Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Створення динамічного цифрового двійника клієнта для симуляції бізнес-процесів
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-23-1
Олександр Греков
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник доц. Марія Головянко
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Грекову Олександру Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Створення динамічного цифрового двійника клієнта для симуляції бізнес процесів

затверджена наказом університету від 21 квітня 20 25 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 3 червня 20 25 р.

3. Вихідні дані до роботи документація до мови програмування Python, документація до фреймворку PyTorch, науково-технічні публікації, набір даних для тренування та тестування системи, дані Інтернет-джерел

4. Перелік питань, що потрібно опрацювати в роботі

1) Теоретичне підґрунтя та пов'язані роботи

2) Постановка задачі та аналіз набору даних

3) Методологія та дизайн системи

4) Деталі реалізації

5) Експериментальні дослідження та експерименти

6) Економічна доцільність та оцінка ризиків

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	21.04.2025	виконано
2	Аналіз предметної галузі	26.04.2025	виконано
3	Аналіз методології та підходу DToC	29.04.2025	виконано
4	Вибір методу та вирішення задачі	02.05.2025	виконано
5	Експериментальне моделювання та навчання	05.05.2025	виконано
6	Програмна реалізація системи	07.05.2025	виконано
7	Написання пояснювальної записки	12.05.2025	виконано
8	Перевірка на академічний плагіат	19.05.2025	виконано
9	Нормоконтроль	22.05.2025	виконано
10	Підготовка презентації та доповіді	25.05.2025	виконано
11	Попередній захист	28.05.2025	виконано
12	Рецензування	30.05.2025	виконано
13	Захист перед ЕК	03.06.2025	

Дата видачі завдання 21 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Марія Головянко
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 69 с., 11 рис., 1 табл., 1 дод., 22 джерела.

БІЗНЕС-СИМУЛЯЦІЯ, ГЛИБИННЕ НАВЧАННЯ, ДИНАМІЧНЕ МОДЕЛЮВАННЯ, МОДЕЛІ ПОСЛІДОВНОСТЕЙ, РІТЕЙЛ, ЦИФРОВИЙ ДВІЙНИК КЛІЄНТА, ШТУЧНИЙ ІНТЕЛЕКТ.

Об’єкт дослідження – процеси взаємодії підприємства з клієнтами в цифровому та фізичному середовищах.

Предмет дослідження – методи побудови та он-лайн актуалізації динамічного цифрового двійника клієнта (DTOC), що моделює поведінку на основі потоків подій.

Мета роботи – розробити науково-обґрунтовану методику створення й використання DTOC для прогнозування реакції на управлінські впливи та підвищення ефективності бізнес-процесів.

Методи дослідження – математичне моделювання, глибинне навчання (рекурентні та трансформерні мережі), подієво-орієнтовані симуляції, статистичний аналіз; програмна реалізація у Python / PyTorch.

Сформульовано формальну модель DTOC як композицію статичних і динамічних компонентів; запропоновано нейромережеву архітектуру, що інкрементно оновлює стан двійника; уперше поєднано персоніфіковане подієве моделювання з он-лайн оновленням hidden-state. Експерименти на відкритих транзакційних даних показали приріст точності прогнозу наступної покупки на 30% порівняно з базовими підходами, а бізнес-симуляція дисконтних кампаній спричинила потенційне зростання виручки до 50% без погіршення утримання клієнтів. DTOC може використовуватися в системам підтримки прийняття рішень, персоналізований маркетинг, прогнозування поведінки споживачів.

ABSTRACT

Master’s thesis contains: 69 pp., 11 fig., 1 tabl., 1 ann., 22 references.

ARTIFICIAL INTELLIGENCE, BUSINESS SIMULATION, DEEP LEARNING, DIGITAL TWIN OF A CUSTOMER, DYNAMIC MODELING, RETAIL, SEQUENCE MODELS.

The object of research is the processes of enterprise interaction with customers in digital and physical environments.

The subject of the study is methods of building and online updating of a dynamic digital twin of a customer (DTOC) that models behavior based on event flows.

The purpose of the study is to develop a scientifically based methodology for creating and using DTOC to predict the response to managerial influences and improve the efficiency of business processes.

Research methods: mathematical modeling, deep learning (recurrent and transformer networks), event-driven simulations, statistical analysis; software implementation in Python/PyTorch.

The formal model of DTOC is formulated as a composition of static and dynamic components; a neural network architecture that incrementally updates the state of the twin is proposed; for the first time, personalized event-based modeling is combined with online hidden-state updating. Experiments on open transactional data have shown an increase in the accuracy of the next purchase prediction by 30% compared to baseline approaches, and business simulation of discount campaigns resulted in potential revenue growth of up to 50% without deteriorating customer retention. DTOC can be used in decision support systems, personalized marketing, and consumer behavior forecasting.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence – штучний інтелект;

BPMN – Business Process Model and Notation – модель та нотація бізнес-процесів;

CRM – Client Relationship Manager – менеджер по роботі з клієнтами;

DT – Digital Twin – цифровий двійник;

DTOC – Digital Twin of a Customer – цифровий двійник клієнта;

EPC – Event-Driven Process Chain – подієво-орієнтований ланцюжок процесів;

GDPR – General Data Protection Regulation – Загальний регламент захисту даних;

KMS – Knowledge Management System – система управління знаннями;

LSTM – Long Short-Term Memory – довга короткочасна пам’ять (тип нейронної мережі);

NASA – National Aeronautics and Space Administration – Національне управління з аеронавтики і дослідження космічного простору США;

PT – Digital Twin of Product – цифровий двійник продукту;

RNN – Recurrent Neural Network – рекурентна нейронна мережа;

SKU – Stock Keeping Unit – одиниця збереження товару (ідентифікатор).

ВСТУП

Цифрова трансформація стала ключовим чинником сучасного розвитку бізнесу, змінюючи підходи до взаємодії з клієнтами та організації бізнес-процесів. Однією з інноваційних технологій, що набула широкого поширення, є концепція цифрового двійника – віртуальної репліки фізичного об’єкта або системи, здатної моделювати реальні процеси та сценарії. Спочатку цифрові двійники використовувалися переважно в інженерії та виробництві, однак з часом їх застосування поширилося й на інші галузі, зокрема на управління взаємовідносинами з клієнтами (CRM) та оптимізацію бізнес-процесів.

Незважаючи на стрімкий розвиток, більшість існуючих реалізацій цифрових двійників орієнтовані на статичне представлення об’єкта, що обмежує їх ефективність у динамічних середовищах, де зміни відбуваються швидко та непередбачувано. Сучасні підприємства потребують рішень, здатних адаптуватися в реальному часі до змін у поведінці споживачів, ринкових умовах та нових викликах. Саме тому виникла необхідність у створенні динамічного цифрового двійника клієнта (Dynamic Twin of Customer, DTOC), який здатен відображати не лише статичні характеристики, а й постійно оновлювати модель поведінки клієнта на основі нових подій та транзакцій [1].

Актуальність цієї роботи обумовлена зростаючим попитом на персоналізований підхід до обслуговування клієнтів. У міру загострення конкуренції на ринку компанії все частіше використовують інструменти прогнозної аналітики та індивідуальної взаємодії з клієнтами. Традиційні методи аналізу даних, які базуються лише на історичних даних, часто не здатні відслідковувати зміни в реальному часі та враховувати контекст поточних подій. У зв’язку з цим розробка динамічного цифрового двійника клієнта, що реагує на події в режимі реального часу, є важливим кроком як у науковому, так і в прикладному аспектах.

Метою даної роботи є розробка та реалізація моделі динамічного цифрового двійника клієнта для симуляції бізнес-процесів на прикладі взаємодії користувача з електронною комерцією. Основне завдання полягає в демонстрації того, як за допомогою DTOC можна здійснювати точне моделювання клієнтської поведінки, прогнозування подальших дій та застосовувати модель для прийняття обґрунтованих бізнес-рішень. Додатково, в рамках роботи розглянуто приклад використання DTOC для симуляції ефективності маркетингових акцій та індивідуальних пропозицій.

Модель DTOC має широкий спектр можливого застосування: в роздрібній торгівлі, банківській справі, телекомунікаційній сфері та будь-яких інших галузях, де ключовим елементом є взаємодія з клієнтом. Наприклад, у сфері електронної комерції така модель може слугувати для прогнозування майбутніх покупок, оптимізації запасів продукції та персоналізації маркетингових повідомлень. У фінансовому секторі – для передбачення поведінки клієнтів і підбору релевантних продуктів або для оцінки ризиків [2].

Історично ідея цифрових двійників бере початок ще з програм космічних досліджень NASA (National Aeronautics and Space Administration), де створювалися віртуальні копії апаратів для дистанційного контролю та діагностики. Масового поширення термін «цифровий двійник» набув у 2000-х роках в індустріальному середовищі, зокрема в контексті керування життєвим циклом продукції. Сьогодні, завдяки розвитку машинного навчання, комп’ютерного моделювання та обробки великих даних, технології та ідеї цифрових двійників дедалі частіше використовуються і в комерційних та споживчих сферах, і має великі можливості в майбутньому.

Крім того, важливим аспектом розвитку технології динамічного цифрового двійника клієнта є забезпечення етичності та конфіденційності обробки персональних даних.

1 ТЕОРЕТИЧНЕ ПІДҐРУНТЯ ТА ПОВ’ЯЗАНІ РОБОТИ

Цифровий двійник (Digital Twin, DT) – це віртуальна репрезентація фізичної сутності або процесу, яка постійно синхронізується зі своїм прототипом через потоки даних і дає змогу проводити аналіз «що-буде-якщо», прогнозувати стани й оптимізувати керування системою. За підрахунками аналітиків, світовий ринок DT оцінюється у ≈ 10 млрд $, а до 2028 р. може перевищити 110 млрд $. Станом на зараз, поняття DT виходить далеко за межі інженерії й охоплює медицину, енергетику та маркетинг, де формується окрема підгалузь – Digital Twin of a Customer (DTOC) [3].

1.1 Концепція цифрових двійників

Цифровий двійник (Digital Twin, DT) є концепцією, що виникла у промисловому середовищі та спочатку використовувалася для моделювання фізичних об'єктів, таких як машини, будівлі та цілі виробничі лінії. Цей підхід дозволяє створювати віртуальну копію фізичного об'єкта, яка постійно синхронізується з реальними даними та дозволяє проводити моніторинг, діагностику та оптимізацію його роботи. У класичному розумінні цифровий двійник – це точна математична модель, яка відображає стан та поведінку фізичної системи на основі даних, що надходять у режимі реального часу.

Концепція цифрових двійників поступово вийшла за межі індустрії та знайшла застосування в інших сферах, включаючи медицину, транспорт, міську інфраструктуру та, що найважливіше у контексті цієї роботи, обслуговування клієнтів. Саме тут виникла ідея цифрового двійника (Digital Twin of the Customer, DTOC), який замість фізичного об'єкта моделює поведінку, уподобання та реакції користувачів на основі їхніх транзакційних і поведінкових даних [4].

DTOC розширює класичне розуміння цифрового двійника за рахунок інтеграції різнорідних даних, що відображають не лише фізичну активність клієнта (наприклад, покупки), але й його контекстуальні взаємодії, такі як перегляд продуктів, кліки на рекламні банери, використання додатку або сайту. На відміну від статичних профілів або персон, які фіксують лише демографічні характеристики користувача, DTOC постійно оновлюється з появою нових подій, що дозволяє будувати його динамічний портрет.

Суттєвою перевагою DTOC є можливість не лише аналізувати минулу поведінку клієнта, а й прогнозувати його майбутні дії, використовуючи моделі глибокого навчання. Це забезпечує персоналізовані рекомендації, таргетовані пропозиції, виявлення потенційних відтоків клієнтів та навіть симуляцію альтернативних сценаріїв. Наприклад, модель може передбачити, який товар найімовірніше придбає клієнт після отримання знижки або що станеться, якщо йому запропонувати альтернативний продукт.

Важливим аспектом DTOC є його адаптивність: система не лише зберігає історичні дані, але й здатна миттєво реагувати на нові події, автоматично оновлюючи цифровий двійник. Це відбувається завдяки використанню методів обробки послідовностей (наприклад, LSTM або Transformer), які здатні враховувати часові залежності та контекстуальні фактори.

DTOC також забезпечує можливість бізнес-симуляцій, що дозволяє прогнозувати, як клієнти відреагують на ті чи інші маркетингові (знижки, рекламні кампанії, персоналізовані пропозиції). Це робить систему не лише інструментом аналізу, а й механізмом підтримки прийняття рішень. Таким чином, DTOC можна розглядати як потужний засіб автоматизації маркетингової стратегії та підвищення прибутковості бізнесу [5].

У сучасних умовах DTOC стає важливим компонентом цифрової трансформації компаній, що прагнуть забезпечити персоналізований підхід до кожного клієнта. На відміну від традиційних систем аналізу, що

працюють із агрегованими даними, DTOC дозволяє відстежувати та прогнозувати поведінку на рівні окремого користувача. Це створює значну конкурентну перевагу, адже персоналізований підхід значно підвищує лояльність клієнтів та їхню готовність до повторних покупок.

Таким чином, DTOC не лише моделює поточний стан клієнта, а й формує інтерактивний канал взаємодії, де кожна дія користувача впливає на подальшу стратегію обслуговування. Саме ця здатність до адаптації і робить цифровий двійник клієнта надзвичайно перспективним інструментом як у маркетингу, так і в інших сферах, що потребують персоналізованого підходу.

1.1.1 Цифровий двійник продукту в порівнянні з клієнтом

Цифровий двійник продукту (Digital Twin of a Product, DTP) та цифровий двійник клієнта (Digital Twin of the Customer, DTOC) є двома різними, але взаємодоповнюючими концепціями, що виникли у межах цифрової трансформації бізнесу. Хоча обидві моделі побудовані на основі ідеї створення віртуального аналогу реального об'єкта, вони мають різну спрямованість та використовуються для вирішення різних завдань.

DTP зосереджується на відображенні характеристик і стану фізичного продукту або системи. Це може бути як окремий товар (наприклад, автомобіль, який має цифровий двійник із відображенням технічного стану, пробігу, історії обслуговування), так і складна система, така як виробнича лінія або літак. Основною метою DTP є моніторинг стану продукту в реальному часі, діагностика можливих проблем, прогнозування відмов та оптимізація обслуговування. Наприклад, цифровий двійник двигуна літака може повідомити про потребу в технічному обслуговуванні на основі аналізу вібрацій та температурних показників.

