

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
Дослідження методів інтеграції машинного навчання в інформаційні системи
обробки (маршрутизації) платіжних транзакцій у фінтех компаніях
(тема)

Виконав:

здобувач 2 року навчання,
групи ІУСТМ-24-1

Сергій МУХА

(Власне ім'я, ПРІЗВИЩЕ)

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

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

Освітня програма Інформаційні управляючі системи та технології
(повна назва освітньої програми)

Керівник доц. каф. ІУС Олена МІХНОВА
(посада, власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту

Завідувач кафедри ІУС



(підпис)

Костянтин ПЕТРОВ

(Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
 Кафедра _____ Інформаційних управляючих систем _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 122 Комп'ютерні науки _____
 (код і повна назва)
 Тип програми _____ освітньо-професійна _____
 (освітньо-професійна або освітньо-наукова)
 Освітня програма _____ Інформаційні управляючі системи та технології _____
 (повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____  _____

(підпис)

“24” листопада 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Мусі Сергію Івановичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи _____ Дослідження методів інтеграції машинного навчання в інформаційні системи обробки (маршрутизації) платіжних транзакцій у фінтех компаніях _____

затверджена наказом університету від “24” листопада 2025 р. № 1055Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії “18” грудня 2025 р.

3. Вихідні дані до роботи _____ Науково-технічна література, публікації та інтернет-ресурси з тематики кваліфікаційної роботи; матеріали передатестаційної практики; синтетичні дані платіжних транзакцій; документація платіжних систем (Corefy, Stripe, Adyen); технічна документація ML-фреймворків (LightGBM, XGBoost); стандарти безпеки (PCI DSS, GDPR). _____

4. Перелік питань, що потрібно опрацювати в роботі _____ Аналіз предметної області та архітектури платіжних систем у фінтех-компаніях; аналіз характеристик платіжних даних та їх структури для машинного навчання; дослідження існуючих методів інтеграції ML у маршрутизацію платежів; розробка гібридної архітектури системи, що поєднує статичні бізнес-правила з ML-моделями; формалізація задачі оптимізації маршруту та розробка алгоритму вибору оптимального шляху; експериментальна валідація методу на синтетичних даних; розробка рекомендацій для впровадження у production. _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	24.11.2025-25.11.2025	Виконано
2	Аналіз завдання, підбір літератури	25.11.2025-26.11.2025	Виконано
3	Аналіз літератури з досліджуваної проблеми	26.11.2025-27.11.2025	Виконано
4	Аналіз технічних засобів	27.11.2025-28.11.2025	Виконано
5	Розробка методу	28.11.2025-02.12.2025	Виконано
6	Програмна реалізація	03.12.2025-08.12.2025	Виконано
7	Оформлення пояснювальної записки	08.12.2025-10.12.2025	Виконано
8	Підготовка презентації та доповіді	10.12.2025-11.12.2025	Виконано
9	Перевірка на плагіат	11.12.2025-11.12.2025	Виконано
10	Захист кваліфікаційної роботи	18.12.2025	

Дата видачі завдання 24 листопада 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. каф. ІУС Олена МІХНОВА
(підпис) (посада, власне ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 84 с., 6 рис., 4 табл., 1 дод., 18 джерел, 14 формул.

ГІБРИДНІ СИСТЕМИ, ІНТЕГРАЦІЯ, МАШИННЕ НАВЧАННЯ, ОПТИМІЗАЦІЯ, ПЛАТІЖНА МАРШРУТИЗАЦІЯ, ФІНТЕХ, LIGHTGBM.

Об'єктом дослідження є процес оптимізації платіжної маршрутизації в системах обробки платіжних транзакцій фінтех-компаній.

Мета роботи – розробка та обґрунтування методів інтеграції машинного навчання в системи платіжної маршрутизації для підвищення успішності обробки платіжних операцій.

У роботі проведено комплексний аналіз архітектури платіжної системи Corefy, дослідження методів машинного навчання щодо придатності для маршрутизації платежів, розроблена математична модель задачі оптимізації маршруту та архітектура гібридної системи з LightGBM як основним алгоритмом.

Запропонована гібридна система поєднує статичні бізнес-правила з динамічною адаптацією на основі ML-моделей, забезпечуючи вибір оптимального платіжного шлюзу в реальному часі. Здійснено експериментальну оцінку на синтетичному наборі даних з 1 млн транзакцій, що продемонструвала ROC-AUC 87,3% для LightGBM при латентності 0,39 мс.

Практичне значення роботи полягає у можливості впровадження розробленого методу в операційні платіжні системи фінтех-компаній для підвищення успішності обробки платежів на 2-5 відсоткових пункти та автоматизації вибору маршруту без ручного втручання аналітиків.

ABSTRACT

Master's thesis: 84 pages, 6 figures, 4 tables, 1 appendices, 18 sources, 14 formulas

FINTECH, HYBRID SYSTEMS, INTEGRATION, LIGHTGBM, MACHINE LEARNING, OPTIMIZATION, PAYMENT ROUTING.

The object of research is the process of optimizing payment routing in payment transaction processing systems of fintech companies.

The aim of the work is to develop and substantiate methods for integrating machine learning into payment routing systems to improve the success rate of payment processing operations.

The work presents a comprehensive analysis of the Corefy payment system architecture, research of machine learning methods for payment routing suitability, a mathematical model of route optimization, and the architecture of a hybrid system with LightGBM as the primary algorithm.

The proposed hybrid system combines static business rules with dynamic adaptation based on ML models, ensuring optimal payment gateway selection in real-time. Experimental evaluation was conducted on a synthetic dataset of 1 million transactions, demonstrating ROC-AUC of 87.3% for LightGBM with a latency of 0.39 ms.

