центр розподілу ключів.
4. Центр реєстрації мобільних клієнтів.

5. Центр підготовки електронних ключів мобільних клієнтів.
6. Мобільний клієнт.
7. Центр моніторингу.
8. Програма контролю цілісності.

**Висновки.** У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів визначення контексту користувача на основі архітектури CoDA. Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем визначення контексту користувача на основі архітектури CoDA.
- Досліджена система визначення контексту користувача на основі архітектури CoDA.
- На основі отриманих результатів досліджень створена програмна реалізація системи визначення контексту користувача на основі архітектури CoDA.

Розроблені алгоритми дозволяють успішно вирішувати завдання визначення контексту користувача на основі архітектури CoDA. 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. Kuznetsov, O., Smirnov, O., Akhmetov, B., Alimseitova, Z., Imoize, A.L. «Deep Learning Frontiers in Copy-Move Forgery Detection: Advances, Challenges, and Future Directions». *Advancements in Cybersecurity Next Generation Systems and Applications*, 2025. 202-229.
2. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.). Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.
3. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Driciev, O., Smirnov, O., Dorenskiy, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.
4. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуй А.М. «Модель шляхів отримання вхідних даних комп'ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». *Центральноукраїнський науковий вісник. Технічні науки*. 2025. Вип. 11(42), ч. II. С.52-62.
5. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В. «Методи забезпечення відмовостійкості інтелектуальних систем підтримки оператора». VIII міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 24-25 квітня 2025 р. – Кропивницький: ЦНТУ. – 2025. – С. 44-46.
6. Смірнов, О.А., Константинова, Л.В., Коноплицька-Слободенюк, О.К., Козірова, Н.В., Якименко, Н.М., Доренський, О.П., Буравченко, К.О. «Дослідження інструментів штучного інтелекту для роботи з базами даних та аналізу даних». *Кібербезпека: освіта, наука, техніка*. 2025. №3(27), С. 429–448.
7. Lakhno, V., Malyukov, V., Smirnov, O., Bebesko, 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.
8. Smirnov O., Fedorov E., Neskrodieva A., Neskrodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». *CEUR Workshop Proceedings Volume 3664*, 2024, Pages 11-23.
9. 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.
10. Malyukov V., Bebesko 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.
11. 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.
12. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchov, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings, Volume 3530*, 2023, pp. 256-265.

13. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.
14. 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.
15. Smirnov, O., Neskorođieva, T., Fedorov, E., Rudakov, K., Neskorođieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12.
16. 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.
17. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskiy, V., Khorolska, K., Bebishko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, K.L., Vuppapapati, 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.
18. 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.
19. 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>.
20. Smirnov O., Neskorođieva 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.
21. 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.
22. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.
23. 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.
24. 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.
25. 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.
26. 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.
27. Smirnov O. Kuznetsov A., Zaichenko 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.
28. 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.
29. 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.
30. 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.
31. 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.
32. 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.
33. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.
34. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.