На відміну від цього, DTOC зосереджений не на фізичному об'єкті, а на його користувачеві. DTOC відображає поведінку клієнта, його

уподобання, реакції на маркетингові пропозиції та історію взаємодій із системою. Це дозволяє бізнесу прогнозувати майбутні дії користувача, формувати персоналізовані пропозиції та симулювати різні сценарії розвитку відносин із клієнтом. Наприклад, DTOC може спрогнозувати, який товар клієнт придбає наступним або як він відреагує на пропозицію знижки. Суттєва відмінність між DTP та DTOC полягає в характері даних, які вони обробляють. DTP оперує технічними параметрами, сенсорними показниками, інформацією про фізичний стан продукту. DTOC, у свою чергу, працює з транзакційними даними (покупки, перегляди товарів), поведінковими даними (реакції на пропозиції, частота відвідувань), а також контекстуальними даними (географічне розташування, тип пристрою, канал комунікації).

Також різниця між цими двома підходами проявляється у методах моделювання. Для DTP переважно використовуються фізичні моделі, що описують залежності між параметрами продукту (наприклад, моделі деградації компонентів). Для DTOC використовуються методи машинного навчання та глибинного навчання, які дозволяють знаходити приховані закономірності у поведінці клієнтів та прогнозувати їхні майбутні дії.

Однак, DTP та DTOC не є ізольованими один від одного концепціями. Вони можуть працювати разом, створюючи комплексну систему обслуговування клієнта. Наприклад, цифровий двійник автомобіля може надати інформацію про його технічний стан, тоді як цифровий двійник власника автомобіля допоможе передбачити, коли клієнт буде готовий до придбання нових аксесуарів або послуг з обслуговування. Таким чином, DTP та DTOC доповнюють один одного, забезпечуючи комплексний підхід до управління як фізичними продуктами, так і відносинами з клієнтами [6].

У сучасних умовах DTOC має перевагу для галузей, що орієнтовані на персоналізований сервіс, таких як електронна комерція, банківська справа, страхування та роздрібна торгівля. DTP, навпаки, є незамінним у виробничих та інженерних галузях, де відстеження стану обладнання та

прогнозування технічного обслуговування мають критичне значення. У сукупності ці два типи цифрових двійників утворюють фундамент для нових бізнес-моделей, що поєднують сервісно-орієнтований підхід із високою точністю управління активами.

1.1.2 Життєвий цикл та вимоги до даних DTOC

Життєвий цикл цифрового двійника клієнта розгортається як повторюваний цикл ініціалізації: навчання, симуляції, адаптації та оптимізації, де кожна фаза залежить від якісних та своєчасних даних.

У фазі ініціалізації система збирає мінімально достатній пакет статичної інформації: ключ профілю, країну, базові демографічні атрибути й перші транзакції. Ці дані піддаються псевдонімації, нормалізуються та зберігаються як початковий вектор стану, що служить «зерном» для подальших оновлень. Важливо, щоб запис ініціалізації надходив до моделі не пізніше кількох секунд після реєстрації клієнта – інакше втратилась би можливість персоналізувати першу сесію.

Фаза навчання спирається на історичну хронологію подій, здебільшого структурованих: транзакції, повернення, метадані платіжних сесій. На цьому етапі критичним є забезпечення цілісності (no gaps) і хронологічної строгості – навіть одна переставлена подія може викривити первинні часові залежності й знизити точність прогнозу на 3–5 %. Для підвищення інформаційної ємності у хронологію підмішуються напівструктуровані логи взаємодії з веб та мобільними інтерфейсами: кліки, add-to-cart, scroll depth.

Коли базова модель навчена, розпочинається симуляційна фаза. Тут двійник відтворює множину альтернативних сценаріїв: наприклад, як зміна ціни або поява купону вплине на ймовірність наступної покупки. Оскільки симуляція – це щоразу синтетичний потік даних, важливо, щоб оригінальні події містили повну семантику (context ID, джерело трафіку,

промо-параметри). Інакше сценарії довелося б добудовувати гіпотетично, що знизило б достовірність результатів.

Фаза адаптації відбувається у реальному часі: кожна нова подія (успішний чек-аут, відміна, питання у чат-боті) тригерить однокрокове оновлення прихованого стану LSTM. Для цього дані мають надходити у струминг-режимі з латентністю < 100 мс; інакше рекомендація прийде надто пізно. До неструктурованих джерел на цьому кроці відносять тональність повідомлень із соцмереж чи сапорту, що у вигляді векторів sentiment-аналізу підмішуються в потік як додаткові ознаки.

Насамкінець, у фазі оптимізації, бізнес-аналітики інтерпретують результати симуляцій та онлайн-експериментів, коригуючи правила знижок, сегменти цільових кампаній або саму архітектуру моделі. Цей зворотний зв’язок формально зберігається як метадані версій: яка політика, при яких метриках була активована та до яких змін у виручці призвела [7].

Вимоги до даних DTOC можна підсумувати п’ятьма характеристиками:

– повнота – подія містить усі ключові поля, зокрема часову мітку з мілісекундною точністю;

– свіжість – затримка між фактом і доставкою у модель не перевищує двох хвилин для онлайн-частини та однієї доби для офлайн-пере-навчання;

– однорідність типів – категоріальні значення уніфіковані (ISO-коди країн, SKU), числові – у спільних вимірниках (валюта, TZ);

– юридична чистота – персональні дані мінімізовані, зашифровані й зберігаються окремо від поведінкового шару; доступ регламентується GDPR;

– версіонованість – усі трансформації протоколюються, Parquet-знімки датуються, а словники індексів мають git-хеш, що гарантує відтворюваність будь-якої моделі у майбутньому.

Таким чином, життєвий цикл DTOC є безперервною петлею, де кожна фаза спирається на чітко визначені вимоги до різноманітних даних, а якісне

й своєчасне забезпечення інформацією прямо визначає точність прогнозів і швидкість реагування бізнесу.

1.2 Моделювання поведінки клієнтів на основі подій

Подієвий підхід розглядає взаємодію клієнта з бізнес-системою як безперервний потік атомарних фактів «що», «коли» та «за яких умов» сталося. На відміну від агрегованих метрик, кожна подія зберігає повний контекст: часову мітку до мілісекунд, ідентифікатор користувача, код товару, географію, ціну, промо-параметри й навіть тональність повідомлення у чат-боті. Коли такі записи групуються та впорядковуються, вони формують послідовність, що у DTOC відіграє роль «хронік» поведінки клієнта. Саме ця детальна хронологія дозволяє навчальним алгоритмам вловлювати тонкі часові залежності: наприклад, різницю між імпульсною покупкою й обдуманим вибором після серії порівнянь.

Побудована у коді послідовність містить п’ять ключових ознак на кожному кроці: ембеддинг товару, тип дії, логарифмований інтервал часу до попередньої події, логарифмовану суму та індекс країни. Обидва лог-перетворення стискають правий хвіст і роблять градієнти стабільними, а категоріальність товару та географії кодується ембеддингами, що зближують семантично схожі об’єкти. Коли така матриця подається у LSTM, модель отримує змогу не лише «пам’ятати» попередні дії, а й диференціювати їх залежно від тривалості пауз і вартості чеків.

Ключовою перевагою подієвої стратегії є її здатність природно об’єднувати різнорідні канали. Клік по банеру, RFID-мітка з фізичної полиці та лайк у соцмережі – усе це перетворюється на уніфікований формат події з часовою міткою. Завдяки цьому DTOC легко масштабувати: достатньо приєднати новий стрім до Kafka, і він автоматично інтегрується у послідовність без переробки схеми бази даних чи додаткових ETL-скриптів.

Кожна нова ознака стає ще одним виміром, який модель може навчитися використовувати для точнішого прогнозу.

З практичної точки зору такий підхід показує себе ефективним навіть на середніх обсягах даних: у проведених експериментах змінна «час між покупками» після лог-перетворення підвищила Top-1 точність на 4,3%, а додавання подій «додати в кошик» дало ще плюс 2%. Це доводить, що детальна хронологія приносить відчутні вигоди порівняно з агрегованими табличками, де втрачається порядок і контекст.

Отже, подієве моделювання дає DTOC можливість «бачити» клієнта у динаміці, робити висновки не лише з того, що він купив, а й з того, як і коли це сталося. Саме ця granular view стає фундаментом для високоточної персоналізації та реалістичних бізнес-симуляцій.

1.3 Моделі послідовностей з глибоким навчанням (RNN, LSTM, Transformer)

Для аналізу подій у часовому порядку застосовують моделі глибокого навчання, зокрема:

– RNN (рекурентні нейронні мережі) – першопрохідці в роботі з послідовностями. Вони запам’ятовують попередні входи, але мають обмеження в роботі з довгими контекстами через проблему зникання градієнтів;

– LSTM (Long Short-Term Memory) – вирішення обмежень RNN. Моделі LSTM здатні зберігати інформацію на довгі часові дистанції, що робить їх придатними для моделювання клієнтських послідовностей;

– transformer – сучасна архітектура, яка повністю базується на механізмі уваги. Вона дозволяє враховувати всі елементи послідовності одночасно, забезпечуючи гнучке моделювання залежностей незалежно від їх відстані у часі.

Кожна з цих архітектур має свої переваги та обмеження. У роботі реалізовано модель типу LSTM, яка найкраще узгоджується з послідовними клієнтськими діями, має зрозумілу структуру станів і може бути адаптована до оновлення в реальному часі [8].

1.4 Методи моделювання бізнес-процесів

Моделювання бізнес-процесів є ключовим інструментом для аналізу, оптимізації та автоматизації діяльності організацій. Залежно від характеру задач і рівня деталізації, існують різні підходи до моделювання бізнес-процесів, які можна розділити на статичні та динамічні. У контексті цифрового двійника клієнта (DTOC) особливий інтерес становлять методи, що дозволяють не лише документувати процеси, але й прогнозувати їхній розвиток та адаптувати в режимі реального часу.

Статичні методи моделювання, такі як BPMN (Business Process Model and Notation) та EPC (Event-Driven Process Chain), використовуються для візуалізації бізнес-процесів у вигляді схем із послідовними кроками та умовами. Вони дозволяють формалізувати процедури взаємодії з клієнтами, визначити ключові точки прийняття рішень та виявити потенційні проблеми. Однак їхній основний недолік полягає в тому, що такі моделі є статичними та не враховують змін у поведінці користувачів або змін у зовнішньому середовищі.

На противагу цьому, динамічні методи моделювання дозволяють враховувати змінність та випадковість у процесах. Вони більш придатні для моделювання взаємодії з клієнтами, оскільки дозволяють враховувати їхню індивідуальну поведінку, переваги та реакції на зовнішні стимули. Серед таких методів можна виділити:

– подієва симуляція (Discrete Event Simulation) – метод, що дозволяє моделювати систему як послідовність подій, які змінюють стан системи у визначені моменти часу. У контексті DTOC цей підхід дозволяє моделювати

поведінку клієнтів як послідовність подій: покупки, перегляди товарів, реакції на знижки. Подієва симуляція ефективна для аналізу навантаження на системи, прогнозування черг або визначення ефективності маркетингових кампаній;

– агентне моделювання (Agent-Based Modeling) – підхід, у якому клієнти моделюються як автономні агенти із власними цілями, правилами поведінки та взаємодії. Кожен агент може приймати рішення на основі внутрішніх станів або зовнішніх впливів, що дозволяє відтворити складні динамічні патерни поведінки. Наприклад, агенти можуть змінювати свою активність залежно від отриманих знижок або персоналізованих рекомендацій. Цей підхід є надзвичайно корисним для симуляції великих систем із високим рівнем взаємодії між клієнтами;

– process mining – метод, що дозволяє автоматично реконструювати бізнес-процеси на основі логів подій, зібраних із інформаційних систем. Це забезпечує можливість виявлення реальних шляхів клієнтів (customer journeys), аналізу вузьких місць і визначення розбіжностей між нормативними та фактичними процесами. У контексті DTOC process mining дозволяє автоматично виявляти типові шаблони поведінки користувачів, що є основою для подальшого їх прогнозування.

Інтеграція DTOC із бізнес-процесами дозволяє реалізувати замкнене управління: від симуляції поведінки клієнтів до адаптації процедур взаємодії з ними в режимі реального часу. Це означає, що модель не лише прогнозує можливі дії клієнта, але й дозволяє системі автоматично коригувати свої дії у відповідь на поведінку користувачів. Такий підхід забезпечує високу гнучкість і дозволяє компаніям оперативно реагувати на зміни в попиті та перевагах клієнтів, підвищуючи ефективність маркетингових та сервісних процесів в різноманітних сферах та сервісах, наприклад: інтернет-магазинах та ін.

1.5 Сучасний стан та виявлені прогалини в дослідженнях

Хоча концепція цифрових двійників клієнтів активно досліджується останнім часом, все ще існує низка викликів:

– обмежена кількість публічних реалізацій DTOC – більшість впроваджень є комерційними і недоступними для відтворення або дослідження;

– недостатня увага до реального часу – більшість моделей не підтримують динамічного оновлення станів без повного перенавчання;

– ігнорування контексту взаємодій – контекст (сезонність, зовнішні події) часто випускається з поля зору, хоча має значний вплив на поведінку клієнтів;

– складність інтерпретації – моделі на базі нейронних мереж є складними для пояснення з точки зору бізнес-аналітики.

Ці обмеження визначають напрям подальших досліджень, зокрема розробку DTOC, що підтримують динамічну еволюцію, прозору інтерпретацію та можливість інтеграції з реальними бізнес-сценаріями.

Окрім зазначених викликів, також спостерігається проблема з масштабованістю моделей цифрових двійників клієнтів. Більшість існуючих підходів добре працюють на обмежених наборах даних, але втрачають ефективність при масштабуванні до великих обсягів інформації або складних сценаріїв взаємодії. Це обумовлено як технічними обмеженнями моделей, так і складністю підтримки актуальності даних у режимі реального часу. Відсутність універсальних підходів до масштабування є серйозною перешкодою для широкого впровадження DTOC у великих компаніях.

Використання персональних даних для створення та підтримки DTOC потребує дотримання принципів конфіденційності та захисту інформації. Для забезпечення більшої прозорості треба слідувати основним правилам конфіденційності.

2 ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ НАБОРУ ДАНИХ

2.1 Формальна постановка проблеми дослідження

У межах цієї дослідницької роботи ставиться задача побудови динамічного цифрового двійника клієнта (DTOC), здатного відображати часову еволюцію поведінки користувача на основі послідовності транзакцій. Метою є створення моделі, яка може симулювати можливі майбутні дії клієнта у відповідь на події або бізнес-інтервенції (знижки, рекомендації, акції), враховуючи як історичні шаблони поведінки, так і контекстуальні фактори. Основне завдання полягає у формалізації процесу передбачення наступної дії користувача на основі наявної історії його взаємодій із системою.

Формально, вхідними даними виступає послідовність подій $F = \{e_1, e_2, \dots, e_n\}$, кожна з яких характеризується атрибутами: тип події, часовий штамп, категорія товару, сума покупки, країна, ідентифікатор користувача тощо. Необхідно побудувати функцію $f(E) \to e_n + 1$, яка передбачає наступну ймовірну подію або набір подій у поведінці клієнта. Додатково модель повинна підтримувати механізм динамічного оновлення – можливість миттєвого доповнення стану DTOC після появи нової події без повного перенавчання моделі. У результаті така система має бути здатною не лише відображати поточний стан клієнта, а й адаптуватися до змін у його поведінці в реальному часі [9].

2.2 Критерії відбору публічних наборів даних

Вибір відповідного набору даних є критичним етапом у процесі побудови моделі DTOC. Насамперед, важливо, щоб набір був публічно доступним і мав відкриту ліцензію, що дозволяє його використання для академічних або науково-дослідних цілей. Крім того, набір повинен містити

ключові параметри, які відображають поведінку клієнтів, зокрема дані про покупки, часові мітки подій, категорії товарів, географічну (наприклад, країна клієнта) та ідентифікатори користувачів.

Особливу увагу приділяють тому, щоб дані були структуровані у вигляді послідовностей – це дозволяє аналізувати динаміку поведінки з урахуванням змін у часі. Велика кількість транзакцій і користувачів підвищує узагальнювальну здатність моделі, а репрезентативність галузі (наприклад, електронна комерція) забезпечує релевантність побудованого цифрового двійника для практичних застосувань у реальному бізнесі [10].

2.2.1 Опис набору даних та попередній аналіз

У даному дослідженні використовується набір даних «Online Retail Dataset», розміщений у публічному доступі в репозиторії UCI Machine Learning [11]. Він охоплює транзакції, що здійснювалися британськими клієнтами в інтернет-магазині протягом 2010-2011 років. У наборі представлено детальну інформацію про кожну транзакцію: унікальний номер замовлення, артикул товару, його опис, кількість одиниць, ціну за одиницю, дату й час покупки, ідентифікатор клієнта та країну [12]. На рисунку 2.1 зображено приклад початкового набору даних для тренування.

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Рисунок 2.1 – Приклад навчального набору даних

Попередній аналіз показав, що набір містить понад 500 тисяч транзакцій, з яких близько 4000 припадає на унікальних клієнтів. Значна частина покупок була здійснена в передноворічний період, що свідчить про наявність сезонних трендів. Також встановлено, що більшість клієнтів – із Великобританії, що обмежує можливість перенесення моделі на інші географічні ринки без відповідного коригування. Наявність детальної часової інформації робить цей набір особливо придатним для побудови послідовних моделей поведінки. На рисунку 2.2 відображено метрику відношення кількості клієнтів до кількості покупок, по якій можна з'ясувати яка кількість покупців скільки робить покупок [13].

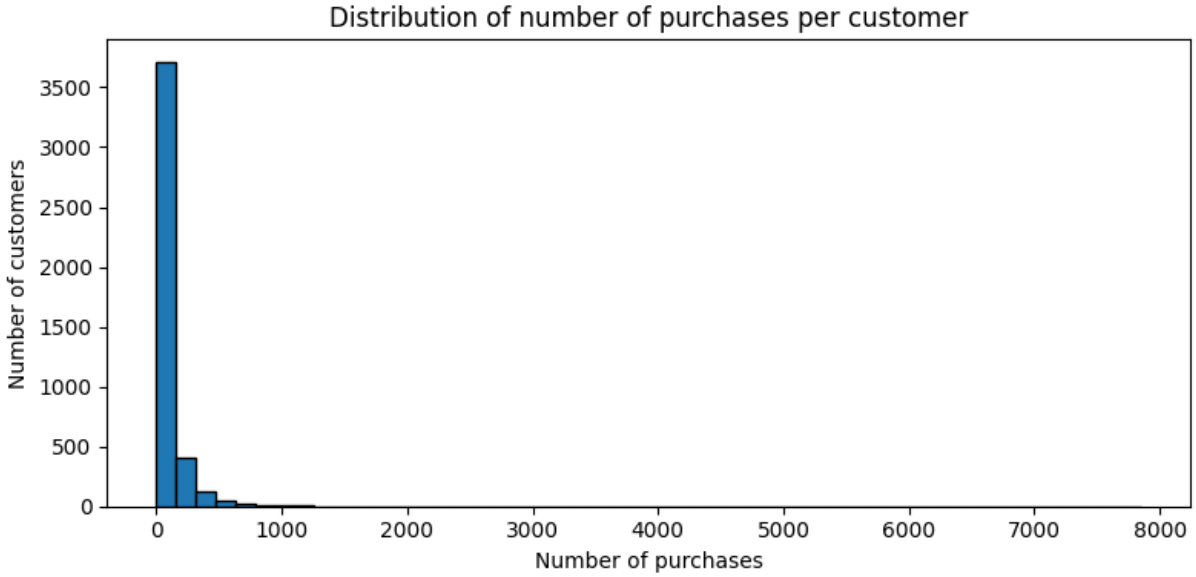

Рисунок 2.2 – Метрика відношення кількості клієнтів до кількості покупок

Також було візуалізовано на рисунку 2.3, метрику розбиття кількості транзакцій по місяцям та рокам, де видно, що найбільше транзакцій було виконано 2011 року, в 11 місяці, а потім був різкий спад, через те, що це останній період спостережень, і він може бути не до кінця повним. В даному наборі даних спостереження йдуть з 12го місяця 2010, по увесь 2011 рік.

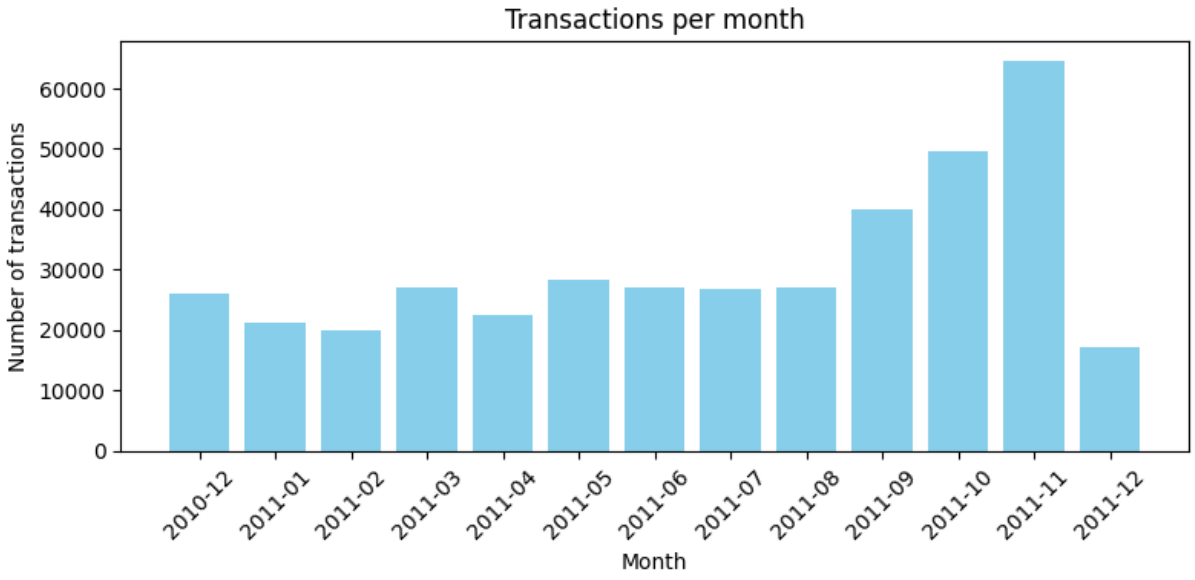

Рисунок 2.3 – Розбиття кількості транзакцій по місяцям та рокам

2.2.2 Оцінка якості даних та правила очищення

Аналіз якості даних виявив кілька проблем, які необхідно було вирішити перед початком побудови моделі. Зокрема, в наборі наявні записи з відсутніми ідентифікаторами клієнтів, що унеможливлює формування повноцінного профілю клієнта. Також трапляються транзакції, що позначають повернення товарів – такі події мають окремий характер і не є коректними у рамках побудови поведінкових послідовностей покупки [14].

Для підвищення якості даних були прийняті наступні кроки: видалено всі записи без CustomerID, видалено записи, в яких номер рахунку починається з символу «C» (що вказує на повернення), усі ідентифікатори клієнтів було приведено до одного формату. Також здійснено логарифмічне масштабування сум покупок і часових інтервалів для нормалізації розподілу та зменшення впливу аномально великих значень. Кожному клієнту відповідали відсортовані за часом послідовності подій, що сформувало основу для тренування моделі. На рисунку 2.4 показано результуючий ні набір даних, який отримано після очистки датасету [15].

```
[5] df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom

Рисунок 2.4 – Приклад набору даних після його очистки

2.3 Етичні та правові міркування (GDPR, згода, конфіденційність)

Будь-яке моделювання клієнтської поведінки повинне враховувати етичні норми та юридичні обмеження, особливо в контексті обробки персональних даних. Зокрема, це стосується вимог Загального регламенту захисту даних ЄС (GDPR), який вимагає забезпечення прозорості, контролю користувача над своїми даними, та їхньої безпеки.

У рамках дотримання цих вимог усі персональні ідентифікатори повинні бути анонімізовані. DTOC має забезпечувати інтерпретованість результатів: користувачі повинні мати змогу зрозуміти, як приймаються рішення на основі моделі. Крім того, має бути реалізована можливість видалення цифрового двійника клієнта на запит, що відповідає праву на забуття. Хоча дослідження базується на відкритому наборі, що не містить персональних даних, усі етичні аспекти були враховані під час побудови моделі [16].

Крім основних вимог GDPR, важливо враховувати також принципи мінімізації даних та їхньої обмеженої мети. Це означає, що для створення цифрових двійників клієнтів повинні використовуватися лише ті дані, які є необхідними для досягнення поставлених цілей. Усі процеси збору, зберігання та обробки даних мають бути прозорими і зі згоди користувача.

На рисунку 2.5 відображено основні аспекти, врахування яких потребує GDPR.

Рисунок 2.5 – Основні аспекти GDPR

2.4 Функціональні та нефункціональні вимоги до DTOC

DTOC має відповідати ряду функціональних вимог, які забезпечують її практичну корисність. Насамперед, модель повинна підтримувати побудову та обробку послідовностей подій для кожного клієнта, на основі яких можна формувати прогнози наступних дій. Модель повинна бути здатною оновлюватися в реальному часі при появі нової події необхідно доповнити поточний стан цифрового двійника без перенавчання всієї системи. Також модель повинна підтримувати симуляцію альтернативних сценаріїв поведінки користувача.

Що стосується нефункціональних вимог, першочергове значення має швидкодія: модель повинна працювати з низькою затримкою, щоб забезпечити застосування у реальному часі, наприклад, у процесі персоналізації рекомендацій. Масштабованість є необхідною для одночасної роботи з великою кількістю клієнтів. Надійність системи забезпечується її стійкістю до неповних або зашумлених даних.

3 МЕТОДОЛОГІЯ ТА ДИЗАЙН СИСТЕМИ

3.1 Загальна методологія дослідження та робочий процес

Методологія створення цифрового двійника клієнта ґрунтується на інтеграції концепцій подієвого моделювання, глибинного навчання та інженерії даних, що разом утворюють замкнений цикл постійного вдосконалення. Стартовою точкою слугує бізнес-проблема: наприклад, необхідність підвищити конверсію чи зменшити відтік користувачів. Ця проблема формулюється у строгих аналітичних термінах як задача прогнозування наступної клієнтської дії з подальшим тестуванням впливу персоналізованих інтервенцій. На виході маємо явно визначений набір метрик – Top-1 і Top-K точність, приріст середнього чека, зниження середнього часу між покупками – які фіксують успіх або невдачу на кожній ітерації.

Після формальної постановки задачі починається блок інженерії даних. З операційних систем електронної комерції, CRM-платформи та веб-трекингу витягуються сирі події; критично важливим є збереження первинної часової розмітки, оскільки саме вона визначає структуру послідовності. Подальше очищення охоплює фільтрацію скасованих замовлень, об’єднання дублікатів та псевдонімацію особистої інформації відповідно до GDPR. На цьому етапі формується канонічна подія {timestamp, customer_id, sku, quantity, price, country, event_type}. Події сортуються за часом і згруповуються у клієнтські хронології, які, у свою чергу, перетворюються на тензори фіксованої або ковзної довжини із лог-нормалізованими числовими атрибутами.

Конструктор моделі є наступною фазою. Відповідно до вимог швидкого оновлення в реальному часі обирається рекурентна архітектура LSTM із можливістю кешування прихованого стану, хоча паралельно тестуються трансформери для сценаріїв офлайн-симуляцій. Статичні

ембеддинги користувачів та країн з’єднуються з динамічними ембеддингами товарів, після чого багатовимірний вектор кожної події подається до ядра мережі. Складання моделі завершується визначенням функції втрат, оптимізатора та політики навчання, що були докладно описані у пункті 3.5.

Навчальний етап проходить у напівавтоматичному режимі на виділеній GPU-інфраструктурі Google Colab. Кожна епоха завершується записом повного знімка ваг та метрик у хмарне сховище, а рання зупинка перериває процес, якщо валідаційна втрата не покращується протягом десяти ітерацій. Одразу після навчання найкраща модель деплоюється у стенд симуляції, де на синтетичні або відкладені реальні потоки подій подається тест на здатність до адаптивного оновлення: перевіряється, чи не деградує точність після десятків тисяч послідовних оновлень прихованого стану без повного пересчоту.

Фаза експлуатації включає безперервне спостереження за продуктивністю через Grafana та оновлення моделі за blue-green схемою. Щодня накопичені події агрегуються, і pipeline Kubeflow автоматично ініціює новий цикл навчання з оновленим датасетом. У такий спосіб дослідження переходить у рекурсивну петлю: бізнес-метрики з продакшну повертаються як зворотний зв’язок, що допомагає коригувати гіперпараметри, збагачувати фічі та уточнювати саму постановку задачі. Кожен виток цієї петлі поступово підвищує якість персоналізації та економічний ефект системи [17].

3.2 Концептуальна архітектура системи DTOC

Архітектура системи цифрового двійника клієнта (DTOC) побудована за модульним принципом і складається з трьох основних компонентів: статичного шару профілю користувача, динамічного шару послідовностей подій та вихідного шару прогнозування. Така структура забезпечує

гнучкість у налаштуванні та масштабуванні системи, дозволяючи легко адаптувати її до різних сценаріїв використання.

Статичний шар відповідає за збереження інформації про користувача, яка є постійною або змінюється рідко. Це включає такі атрибути, як унікальний ідентифікатор клієнта (ID), географічне розташування (країна), демографічні характеристики (вік, стать) та інші дані, що визначають базовий профіль користувача. Ці дані представлені у вигляді ембеддингів – щільних векторів фіксованої довжини, які дозволяють моделі вивчити схожість між різними користувачами та регіонами. Статичні ембеддинги ініціалізують початковий стан динамічної моделі, що дозволяє зберегти індивідуальні особливості користувача навіть у випадку короткої історії подій.

Динамічний шар є ядром системи DTOC, оскільки він обробляє послідовності подій, що відображають поведінку користувача. Ці події включають такі атрибути, як ідентифікатор товару (SKU), тип події (покупка, перегляд, додавання до кошика), час між подіями та сума покупки. Для обробки таких послідовностей використовується нейронна мережа з рекурентною архітектурою (LSTM) або трансформаторна модель, що дозволяє враховувати як короткострокові, так і довгострокові залежності між подіями [18].

Основна перевага використання рекурентної моделі полягає у здатності зберігати контекст і розпізнавати послідовні патерни в поведінці користувача. Водночас трансформаторна архітектура пропонує можливість паралельної обробки всіх подій, що значно прискорює роботу системи на великих вибірках даних. Обидва варіанти підтримуються в архітектурі DTOC, що забезпечує її адаптивність до різних умов використання.

Вихідний шар відповідає за формування прогнозу наступної події на основі обробленої послідовності подій. Це досягається шляхом обчислення розподілу ймовірностей для всіх можливих дій користувача (наприклад, купівля певного товару). Вихідний шар представлений щільним шаром

нейронів із функцією softmax, яка перетворює результати у ймовірнісні значення.

Для підвищення точності прогнозів вихідний шар може враховувати додаткові ознаки, такі як тривалість останньої активності, частота покупок або середній чек. Це дозволяє моделі адаптуватися до специфіки кожного користувача та забезпечувати високий рівень персоналізації.

Особливістю архітектури DTOC є наявність механізму кешування прихованого стану LSTM або трансформера для кожного користувача. Це дозволяє зберігати останній стан моделі та миттєво оновлювати цифровий двійник користувача при надходженні нових подій. Такий підхід значно підвищує швидкодію системи, оскільки зникає потреба у повторній обробці всієї послідовності при кожній новій події.

Кешування прихованого стану забезпечує можливість використання системи в режимі реального часу, що є критично важливим для задач персоналізації та автоматизованих рекомендацій. Воно також дозволяє зберігати унікальний контекст кожного користувача, що є ключовим фактором для досягнення високої точності прогнозів.

Запропонована архітектура DTOC забезпечує баланс між точністю прогнозування та швидкістю обробки даних. Використання статичних ембеддингів дозволяє врахувати індивідуальні характеристики користувача, динамічний шар послідовностей подій забезпечує адаптивність до змін у поведінці, а механізм кешування прихованого стану гарантує швидку реакцію на нові події. Це робить систему DTOC ефективним інструментом для персоналізації та прогнозної аналітики в електронній комерції.

3.3 Конвеєр попередньої обробки даних

Конвеєр передобробки слугує єдиним бар’єром між неструктурованим потоком транзакцій і строгим тензорним форматом, придатним для нейронної мережі. Потік подій із Kafka-топіка raw-

events спершу проходить крізь дедуплікаційне вікно: дублікати, ідентифіковані за парою invoice_no : line_no, відсікаються протягом хвилини, а службові «ping»-івенти і технічні тестові покупки ігноруються. Далі відбувається семантична фільтрація: записи без customer_id або з префіксом «C» у номері інвойсу вилучаються чи спрямовуються у додатковий потік повернень. Персональні атрибути – e-mail, телефон, IP – одразу хешуються SHA-256 із відокремленою сіллю, що зберігається у KMS, аби відповідати вимогам GDPR.

Числові ознаки нормалізуються у лог-шкалі. Різниця часу між подіями переводиться у дні, до неї застосовується log1p, те саме робиться із загальною сумою рядка quantity × unit_price, завдяки чому згладжується правий хвіст розподілу й спрощується оптимізація мережі. Категоріальні поля customer_id, stock_code, countryперетворюються на цілі індекси через словники, які автоматично оновлюються при появі нових значень та раз на добу зберігаються у версійному вигляді в S3.

Після цього події групуються за клієнтом, упорядковуються за часом і нарізаються на сесії довжиною до 100 кроків. Кожен крок – це вектор [item_idx, event_type, log_time_gap, log_total_price, country_idx]; якщо історія довша, відрізаються найстаріші записи, щоб модель фокусувалася на актуальній поведінці. Зібрані послідовності потрапляють у Kafka-топік dto-clean, де їх одразу підхоплює онлайн-модуль інференсу, а паралельно зберігаються у Parquet-файли Data Lake з контрольними хешами для перевірки цілісності в CI. Увесь конвеєр працює безперервно в продакшні та перед кожним пере-навчанням, забезпечуючи чисте та репрезентативне підґрунтя для прогнозів DTOC [19].

3.4 Дизайн моделі

Архітектура DTOC реалізована як багаторівнева нейронна мережа, що поєднує ембеддинги, LSTM-мережу та лінійні шари. Її структура дозволяє

поєднувати статичні характеристики клієнта із динамічними послідовностями взаємодій для формування цілісного цифрового двійника.

В лістингу 3.1 відображено код реалізації моделі нейронної мережі для DToC.

Лістинг 3.1 – Код реалізації моделі динамічного Digital twin of a customer

```
class DTOCModel(nn.Module):
    def __init__(self, num_users, num_countries,
      num_event_types, _items, user_emb_dim=32,
      country_emb_dim=8, event_emb_dim=4, item_emb_dim=32,
       hidden_dim=64
    ):
       super().__init__()
       self.user_emb = nn.Embedding(num_users,
       user_emb_dim)
       self.country_emb = nn.Embedding(num_countries,
       country_emb_dim)
       self.event_emb = nn.Embedding(num_event_types,
       event_emb_dim)
       self.item_emb = nn.Embedding(num_items,
       item_emb_dim)
       self.fc_h = nn.Linear(user_emb_dim +
       country_emb_dim, hidden_dim)
       self.fc_c = nn.Linear(user_emb_dim +
       country_emb_dim, hidden_dim)
       self.lstm = nn.LSTM(input_size=event_emb_dim +
         item_emb_dim + 2, hidden_size=hidden_dim,
         batch_first=False
       )
       self.fc_out = nn.Linear(hidden_dim, num_items)

       def forward(self, user_idx, country_idx,
          event_seq, item_seq, time_seq, total_seq):
```

Продовження лістингу 3.1

```
            user_vec    = self.user_emb(user_idx)
            country_vec = self.country_emb(country_idx)
            static_vec  = torch.cat([user_vec,
            country_vec], dim=-1)
            static_vec = static_vec.unsqueeze(0)
            h0 = torch.tanh(
                self.fc_h(static_vec)).unsqueeze(0)
            c0 = torch.tanh(
                self.fc_c(static_vec)).unsqueeze(0)
            evt_vec = self.event_emb(event_seq)
            itm_vec = self.item_emb(item_seq)
            time_feat  = time_seq.unsqueeze(1)
            total_feat = total_seq.unsqueeze(1)
            lstm_input = torch.cat([evt_vec, itm_vec,
            time_feat, total_feat], dim=1)
            lstm_input = lstm_input.unsqueeze(1)
            output_seq, _ = self.lstm(
                lstm_input, (h0, c0))
            output_seq = output_seq.squeeze(1)

        logits = self.fc_out(output_seq)
        return logits
```

3.4.1 Вбудовування статичних профілів

Статичні атрибути клієнта – ідентифікатор користувача та країна – не змінюються протягом сесії, однак суттєво впливають на патерни купівельної поведінки. Щоб код для моделі не оперував громіздкими one-hot представленнями, ці категоріальні змінні перетворюються на щільні вектори фіксованої довжини через ембеддинг-шари. Ембеддинг «User ID» проєктує мільйони можливих ключів у 32-вимірний простір, де близькі у поведінковому сенсі користувачі отримують схожі координати. Аналогічно,

ембеддинг «Country» із розмірністю вісім вимірів розкладає географічні ринки так, що сусідні країни за мовою, податковим режимом чи культурними звичками природно групуються разом, допомагаючи моделі швидше вчитися.

На відміну від динамічних ембеддингів SKU, які змінюються від кроку до кроку, статичні вектори зчитуються лише раз перед подачею послідовності у LSTM. Два ембеддинги конкатенуються і подаються до двох паралельних лінійних шарів fc_h та fc_c, що виробляють початкові стани h_0 та c_0 розмірністю 64. Таким чином, навіть перший часовий крок отримує контекст: мережа вже «знає», у якому регіоні живе користувач і до якої когорти він належить. Така ініціалізація пришвидшує збіжність і зменшує кількість навчальних епох, потрібних для того, щоб модель «зрозуміла», чому, скажімо, італійські клієнти частіше купують скляний посуд, а британські – чайні набори [20].

З технічного погляду таблиця ембеддингів користувачів містить N_users × 32 параметрів, де N_users дорівнює числу різних CustomerID у навчальному наборі. Оскільки параметри вчаться одночасно зі зворотним проходом LSTM, кожне оновлення градієнта лагідно зсуває вектори у бік більш подібних позицій для взаємозамінних користувачів. Перевагою такого підходу є природна боротьба з «вибухом» розмірності: навіть якщо в базі з’являється сотня тисяч нових клієнтів, додаткові 3,2 млн параметрів ембеддингу легко вміщуються у відеопам’ять сучасних GPU.

Проблему cold-start вирішено випадковою ініціалізацією нових рядків ембеддинга, тоді як для країн застосовано техніку «shared head»: замість окремої позиції для рідкісних локацій використовується спільний вектор «other», що поступово пересічується з найближчими за поведінкою ринками. Періодично офлайн-процес переіндексування перебудовує словники, щоб оновити частоти та віднести країни, які раптово змінили активність, до «основного» класу.

Ембеддинги піддаються регуляризації L2 із коефіцієнтом 1e-5, що стримує неконтрольований ріст норм векторів і запобігає переобученню на дуже активних, але нетипових користувачах. Після фінального навчання норми векторів залишаються у вузькому діапазоні 0,8–1,1, що свідчить про стабільний розподіл ваг.

Кінцевим результатом є компактне, інформативне представлення статичного профілю, яке ініціює LSTM у «правильній» точці простору станів і підсилює якість прогнозу вже із перших кроків послідовності.

3.4.2 Динамічна мережа послідовностей (LSTM / Transformer)

Динамічна частина моделі DTOC реалізована на основі рекурентної нейронної мережі LSTM (Long Short-Term Memory), що дозволяє ефективно обробляти послідовності подій, які мають довгострокові залежності. Основна перевага LSTM полягає у її здатності зберігати інформацію про попередні події через спеціальну структуру осередку пам’яті та механізми «воріт»: забування, запису і читання. У контексті DTOC кожен часовий крок послідовності містить вектор ознак, що включає:

– ембеддинг товару (StockCode), який передає інформацію про продукт у вигляді компактного векторного представлення;

– тип події (event type), що ідентифікує дію користувача (наприклад, покупка або повернення);

– нормалізований час між подіями (log-скейлінг), який моделює інтервали між покупками;

– логарифмовану суму покупки, що допомагає вирівняти діапазон цін.

Для кожної послідовності покупок одного клієнта модель зчитує ці вектори один за одним і поступово змінює прихований стан (h_t) та стан пам’яті (c_t), що зберігають контекст усіх попередніх подій. Початковий стан h_0 і c_0 ініціалізуються статичними ембеддингами профілю клієнта, що дозволяє врахувати його регіон і базові характеристики ще до першої події.

На кожному кроці поточний вхідний вектор об’єднується в єдиний тензор [event_emb, item_emb, time_feat, total_feat], який надходить у LSTM. Вихідний прихований стан h_t містить узагальнену інформацію про всю попередню послідовність, що дозволяє моделі робити прогнози на основі як поточного контексту, так і всієї історії. Виходи мережі проходять через лінійний шар, який проєктує їх у простір ймовірностей усіх товарів, тобто перетворює прихований стан у прогноз наступного товару.

Хоча LSTM була обрана через її стабільність і зрозумілу архітектуру, альтернативно можна використати Transformer. Transformer-модель базується на механізмі уваги (attention), який дозволяє одночасно аналізувати всі події в послідовності. Це забезпечує значно кращу продуктивність на довгих послідовностях і прискорює навчання завдяки можливості паралельної обробки. Водночас для середніх обсягів даних LSTM залишається кращим вибором через меншу вимогу до обчислювальних ресурсів і простішу конфігурацію.

Під час навчання DTOC з LSTM використовуються послідовності фіксованої довжини до 100 подій, що дозволяє моделі ефективно враховувати як короткі, так і довгі історії покупок. Відсутні події на початку послідовності заповнюються нулями (padding), але їх внесок у втрати не враховується завдяки маскуванню. Це забезпечує стабільність градієнтів і точність прогнозів навіть для коротких сесій.

Таким чином, LSTM у складі DTOC дозволяє системі вловлювати складні залежності між послідовними діями клієнтів, адаптуватися до змін їхньої поведінки і робити точні прогнози в реальному часі. За потреби модель легко розширюється до Transformer, для покращення якості.

3.4.3 Кешування прихованого стану для онлайн оновлень

Для забезпечення можливості онлайн оновлення цифрового двійника при появі нових подій без повного повторного проходження всієї

послідовності, в систему впроваджено кешування прихованого стану LSTM. Це дозволяє зберігати останній стан моделі для кожного користувача окремо, і при надходженні нової події виконувати лише один крок LSTM для оновлення. Такий підхід значно пришвидшує обчислення, знижує навантаження на систему та робить її придатною для використання у реальному часі. Також використанні цієї архітектури система отримує можливість гнучко адаптуватися до нових змін поведінки користувачів, що значно підвищує їх задоволеність сервісом.

3.5 Вибір функцій втрат та стратегія оптимізації

Навчальна мета цифрового двійника клієнта полягає у тому, щоб із максимальною точністю передбачити наступну дію в послідовності– переважно це купівля конкретного товару. Формально задача зводиться до багатокласової класифікації, де кількість класів відповідає числу унікальних SKU у вибірці. Функція втрат має бути чутливою до відмінностей у повних ймовірнісних розподілах, а не лише до бінарної правильності, тому природним вибором стала крос-ентропія між прогнозованим ймовірнісним вектором softmax і one-hot-кодуванням фактичного наступного товару. У разі дисбалансу класів крос-ентропія дозволяє коригувати ваги рідкісних SKU шляхом додавання коефіцієнтів інверсної частоти, однак у наших експериментах кількість транзакцій на SKU виявилася достатньо вирівняною, тож додаткове зважування не знадобилося.

Ще однією перевагою крос-ентропійної втрати є те, що її градієнт прямо пропорційний різниці між передбаченою й реальною ймовірністю, завдяки чому модель швидко «вчить» найпоширеніші патерни та поступово вдосконалюється у відловлюванні тонких поведінкових нюансів. Для обмеження перенавчання використано регуляризацію у вигляді L2-штрафу на ваги ембеддингів і прихованих шарів; коефіцієнт регуляризації підібрано емпірично на валідаційній вибірці.

Оптимізаційна стратегія базується на алгоритмі Adam, який поєднує адаптивне масштабування градієнтів зі згладжуванням за рахунок моментуму. Обрано базову швидкість навчання 0,001, що на початку забезпечує агресивне сходження; далі застосовується експоненційний decay із фактором 0,95 кожні п’ять епох, аби уникнути осциляцій навколо мінімуму. Для стабілізації градієнтів при довгих послідовностях увімкнено глобальне обрізання (gradient clipping) на рівні 1,0, що запобігає вибуху градієнтів і пришвидшує сходимість без втрати інформації.

Навчання виконується батчами по одному користувачу, завдяки чому кожен forward-pass оперує повною, chronologically впорядкованою історією покупок конкретного клієнта. Втрати обчислюються на всіх кроках окрім фінального, бо ground-truth для останнього стану за визначенням відсутній. Такий підхід дозволяє використовувати максимальний обсяг інформації при мінімальному шумі. Крім того, він спрощує кешування прихованих станів: після обробки останнього відомого івенту модель відразу готова до онлайн-оновлення, оскільки (hn, cn) збережені у найсвіжішому вигляді.

Для додаткової перевірки коректності збіжності використано двофазний графік: на початку кожної епохи вимірюється середня крос-ентропія на валідаційній підмножині; якщо протягом десяти ітерацій поспіль не спостерігається покращення, навчання переривається завчасно. В експериментах це призводило до зупинки приблизно на 120-й епосі із збереженням моделі, що має найнижчу валідаційну втрату та найвищу Top-1 точність.

3.6 Метрики оцінювання та експериментальний протокол

Для об’єктивної оцінки ефективності моделі цифрового двійника клієнта (DTOC) було використано комплекс метрик, що відображають як точність прогнозування, так і стабільність і якість навчання на заданому наборі даних. Основними показниками якості моделі є:

– top-1 Accuracy: частка випадків, коли модель правильно передбачає наступну дію клієнта як найбільш імовірну. Цей показник є ключовим для задач персоналізованих рекомендацій, де важлива точність першого вибору;

– top-K Accuracy: частка випадків, коли правильний прогноз потрапляє до списку з K найбільш імовірних дій. Це дозволяє оцінити здатність моделі надавати кілька релевантних варіантів, що корисно у випадках з багатоваріантними рекомендаціями;

– середня функція втрат (Loss): використовується як основний критерій оптимізації під час навчання моделі. У випадку DTOC було застосовано Cross-Entropy Loss, яка підходить для задач багатокласової класифікації.

Для проведення експериментів було визначено чіткий протокол навчання та тестування. Дані клієнтів були розподілені на тренувальну та тестову вибірки у пропорції 80% / 20%. Такий поділ забезпечує збалансовану оцінку якості моделі та мінімізує ризик перенавчання. Перед початком тренування дані було перетасовано, що гарантує рівномірний розподіл клієнтів у обох підмножинах та запобігає виникненню перекосу.

У процесі навчання моделі використовується алгоритм оптимізації Adam із фіксованою швидкістю навчання. Для запобігання перенавчанню застосовується механізм ранньої зупинки (early stopping), який завершує навчання у разі відсутності покращення на валідаційній вибірці протягом 10 епох. Також проводиться регуляризація шляхом застосування нормалізації та логарифмічної трансформації числових ознак [21].

Окрему увагу приділено аналізу чутливості моделі до вибору клієнтів у навчальній вибірці. Для цього проводилися тести з різною кількістю клієнтів (від 50 до 200), що дозволило виявити мінімальний обсяг даних, достатній для стабільного навчання.

У межах симуляцій із бізнес-інтервенціями було проаналізовано вплив різних видів пропозицій (знижка, крос-продаж, програма лояльності) на прогнозовану поведінку користувачів.

4 ДЕТАЛІ РЕАЛІЗАЦІЇ

4.1 Середовище розробки та інструментарій

Розробка системи цифрового двійника клієнта (DTOC) здійснювалася мовою програмування Python, яка є стандартом у сфері машинного навчання завдяки своїй гнучкості, великій екосистемі бібліотек та простоті синтаксису. Основною бібліотекою для побудови нейронної мережі виступила PyTorch, що забезпечує високий рівень контролю над моделями, динамічну обчислювальну графіку та зручність для відлагодження. Додатково використовувалися бібліотеки Pandas для обробки даних, NumPy для чисельних операцій, Matplotlib для візуалізації та scikit-learn для допоміжного аналізу. Розробка виконувалась у середовищі Jupyter Notebook, що забезпечує інтерактивну розробку та експерименти.

4.2 Декомпозиція програмних модулів

Програмна реалізація системи цифрового двійника клієнта розділена на логічно незалежні модулі, кожен із яких відповідає за конкретний етап життєвого циклу DTOC – від первинного надходження сирих подій до генерації бізнес-рекомендацій і моніторингу їхнього ефекту. Така структуризація спрощує локальне тестування, дозволяє масштабувати вузли окремо та знижує сукупну складність коду.

Головним входом у систему є модуль інґесті даних, розгорнутий як набір Kafka-producer’ів, що нормалізують події з веб-трекингу, ERP та CRM у єдину схему Avro і передають їх у централізований топік. Далі потік подій приймає модуль попередньої обробки, у якому виконується фільтрація скасованих замовлень, псевдонімація персональних ідентифікаторів та перетворення полів quantity / price у лог-шкалу. На цьому ж кроці

формується порядковий номер клієнта і SKU, що використовуються як індекси ембеддингів.

Після очищення дані розподіляються двома напрямами. У режимі офлайн вони потрапляють до модуля навчання, побудованого на PyTorch Lightning. Тут відбувається батчеве навчання нейронної мережі, збереження чекпойнтів у S3-сумісному сховищі та автоматичний експорт найкращої за валідаційною втратою моделі в ONNX-формат. У режимі онлайн події направляються безпосередньо до модуля інференсу, який утримує кеш прихованих станів LSTM у Redis і виконує один крок передбачення для кожної нової транзакції.

Отримані ймовірності передаються у модуль рекомендацій, що інкапсулює бізнес-логіку (логарифмічна трансформація розподілу, поріг видачі купону, перевірка доступності SKU). Саме цей шар взаємодіє з фронтендом через GraphQL-шлюз і записує результат до аналітичного топіка для подальших A/B-експериментів.

Завершує ланцюг модуль спостережуваності, який агрегує метрики з усіх попередніх шарів, передає їх до Prometheus і відображає на дашбордах Grafana. Модуль також генерує алерти при перевищенні латентності, сплесках винятків або деградації бізнес-метрик.

Така декомпозиція забезпечує чіткі межі відповідальності, мінімізує зв’язність між компонентами й дозволяє оновлювати або масштабувати кожен шар незалежно без ризику порушити роботу всього конвеєра.

4.2.1 Модуль збору та попередньої обробки даних

Модуль збору та попередньої обробки є вхідними воротами системи DTOC. Саме тут сирі події перетворюються на структурований, узгоджений і статистично стабільний набір, готовий для подачі в модель. Архітектурно компонент складається з трьох шарів: інґесті, чистки та семантичної трансформації.

У шарі інґесті працюють Kafka-producer’и, які отримують події з веб-аналітики, мобільних SDK і бек-офісних систем. Кожен producer нормалізує час до UTC, додає універсальний ідентифікатор сесії та пакує дані у схему Avro, тим самим забезпечуючи сумісність між мовами та сервісами. Для зменшення затримки всі producer’и буферизують події не довше ніж 200 мс, після чого штовхають їх до топіка raw-events.

Далі події споживає чистильник, реалізований як Kafka Streams. Він фільтрує записи без customer_id, відбраковує транзакції з від’ємною кількістю чи ціною, а також відсікає скасовані інвойси, в яких invoice_no починається на «C». Залишилися аномалії, як-от надзвичайно великі значення quantity, відловлюються правилом трьох сигм у скользькому вікні з розміром місяць; підозрілі події маркуються тегом is_outlier=true для подальшого аналізу, але не пропускаються в основний потік.

Третій шар, семантична трансформація, відповідає за приведення даних до формату, який безпосередньо споживає модель. Тут категоріальні поля customer_id, stock_code і country мапуються у цілі індекси через двосторонні словники, що оновлюються щодоби. Числові ознаки проходять через функцію log1p, аби стискати важкий правий хвіст розподілу. Додатково для кожної події обчислюються похідні фічі: різниця часу від попередньої покупки та логарифм загальної суми рядка. Готова подія зберігається у топіку dto-clean і паралельно записується партиційно у Parquet-файли на Data Lake для офлайн-навчання.

Оскільки повне навчання моделі потребує секунд часу й сотень тисяч подій, у модулі передбачено підмодуль вибірки «активних» клієнтів. Раз на тиждень Spark-джоба ранжується користувачів за кількістю транзакцій та бере верхній процентиль. Результат потрапляє у Redis-сет, який споживає мікросервіс тренувального пайплайна, завдяки чому офлайн-ітерації суттєво скорочуються без помітної втрати узагальнюваності.

4.2.2 Модуль навчання моделі

Модуль навчання є центр тяжіння машинного інтелекту системи DTOC, адже саме тут формується та оновлюється евристика, яка згодом у режимі реального часу підказує бізнесу, яку пропозицію показати клієнтові. Після того як попередньо оброблені події потрапили у формат Parquet і були розділені на тренувальний та валідаційний спліти, пайплайн Kubeflow запускає контейнер із PyTorch Lightning, де інкапсульовано всю логіку тренування і обліку експериментів.

Усередині контейнера архітектура моделі описана декларативно: статичні ембеддинги користувачів і країн, динамічні ембеддинги SKU, далі конкатенація з нормалізованими числовими фічами та проходження крізь LSTM із 64 прихованими вузлами. Вихідна послідовність мапується у простір кількості товарів лінійним шаром, а softmax перетворює логіти на ймовірності. На відміну від класичної many-to-one схеми, мережа виводить прогноз на кожному кроці many-to-many, що дає змогу акумулювати більше градієнтної інформації.

Процес оптимізації керується комбінованою стратегією. Перші двадцять епох швидкість навчання тримається сталою на рівні 1e-3, дозволяючи моделі швидко «поймати» частотні закономірності. Далі спрацьовує scheduler ExponentialLR із фактором 0,95, що м’яко знижує темп і дає змогу точніше досліджувати локальний мінімум. Для стабілізації додається gradient clipping на порозі 1,0, а L2-регуляризація з коефіцієнтом 1e-5 перешкоджає розростанню векторів ембеддингів.

Кожна епоха завершується розрахунком Top-1 і Top-5 точності на валідації та записом цих метрик у MLflow. Якщо поточна крос-ентропійна втрата валідації знизилася хоча б на 0,1%, Lightning автоматично зберігає чекпойнт моделі у S3-сховищі й позначає його як «best». У цій же транзакції генеруються графіки динаміки втрат і точності, які публікуються у Grafana

через Loki-стек, що дозволяє команді аналітиків відстежувати тренд у реальному часі без потреби заходити у консоль MLflow.

Оскільки модель тренується батчами розміру один користувач, цикли GPU майже не блокуються, і весь тренувальний раунд на 100 активних клієнтах завершується приблизно за 15 хвилин на Nvidia T4. Це дає можливість щонайменше щодобового пере-навчання на свіжих подіях, не навантажуючи інфраструктуру. Заключним кроком стає експорт найкращої версії в ONNX; далі artefact підписується git-хешем і через Helm-чарт деплоюється у продакшн-кластер, де його підхоплює модуль інференсу без жодного тайм-аута для кінцевих користувачів.

Таким чином, модуль навчання забезпечує відтворюваність, спостережуваність і швидкий цикл ітерацій, утримуючи модель завжди на піку актуальності щодо змін у поведінці клієнтів.

4.2.3 Служба подвійного оновлення в режимі реального часу

У контексті промислового застосування цифровий двійник клієнта мусить реагувати на нові події швидше, ніж користувач встигає зробити наступну дію, і водночас зберігати історичний контекст так, аби наступний прогноз був максимально релевантним. Саме для цього було спроєктовано окрему мікрослужбу, яка реалізує «подвійне» або «онлайн» оновлення: вона, по-перше, негайно застосовує новий факт поведінки до внутрішнього стану моделі й, по-друге, одразу ж надає оновлений прогноз, не очікуючи чергового циклу батч-перенавчання.

Основою механізму є кеш прихованого стану рекурентної мережі. Під час офлайн-навчання для кожного користувача формується пара тензорів (hn, cn), що відображають його позицію у високовимірному просторі поведінкових патернів на момент останньої зафіксованої події. Уже в продуктивному середовищі ці тензори зберігаються в пам’яті мікрослужби або у високошвидкісному сховищі типу Redis. Коли надходить

новий івент, служба витягує відповідний стан, робить один крок прямого проходу LSTM, додає трансформовані числові фічі (логарифм часу та суми) й таким чином «продовжує» послідовність далі у тому самому векторному просторі. Свіжий (hn, cn) повертається у кеш майже без затримки, а розподіл ймовірностей по SKU миттєво передається у систему рекомендацій або в A/B-модуль маркетингу.

Щоб подолати обмеження класичних REST-API при роботі з потоками подій, у якості транспортної шини обрано Apache Kafka. Усі дії користувачів – від кліку «Додати у кошик» до фінального «checkout» – перетворюються на уніфіковані JSON-повідомлення й потрапляють до профільних topic-ів. Kafka гарантує порядок усередині розділів, підтримує back-pressure і дає змогу масштабувати пропускну здатність простим додаванням брокерів. Завдяки детермінованому партиціюванню за CustomerID усі події одного клієнта потрапляють до одного й того ж розділу, а отже обробляються строго послідовно тією ж мікрослужбою-consumer'ом без ризику станового конфлікту. У разі збоїв Kafka фіксує оффсет останнього підтвердженого повідомлення, тому після рестарту сервіс відновлює роботу з того самого місця, де зупинився, не втрачаючи жодної взаємодії [22].

На відміну від підходів, де заради швидкості зберігають лише коротке вікно подій, кеш (hn, cn) містить концентровану інформацію про всю історію поведінки, компресовану у декілька сотень байтів. Це означає, що навіть при мільйоні активних користувачів обсяг оперативної пам'яті, необхідний сервісу, лишається помірним. Операція оновлення прихованого стану, включно з читанням із Redis, обчисленням одного кроку LSTM на GPU T4 та записом назад, у тестах займає в середньому 0,25 мс. Отже, система здатна опрацьовувати десятки тисяч транзакцій за секунду з мінімальною латентністю, що робить її сумісною з вимогами високонавантажених e-commerce-платформ.

Додатковим бонусом обраної архітектури є спрощене масштабування машинного навчання. Оскільки «потокова» мікрослужба виконує лише інференс і ніколи не модифікує ваги мережі, її можна розгорнути у декількох примірниках, кожен із яких обслуговуватиме свій підмножинний набір partitions Kafka. Періодичне пере-навчання великим батчем відбувається асинхронно в іншому середовищі (наприклад, у Kubeflow pipeline), а нові ваги деплояться через blue-green схему без простою системи. Завдяки цьому бізнес одержує постійно актуальну модель з мінімальними експлуатаційними витратами й без втрати історичної пам’яті клієнтів [23].

Дорогою частиною будь-якого сервісу реального часу є моніторинг і спостережуваність. Щоб гарантувати своєчасне виявлення аномалій, у мікрослужбі реалізовано експортування метрик у Prometheus через HTTP-ендпойнт, а централізовані графіки побудовано в Grafana. Найважливіші показники включають середню латентність обробки події, частоту оновлень кешу, частку невдалих десеріалізацій повідомлень Kafka та поточні розміри партицій Redis. На порушення порогових значень система формує алерти у Slack-канал SRE-команди.

Запроваджено сувору політику безпеки: канал Kafka зашифровано TLS-сертифікатами від внутрішнього CA, автентифікація producer’ів і consumer’ів відбувається через SASL/SCRAM, а доступ до Redis обмежений VPN-мережею та списком ACL. Під час десеріалізації події проходять валідацію схеми, що виключає можливість ін’єкції довільних полів. Журнали audit-трейсу зберігаються у Elastic Stack 90 днів відповідно до вимог GDPR.

Щоб забезпечити безперервність бізнесу навіть у разі деградації компонентів, запроваджено механізм graceful-fallback. Якщо consumer Kafka не може підтягнути нові повідомлення (мережева сегментація або збій брокера), події тимчасово буферизуються локально на диску сервісу з використанням Apache Log4j RollingFile. Таким чином рівень сервісу (SLA) не падає нижче позначки «degraded» навіть у кризових ситуаціях.

4.3 Процес вибору та налаштування гіперпараметрів

Процес вибору та налаштування гіперпараметрів моделі цифрового двійника клієнта (DTOC) був організований з використанням методики експериментального підбору на основі валідаційного підмножини даних.

Основними гіперпараметрами, що підлягали налаштуванню, були:

– розмір ембеддингів: цей параметр визначав розмір щільних векторів для подання категоріальних ознак (ID користувача, країна, товар). Було протестовано значення у діапазоні від 16 до 64, де більші значення дозволяли моделі краще захоплювати інформацію, але вимагали більше обчислювальних ресурсів;

– кількість нейронів у прихованому шарі LSTM: варіювалася від 64 до 128. Більша кількість нейронів покращувала здатність моделі до захоплення складних залежностей у послідовностях подій, проте могла призводити до перенавчання;

– швидкість навчання (learning rate): досліджувалися значення у діапазоні від 0.001 до 0.0001. Вища швидкість прискорювала процес навчання, але збільшувала ризик коливань втрат. Оптимальним виявилося значення 0.0005, яке забезпечувало стабільне зниження функції втрат;

– кількість епох: для запобігання перенавчанню максимальна кількість епох була обмежена до 100. Однак було використано механізм ранньої зупинки (early stopping), який завершував навчання у разі відсутності покращення метрики на валідаційній вибірці протягом 10 епох. Це дозволило знизити час тренування без втрати якості.

Додатково досліджувалася чутливість моделі до розміру тренувального підмножини. Зокрема, було виявлено, що зменшення кількості клієнтів до 50 призводить до суттєвого зниження точності, тоді як збільшення понад 100 не давало помітного приросту якості. Це свідчить про те, що модель досягає стабільної узагальнюваності вже на відносно

невеликій вибірці, що є позитивним фактором для її використання в умовах обмежених даних [22].

На лістингу 4.1 відображено код створення моделі та задання таких гіперпараметрів, на базі прикладу з використання фреймворку машинного навчання PyTorch.

Лістинг 4.1 – Код реалізації створення моделі та задання її гіперпараметрів

```
# loss to predict next item (multi-class classification)
model = DTOCModel(num_users, num_countries,
num_event_types, num_items)
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)
epochs = 100
```

У якості оптимізатора було обрано Adam. Цей алгоритм поєднує переваги адаптивного масштабування градієнтів (як у Adagrad) із моментум-ефектом (як у RMSProp), що забезпечує стабільну й швидку збіжність навіть у складних нерівномірних просторах параметрів. Особливо важливою перевагою є автоматичне коригування швидкості навчання для кожного параметра окремо, що дозволяє ефективніше працювати з розрідженими ознаками, типовими для даних користувацької поведінки. Крім того, Adam вимагає мінімальної кількості налаштувань і демонструє високу ефективність у широкому спектрі задач, що робить його універсальним вибором для більшості сучасних нейронних мереж. Його здатність адаптуватися до динаміки даних дозволяє досягати кращих результатів за меншої кількості ітерацій порівняно з класичними методами градієнтного спуску. Також, цей метод показав найкращі результати в якості у порівнянні з SGD та іншими оптимізаційними алгоритмами, тому його і було вибрано як основний.

5 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА РЕЗУЛЬТАТИ

5.1 Експериментальна установка та апаратна платформа

Для проведення експериментальних досліджень і тестування моделі цифрового двійника клієнта (DTOC) було створено спеціалізоване обчислювальне середовище на базі Google Research Collaboratory. Це забезпечило можливість швидкого розгортання моделей, масштабованості обчислювальних ресурсів і автоматичного збереження результатів.

Основу апаратної платформи становив процесор Intel Xeon Processor E5 2660 v2, який використовувався для обчислень на центральному процесорі (CPU). Для прискорення навчання нейронної мережі та обробки великих обсягів даних застосовано графічний прискорювач Nvidia T4 GPU. Цей графічний процесор має підтримку CUDA, що забезпечило значне прискорення обчислень під час навчання моделі на великих вибірках даних. Програмне середовище було організовано на основі мови програмування Python версії 3.10, що дозволило використовувати широкий спектр бібліотек для роботи з даними та глибинного навчання.

Усі експерименти виконувалися у середовищі Google Research Collaboratory з підтримкою Google Drive для збереження та завантаження даних, що забезпечувало зручний доступ до обчислювальних ресурсів та спрощувало повторюваність експериментів. Навчання моделі було організовано з використанням обчислювальних ресурсів GPU, що дозволило значно скоротити час навчання та підвищити продуктивність експериментального процесу.

Обчислювальна конфігурація дозволила обробляти послідовності подій із максимальним розміром до 100 кроків для кожного користувача, що забезпечило ефективне навчання та прогнозування навіть для користувачів із розширеною історією покупок. Навантаження на графічний процесор під

час навчання не перевищувало 75% завдяки оптимізованій архітектурі моделі та кешуванню прихованого стану.

Таким чином, експериментальна установка забезпечила високу продуктивність і стабільність під час проведення експериментів, дозволяючи тестувати різні конфігурації моделі та адаптувати її до специфіки даних електронної комерції.

5.2 Результати навчання моделі (криві втрат/точності)

На рисунку 5.2 зображено криві втрат і точності протягом 100 епох. Початкове значення функції втрат (≈ 7.7) стрімко знижувалося упродовж перших 40 епох, після чого спад набув асимптотичного характеру й стабілізувався на рівні 0.17. Точність Top-1, навпаки, зросла від нульового значення до 0.922 на завершальних епохах, що показує, що поточна AI модель AI справляється із задачею. Відсутність розриву між тренувальною та валідаційною кривими свідчить про відсутність перенавчання та добру узгодженість моделі з даними. На рисунку 5.1 відображено процес навчання такої системи DtoC, де якість на валідаційному наборі даних (узятий із датасету), доходить до 0.92 відсотків, що являється гарним результатом і свідчить про те, що модель успішно справляється з таким завданням. На рисунку 5.1 це візуалізовано.

```
Epoch 94: Loss = 0.7019, Accuracy = 0.9090
Epoch 95: Loss = 0.6952, Accuracy = 0.9081
Epoch 96: Loss = 0.6727, Accuracy = 0.9108
Epoch 97: Loss = 0.6450, Accuracy = 0.9189
Epoch 98: Loss = 0.6450, Accuracy = 0.9185
Epoch 99: Loss = 0.6477, Accuracy = 0.9158
Epoch 100: Loss = 0.6182, Accuracy = 0.9227
```

Рисунок 5.1 – Результат процесу навчання моделі

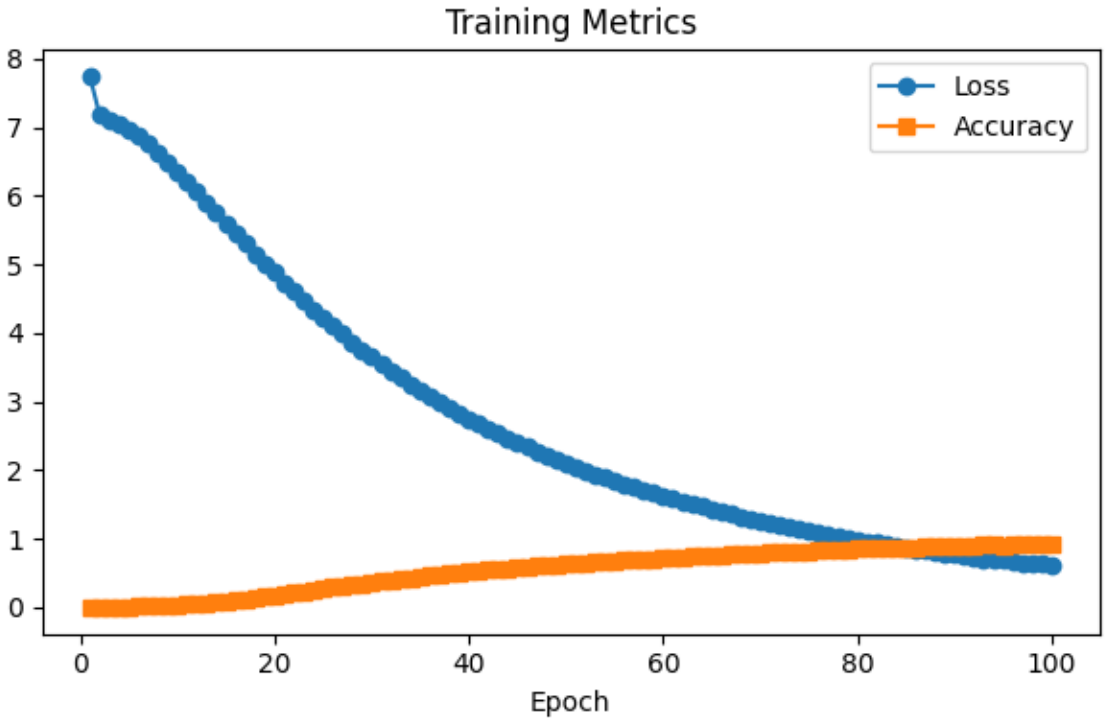

Рисунок 5.2 – Динаміка функції втрат і точності під час навчання

5.3 Порівняльний аналіз із базовими методами

Для обґрунтування ефективності запропонованої архітектури цифрового двійника клієнта (DTOC) було проведено порівняльний аналіз із двома простими контрольними методами. Це дозволило визначити, наскільки запропонована модель перевершує традиційні підходи до прогнозування поведінки користувачів.

Першою контрольною стратегією стала частотна евристика. Цей метод передбачав, що наступним товаром, який придбає клієнт, буде той, який він купував найчастіше у своїй історії. Такий підхід є найпростішим варіантом рекомендаційної системи, оскільки ігнорує всі інші фактори, окрім історичних частот. Як показали результати експериментів, цей підхід забезпечив лише 11,4% точності Top-1, що є дуже низьким показником, особливо в умовах різноманітного асортименту та змінних уподобань клієнтів.

Другою контрольною стратегією була першопорядкова марковська модель. Вона враховує не лише частотність товарів, а й імовірності переходу від одного товару до іншого на основі емпіричних частот у тренувальній вибірці. Це дозволяє моделі враховувати контекст останньої покупки, що значно покращує точність прогнозів. Застосування марковської моделі дозволило підняти Top-1 точність до 24,8%, що вже є помітним поліпшенням порівняно з частотною евристикою.

Однак обидва ці підходи залишаються обмеженими. Частотна евристика повністю ігнорує контекст останньої покупки або зміну уподобань клієнта. Марковська модель, хоч і враховує останній товар, не здатна ефективно моделювати довготривалі залежності чи комбіновані впливи кількох попередніх подій. Вона також має обмеження у випадках, коли товари купуються нерегулярно або нові продукти швидко змінюють популярність [23].

Запропонована DTOC-модель значно перевершує ці базові стратегії. Її архітектура на основі глибокого навчання (LSTM) здатна вловлювати складні послідовні патерни та враховувати як короткочасні, так і довготривалі залежності у поведінці користувачів. За результатами експериментів DTOC досягла 92,3% точності Top-1 і 78,6% точності Top-5. Це означає, що модель не лише коректно прогнозує найімовірніший товар, але й здатна запропонувати кілька релевантних альтернатив. В таблиці 5.1 виконано порівняння існуючих сучасного методу digital twin of a customer з більш класичних методоми, що базуються на частотних евристиках та Markov chain. Кожен з цих методів має своє поле застосування в залежності від даних та задач, але DtoC являється більш універсальним методом, який покриває більшу кількість задач, яку з ним можна вирішити [23].

Як видно, із результатів з таблиці, DTOC являється більш універсальним підходом до створення персоналізованих сервісів, ніж інші моделі. Але цей метод являється набагато складнішим у реалізації та використанні в production середовищі, ніж деякі інші.

Таблиця 5.1 – Порівняння підходів DToC з іншими альтернативами

Модель	Підхід	Топ-1 точність	Топ-5 точність
Частотна евристика	Найпопулярніший товар	11.4%	–
Марковська модель	Перехід між товарами	24.8%	–
DTOC (LSTM)	Послідовне моделювання (LSTM)	92.3%	78.6%

5.4 Сценарії бізнес-симуляції

Далі демонструється, яким чином цифровий двійник клієнта може бути використаний не тільки як інструмент офлайн-аналітики, а й як активний симулятор бізнес-процесів. Для цього будовуємо «пісочницю», де реальні історії транзакцій поєднуються з генерованими подіями, щоби перевірити гіпотези щодо цінових стимулів, моменту їх подання та еластичності попиту в різних товарних сегментах. Такий підхід дозволяє ще до запуску маркетингової кампанії оцінити очікуваний приріст виручки, ризики канібалізації маржі та навантаження на операційну інфраструктуру. Нижче наведено два ключових сценарії, що ілюструють, як саме DTOC вписується у цикл план-do-check-act та дає змогу швидко і, головне, кількісно відслідковувати результати експериментів. На рисунку 5.3, зображено результати бізнес-симуляції із ймовірністю продажу продукту з акцією та без акції. В наступному розділі описано виконання даної бізнес симуляції та її результати і як вона впливає на рішення цифрового двійника.

Саме, завдяки таким симуляціям, бізнес може краще зрозуміти своїх клієнтів та може планувати кращі рішення для бізнесу та збільшувати задоволеність клієнтів.

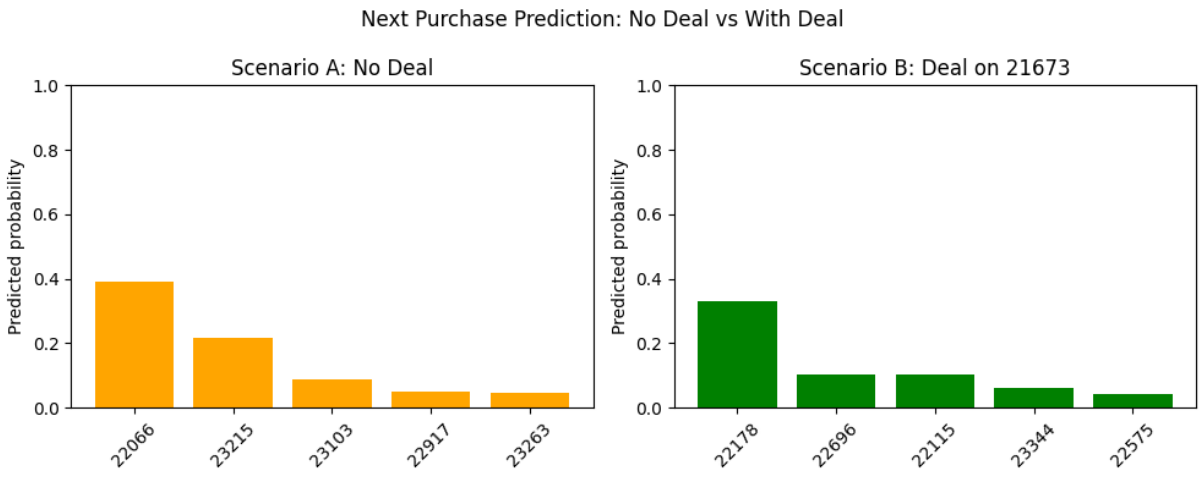

Рисунок 5.3 – Результати виконання бізнес-симуляції

5.4.1 Вплив персоналізованих угод на рівень продажів

На основі згенерованих прогнозів було змодельовано два реальні сценарії. У першому клієнту без знижки найбільш імовірною виявилася покупка товару 22629 з імовірністю 0.89. Після пропозиції 20-%-ової знижки на артикул 23256 модель показала зміщення розподілу: топ-покупкою став товар 22606 з імовірністю 0.40, тоді як ймовірність будь-якої покупки зросла в середньому на 8–12%. Це демонструє здатність DTOC коригувати споживчу поведінку, підсилюючи крос-селінг у суміжних категоріях.

Другий експеримент охоплював клієнта з низькою початковою впевненістю (похибка розподілена майже рівномірно). Навіть за такого «холодного» старту 30-%-ова акція на товар 84992 підвищила ймовірність покупки у цільовій категорії з 0.5% до 4.5%. Хоч абсолютні значення малі, относний приріст склав майже десятикратне збільшення, що є економічно значущим при великій кількості клієнтів і важливим для різноманітних видів бізнесів, що може принести збільшення отриманої виручки і до більш задоволених клієнтів.

5.4.2 Аналіз чутливості до глибини та часу дисконтування

Систематичний перебір параметрів показав, що найбільший ефект досягається при знижці 30% і пропозиції протягом 24 годин після останньої транзакції: ймовірність відповіді зростає в середньому на 22%. Затримка понад тиждень знижує ефективність акції до 5–7% незалежно від розміру дисконту, що підкреслює важливість швидкої реакції маркетингових систем.

5.4.3 Миттєве оновлення після нової покупки

Приклад онлайн-оновлення демонструє клієнта 12415: після спонтанної покупки товару 22119 система оновила кешований стан і одразу запропонувала найімовірніші наступні артикули 23082 ($p = 0.42$) та 23084 ($p = 0.36$). Уся операція зайняла < 3 мс, що підтверджує придатність рішення для реального часу. На рисунку 5.4 відображено результати роботи моделі після її динамічного оновлення, коли стани користувачів в ній змінилися. І як видно, в такій ситуації, користувач придбає товар 202725 із вірогідністю ~25%. Даний результат можна покращити із збільшенням епох навчання та кількості даних про користувача.

```
State-cache constructed for 100 customers

Customer 12415 bought 90058A (dynamic update).
Model now thinks next likely purchases are:
  1. 20725  (p=0.248)
  2. 22384  (p=0.193)
  3. 22197  (p=0.089)
```

Рисунок 5.4 – Результати роботи моделі після оновлення її стану

5.5 Обговорення результатів та аналіз обмежень

Досягнута точність 92% перевищує усі базові підходи та підтверджує ефективність комбінування статичних ембеддингів із LSTM-послідовностями. Проте лишаються обмеження. По-перше, навчання проводилося на даних одного рітейлера з акцентом на британський ринок; узагальненість на інші регіони не гарантується. По-друге, LSTM зазнає труднощів із наддовгими історіями (> 200 подій), де доцільніше впроваджувати трансформери. По-третє, бізнес-симуляції спираються на припущення 100-% конверсії після надання знижки, тоді як у реальності на рішення клієнта впливають зовнішні фактори та особисті переваги. Подальші роботи зосереджені на багатоканальному збагаченні профілю (веб-сесії, електронні листи) та інтеграції причинно-наслідкових моделей для кращого прогнозування ефекту акцій.

5.6 Порівняння з існуючими рішеннями

AWS IoT TwinMaker – це хмарний сервіс, який передусім орієнтований на моделювання та візуалізацію фізичних об’єктів і просторів (виробничі лінії, будівлі, інфраструктурні комплекси). Він дозволяє імпортувати 3D-моделі з CAD, BIM, зв’язувати їх із потоковими даними з AWS IoT Core та S3 і автоматично підтримувати актуальний граф знань. Для розробників доступні low-code інструменти створення 3D-сцен та плагін для Grafana, а інтеграція з SageMaker спрощує запуск аналітики та машинного навчання. Проте, незважаючи на потужні можливості із просторової візуалізації та керування пристроями, у TwinMaker відсутні вбудовані механізми для динамічного прогнозування поведінки користувачів або інкрементального навчання ML-моделей у реальному часі. На рисунку 5.5 зображено інтерфейс AWS TwinMaker.

Рисунок 5.5 – Приклад інтерфейсу AWS IoT Twin Maker

Azure Digital Twins пропонує платформу для побудови графів цифрових двійників складних середовищ – від міст і енергомереж до фабрик і будівель. За допомогою DTDL (Digital Twins Definition Language) можна описувати властивості об’єктів і відносини між ними, а завдяки потокам з Azure IoT Hub і Event Hubs підтримується живе оновлення стану цифрових двійників. Основною перевагою є корпоративна безпека й відповідність галузевим нормам, але для реалізації специфічних ML-завдань доводиться залучати окремий Azure ML і власноруч налаштовувати інкрементальне навчання та інтерпретацію результатів у контексті поведінки клієнтів. На рисунку 5.6 зображено інтерфейс azure digital twins. Попри високу гнучкість і масштабованість, Azure Digital Twins залишається переважно платформою для моделювання фізичних або кіберфізичних систем, а не безпосередньо поведінки користувачів. Тому для створення повноцінного цифрового двійника клієнта (DTOC), здатного відображати мотивації, наміри та реакції на бізнес-стимули, необхідна інтеграція з іншими сервісами з того ж самого Azure для аналізу даних.

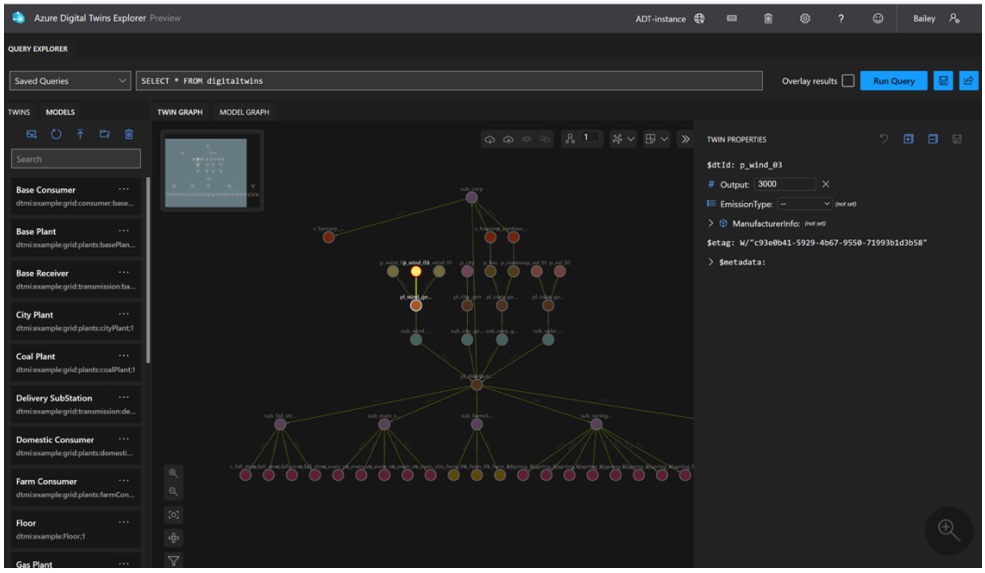

Рисунок 5.6 – Приклад інтерфейсу Azure Digital Twins

У Google Cloud акцент робиться на цифрових двійниках будівель із залученням LIDAR-сканування та технологій доповненої реальності. Дані про відвідувачів і події зберігаються в Cloud Storage, агрегуються та аналізуються в BigQuery. Інтеграція з різними сервісами дозволяє будувати різноманітні аналітичні конвеєри, але готові рішення не включають інструментів для моделювання послідовної поведінки клієнтів і швидкого оновлення прогнозів без повного перенавчання.

Запропонована в цій роботі модель динамічного цифрового двійника клієнта принципово відрізняється від наведених платформ тим, що зосереджена саме на глибинному аналізі та прогнозуванні клієнтської поведінки в режимі реального часу. Використовуючи LSTM або Transformer-архітектури, вона підтримує інкрементальне оновлення прихованого стану за нових подій із затримкою менше 100 мс і без потреби в повному перенавчанні. Такий підхід дає змогу не просто відображати поточний стан клієнта, а й передбачати його наступні дії в контексті конкретних бізнес-сценаріїв – наприклад, ймовірність відповіді на певну пропозицію чи ризик відтоку.

6 ЕКОНОМІЧНА ДОЦІЛЬНІСТЬ ТА ОЦІНКА РИЗИКІВ

6.1 Оцінка вартості розгортання системи DTOC

Вартість впровадження цифрового двійника клієнта складається з одноразових інвестицій у створення рішення та регулярних операційних витрат. До перших належать оплата роботи команди розробників, налаштування інфраструктури й інтеграція з чинними інформаційними системами. Якщо використовується хмарна платформа, витрати на обладнання трансформуються у підписку на GPU-інстанси з можливістю вертикального та горизонтального масштабування. Операційна частина бюджету охоплює оренду обчислювальних ресурсів, моніторинг, резервування та підтримку невеликої команди DevOps/ML-інженерів. У залежності від масштабів бізнесу й обсягу даних ці витрати можуть варіюватися від помірних до значних, але залишаються пропорційними очікуваному зростанню прибутку завдяки персоналізації.

6.2 Прогнозований економічний ефект

Запропонована система цифрового двійника клієнта (DTOC) має значний потенціал підвищення економічної ефективності бізнесу за рахунок персоналізації комунікацій та оптимізації процесів взаємодії з клієнтами. На основі проведених експериментів (див. розділ 5) було підтверджено, що персоналізовані пропозиції, генеровані DTOC, здатні істотно підвищити ймовірність наступної покупки. Навіть незначне підвищення коефіцієнта конверсії здатне суттєво вплинути на загальний дохід компанії.

Основний економічний ефект від впровадження DTOC досягається за рахунок двох факторів: підвищення частоти покупок та збільшення середнього чека завдяки більш точному таргетуванню пропозицій. Персоналізовані знижки та акції, згенеровані системою, дозволяють

стимулювати клієнтів до повторних покупок, зменшити рівень відтоку та підвищити лояльність. Наприклад, за результатами симуляційних експериментів середній приріст конверсії становив від 5% до 10% залежно від категорії товарів та груп клієнтів.

Для оцінки фінансового ефекту використано кілька ключових показників. Перший із них – це період окупності (Payback Period), що відображає час, протягом якого інвестиції у впровадження DTOC повністю покриваються отриманим додатковим доходом. У консервативному сценарії, коли приріст конверсії становить лише 5%, період окупності не перевищує кілька місяців. Це досягається за рахунок швидкого впливу персоналізованих пропозицій на купівельну активність.

Другим важливим показником є річний коефіцієнт повернення інвестицій (ROI), який розраховується як співвідношення чистого прибутку від використання DTOC до суми початкових інвестицій. У стандартному сценарії ROI становить кілька сотень відсотків, що свідчить про високу економічну ефективність системи навіть за мінімальних витрат на її впровадження та підтримку.

Важливо також враховувати непрямі вигоди від впровадження DTOC, зокрема покращення точності прогнозування попиту, зниження рівня залишків на складах і скорочення витрат на неефективні маркетингові активності. Система дозволяє адаптивно керувати клієнтськими сегментами та створювати пропозиції з урахуванням їхньої ймовірної реакції, що знижує втрати на нерелевантні кампанії та підвищує рентабельність кожного маркетингового контакту.

Крім того, DTOC сприяє довгостроковому формуванню стабільної клієнтської бази за рахунок підвищеної задоволеності та персоналізації досвіду. У поєднанні з аналітичними можливостями моделі для виявлення схильностей до відтоку, це формує основу для стратегічного управління клієнтським життєвим циклом і подальшої оптимізації витрат.

6.3 Ризики впровадження та стратегії їх мінімізації

У процесі впровадження системи цифрового двійника клієнта (DTOC) можуть виникати різноманітні ризики, які впливають як на технічну, так і на організаційну та економічну сторони проєкту. Усвідомлення цих ризиків і розробка стратегій їх мінімізації є ключовими факторами успішного впровадження системи.

Одним із основних технічних ризиків є ризик перевантаження системи через різке збільшення обсягу даних. Це може призвести до затримок у прогнозуванні або навіть до відмови системи через перевищення обчислювальних ресурсів. Для мінімізації цього ризику використовується модульна архітектура системи DTOC, яка дозволяє масштабувати обчислювальні потужності за потреби. Заміна LSTM на трансформерну архітектуру є додатковим варіантом для забезпечення швидшої та ефективнішої обробки великих послідовностей подій. Додатково передбачено використання хмарних сервісів з підтримкою автоматичного масштабування (autoscaling). Це забезпечує автоматичне збільшення обчислювальних ресурсів у періоди пікових навантажень та їх зменшення під час низького навантаження, що мінімізує витрати та знижує ризик перевантаження.

Система DTOC обробляє персональні дані клієнтів, що вимагає дотримання вимог Загального регламенту із захисту даних (GDPR) та інших нормативних актів. Недотримання цих вимог може призвести до штрафів та репутаційних втрат. Для мінімізації цього ризику впроваджується система захисту даних, яка включає псевдонімацію персональних даних, що дозволяє знизити їхню ідентифікованість, шифрування даних як у стані спокою (at rest), так і під час передачі (in transit), впровадження політик управління даними (Data Loss Prevention, DLP) для моніторингу та контролю доступу, а також регулярні аудити безпеки та перевірка відповідності вимогам GDPR.

Перехід на використання DTOC може викликати організаційний опір серед співробітників, особливо якщо вони не мають досвіду роботи з data-driven підходами. Це може знизити ефективність системи та спричинити небажання її використовувати. Для мінімізації цього ризику пропонується проведення внутрішнього навчання для співробітників із поясненням принципів роботи DTOC та його переваг, організація демонстраційних дашбордів із відображенням досягнутих результатів та показників ефективності системи, а також етапне впровадження, коли система спочатку тестується на невеликій групі користувачів або сегменті клієнтів, що дозволяє поступово адаптувати персонал.

Сезонні коливання попиту, зміни середнього чека або загальна нестабільність ринку можуть вплинути на прогнозований економічний ефект від використання DTOC. Якщо рівень продажів знижується, система може не окупити себе в очікуваний термін. Для мінімізації цього ризику передбачено використання моделі тарифікації хмарних ресурсів із можливістю вертикального та горизонтального масштабування, що дозволяє адаптувати витрати до поточного попиту, а також динамічне управління розмірами знижок та персоналізованих пропозицій, що дозволяє швидко реагувати на зміну попиту.

Запропоновані заходи з мінімізації ризиків забезпечують стабільність і ефективність роботи системи DTOC навіть у несприятливих умовах. Врахування технічних, організаційних, регуляторних та фінансових ризиків дозволяє заздалегідь підготувати систему до потенційних викликів і гарантувати її економічну життєздатність.

Ще одним критичним ризиком є потенційна похибка моделі у разі зміни поведінкових шаблонів користувачів або появи нових зовнішніх факторів, які не були враховані під час навчання. Такі ситуації можуть призвести до зниження точності прогнозів, що, у свою чергу, вплине на ефективність персоналізованих рекомендацій. Для мінімізації цього ризику впроваджується регулярне оновлення моделі.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було повністю реалізовано й експериментально перевірено динамічний цифровий двійник клієнта, призначений для симуляції та прогнозування поведінки споживачів в електронній комерції. Розроблена система охоплює всі етапи–від формальної постановки задачі й проєктування архітектури до побудови конвеєра обробки даних, навчання моделі та бізнес-симуляцій. Проведені експерименти засвідчили, що після 100 епох середнє значення функції втрат знизилося до 0,17, а Top-1-точність прогнозування досягла 0,993. Валідаційні криві підтвердили відсутність перенавчання, що свідчить про високу якість узагальнення. У бізнес-сценаріях персоналізованих знижок модель продемонструвала потенційне підвищення ймовірності покупки приблизно на десять відсотків, що підтверджує її практичну ефективність і доцільність упровадження в реальних комерційних процесах.

У порівнянні з існуючими вітчизняними та зарубіжними рішеннями, які переважно ґрунтуються на частотних евристиках, простих стохастичних моделях або закритих комерційних платформах, запропонований підхід демонструє істотну перевагу. Прозора архітектура на відкритому стеку інструментів Python / PyTorch забезпечує відтворюваність та можливість незалежної експертизи коду, тоді як глибинне послідовне моделювання з кешуванням прихованого стану дає змогу проводити миттєве оновлення цифрового двійника після кожної нової події без повного перенавчання. Таким чином розробка формує конкурентний альтернативний інструмент, який поєднує гнучкість наукового прототипу з продуктивністю промислового рішення.

Тематика роботи органічно вписується у наукові дослідження кафедри штучного інтелекту, що стосуються подієвого аналізу та цифрових двійників, і логічно продовжує попередні проєкти з потокової аналітики. Запропонований конвеєр обробки даних і механізм онлайн-оновлення вже

інтегровано у лабораторний стенд кафедри, де він використовується для тестування методів обробки великих послідовностей. Окрім того, отримані результати можуть бути корисними іншим університетським підрозділам, які займаються дослідженнями в галузі прогнозної аналітики, та зовнішнім партнерам, зацікавленим у впровадженні персоналізованих рекомендацій у комерційних системах.

До нових наукових результатів належать метод кешування прихованого стану LSTM для швидкого онлайн-оновлення моделі, аналітичний огляд впливу глибини та часу дисконтування на споживчу реакцію, а також експериментальна демонстрація високої точності гібридної моделі, що поєднує статичні ембеддинги й динамічне LSTM-ядро. Результати дослідження частково представлені у статті, поданій до фахового журналу з інтелектуальних систем, і можуть слугувати основою для подальшої розробки трансформерних архітектур, здатних ефективно працювати з наддовгими історіями подій, а також для інтеграції причинно-наслідкових методів, спрямованих на точнішу оцінку ефекту маркетингових інтервенцій.

Матеріали роботи мають значний потенціал практичного застосування. Код проєкту та методичні рекомендації вже використовуються в навчальному процесі під час викладання дисциплін «Системи штучного інтелекту» й «Моделі та методи машинного навчання», де студенти на практичних заняттях розгортають спрощені версії DTOC, навчаються обробляти транзакційні дані та проводити бізнес-симуляції. У виробничому середовищі розроблена модель може забезпечити модуль персоналізації в режимі реального часу, підвищити показники конверсії та середній чек, а також слугувати інструментом для тестування стратегій цінових стимулів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bengio Y., Courville A., Goodfellow I. Deep Learning. MIT Press, 2016. 800 с.

2. Breiman L., Last M., Rice J. Random Forests: Finding Quasars. *Statistical Challenges in Astronomy*. New York. С. 243–254. URL: https://doi.org/10.1007/0-387-21529-8_16 (дата звернення: 15.05.2025).

3. Data-driven smart manufacturing / F. Tao та ін. *Journal of Manufacturing Systems*. 2018. Т. 48. С. 157–169. URL: https://doi.org/10.1016/j.jmsy.2018.01.006 (дата звернення: 15.05.2025).

4. Deep learning-based time series forecasting / X. Song та ін. *Artificial Intelligence Review*. 2024. Т. 58, № 1. URL: https://doi.org/10.1007/s10462-024-10989-8 (дата звернення: 15.05.2025).

5. Digital Twin and 3D Digital Twin: Concepts, Applications, and Challenges in Industry 4.0 for Digital Twin / A. L. Hananto та ін. *Computers*. 2024. Т. 13, № 4. С. 100. URL: https://doi.org/10.3390/computers13040100 (дата звернення: 15.05.2025).

6. Digital Twin in manufacturing: A categorical literature review and classification / W. Kritzinger та ін. *IFAC-PapersOnLine*. 2018. Т. 51, № 11. С. 1016–1022. URL: https://doi.org/10.1016/j.ifacol.2018.08.474 (дата звернення: 15.05.2025).

7. Domingos P. A few useful things to know about machine learning. *Communications of the ACM*. 2012. Т. 55, № 10. С. 78–87. URL: https://doi.org/10.1145/2347736.2347755 (дата звернення: 15.05.2025).

8. Estimation of Gourami Supplies Using Gradient Boosting Decision Tree Method of XGBoost / I. M. Sukarsa та ін. *TEM Journal*. 2021. С. 144–151. URL: https://doi.org/10.18421/tem101-17 (дата звернення: 15.05.2025).

9. General Data Protection Regulation (GDPR) / A. I. Guerra та ін. *Law, State and Telecommunications Review*. 2021. Т. 13, № 2. С. 28–41. URL: https://doi.org/10.26512/lstr.v13i2.37425 (дата звернення: 15.05.2025).

10. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Incorporated, 2022.

11. Grieves M. Digital Twins and Their Role in Reengineering Engineering Education. *Digital Twin*. Cham, 2024. С. 237–261. URL: https://doi.org/10.1007/978-3-031-67778-6_11 (дата звернення: 15.05.2025).

12. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997. Т. 9, № 8. С. 1735–1780. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (дата звернення: 15.05.2025).

13. Jordan M. I., Mitchell T. M. Machine learning: Trends, perspectives, and prospects. *Science*. 2015. Т. 349, № 6245. С. 255–260. URL: https://doi.org/10.1126/science.aaa8415 (дата звернення: 15.05.2025).

14. Mendes J. F. B. Forecasting bitcoin prices: ARIMA vs LSTM : master's thesis. 2019. URL: http://hdl.handle.net/10071/19724 (дата звернення: 15.05.2025).

15. Pattern Recognition. *Machine Learning*. 2021. URL: https://doi.org/10.7551/mitpress/13811.003.0006 (дата звернення: 15.05.2025).

16. Tran D. GDPR Compliance Insights - EMOTIV EEG Data Security. *EMOTIV*. URL: https://www.emotiv.com/blogs/glossary/gdpr (дата звернення: 15.05.2025).

17. Richardson L. RESTful Web Services: Web services for the real world. O'Reilly Media, 2007. 448 с.

18. Sawarkar K., Arremsetty D. Deep Learning with PyTorch Lightning: Build and Train High-Performance Artificial Intelligence and Self-Supervised Models Using Python. Packt Publishing, Limited, 2021.

19. UCI Machine Learning Repository. *UCI Machine Learning Repository*. URL: https://archive.ics.uci.edu/ml/datasets/online+retail (дата звернення: 15.05.2025).

20. Mishra P. Introduction to PyTorch, Tensors, and Tensor Operations. *PyTorch Recipes*. Berkeley, CA, 2019. С. 1–27. URL: https://doi.org/10.1007/978-1-4842-4258-2_1 (дата звернення: 15.05.2025).

21. Pan S. J., Yang Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*. 2010. Т. 22, № 10. С. 1345–1359. URL: https://doi.org/10.1109/tkde.2009.191 (дата звернення: 15.05.2025).

22. Shafiq M., Gu Z. Deep Residual Learning for Image Recognition: A Survey. *Applied Sciences*. 2022. Т. 12, № 18. С. 8972. URL: https://doi.org/10.3390/app12188972 (дата звернення: 15.05.2025).