The practical significance of the work lies in the possibility of implementing the developed approach in operational payment systems of fintech companies to increase payment success rate by 2-5 percentage points and automate route selection without manual analyst intervention.

ЗМІСТ

Скорочення та умовні позначки	8
Вступ	10
1. Аналіз предметної області та постановка задач дослідження	12
1.1 Аналіз предметної області фінтех-систем обробки платежів	12
1.1.1 Поняття та особливості платіжних систем у фінтех-секторі	12
1.1.2 Роль маршрутизації у забезпеченні успішності транзакцій	14
1.2 Опис організаційної структури об'єкта автоматизації	15
1.3 Життєвий цикл платіжної транзакції на об'єкті автоматизації	17
1.4 Маршрутизація як центральна функція системи	18
1.5 Виявлені потреби та можливості покращення маршрутизації	20
1.6 Структура даних для машинного навчання при маршрутизації	21
1.6.1 Процес накопичення та обробки даних	22
1.6.2 Обсяг та характеристики доступних даних у Corefy	23
1.7 Інтеграція ML в існуючу архітектуру маршрутизації	25
1.8 Постановка задач дослідження	25
2. Аналіз існуючих методів вирішення проблеми маршрутизації	27
2.1 Статичні правила маршрутизації	27
2.2 Логістична регресія	28
2.3 Random Forest	29
2.4 Градієнтний бустинг: XGBoost та LightGBM	30
2.5 Нейронні мережі	31
2.6 Порівняльний аналіз методів	32
2.7 Висновок по другому розділу	33
3. Розробка методу інтеграції машинного навчання в систему об'єкта автоматизації	35
3.1 Обґрунтування комбінованого підходу до оптимізації маршрутизації	35
3.2 Архітектура гібридної системи маршрутизації	35

3.3	Формалізація задачі оптимізації маршруту	38
3.4	Алгоритм вибору оптимального маршруту	39
3.5	Пояснення етапів алгоритму.....	43
3.6	Механізми адаптації в режимі реального часу	45
3.7	Інтеграція з архітектурою Corefy	46
3.8	Вибір архітектури ML-моделі	47
3.9	Висновок по третьому розділу	48
4.	Експериментальні дослідження та оцінка ефективності	49
4.1	Опис експериментального середовища	49
4.1.1	Датасет: структура, обсяг, підготовка даних	49
4.1.2	Baseline: поточна rule-based система Corefy	53
4.1.3	Метрики оцінки.....	53
4.1.4	Технічне середовище.....	54
4.2	Результати експериментів.....	55
4.2.1	Порівняння ML-моделей (Random Forest, та LightGBM).....	56
4.2.2	Аналіз впливу параметрів та SHAP-інтерпретація	58
4.2.3	A/B тестування: ML vs Rule-based	59
4.3	Оцінка ефективності для IT-проєкту	61
4.4	Рекомендації щодо впровадження у production.....	62
4.5	Висновок по четвертому розділу.....	62
	Висновки	66
	Перелік джерел посилання.....	68
	Додаток А Графічний матеріал кваліфікаційної роботи	71

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІС – інформаційна система

ІТ – інформаційні технології

Фінтех – фінансові технології

3DS – 3D Secure

API – Application Programming Interface

AVS – Address Verification Service

BIN – Bank Identification Number

CI/CD – Continuous Integration/Continuous Delivery or Deployment

CVV – Card Verification Value

EFB – Exclusive Feature Bundling

GDPR – General Data Protection Regulation

GOSS – Gradient-based One-Side Sampling

IPN – Instant Payment Notification

Leaky ReLU – Leaky Rectified Linear Unit

LightGBM – Light Gradient Boosting Machine

LR – Logistic Regression

MCC – Merchant Category Code

ML – Machine Learning

NN – Neural Networks

PCI DSS – Payment Card Industry Data Security Standard

PR-AUC – Precision-Recall Area Under the Curve

ReLU – Rectified Linear Unit

RF – Random Forest

ROC-AUC – Receiver Operating Characteristic Area Under the Curve

ROI – Return on Investment

SDK – Software Development Kit

SHAP – SHapley Additive exPlanations

SR – Success Rate

TPS – Transactions per Second

XGBoost – Extreme Gradient Boosting

ВСТУП

Платіжна маршрутизація є критичним компонентом сучасних фінтех-систем, що визначає шлях проходження транзакції від клієнта до банку та безпосередньо впливає на успішність платежу. Фінансові технології (фінтех) стосуються застосування інноваційних технологій до продуктів та послуг у фінансовій галузі [1]. Глобальна вартість цифрових платежів зростає з 1,7 трильйонів доларів у 2014 році до 18,7 трильйонів у 2024 році; за прогнозами, до 2030 року показник перевищить 33,5 трильйонів доларів. Проте ці цифри приховують серйозну проблему: більшість платіжних систем досі працює за принципом, розробленим на ранніх етапах цифровізації – вибір платіжного шлюзу здійснюється за статичними правилами, що не адаптуються до змін умов обробки платежів у реальному часі.

Практичні обмеження, з якими стикаються платіжні системи, пов'язані з кількістю інтегрованих шлюзів, тривалістю їхньої інтеграції та неоптимальністю алгоритмів вибору. Кожна додаткова інтеграція вимагає значних часових витрат, а правила маршрутизації часто ґрунтуються на простих атрибутах – тип карти, географічна зона, сума платежу. Такі підходи не враховують динамічність реальних умов: показники успішності шлюзів змінюються залежно від регіону, часу доби, типу платежу й поточного навантаження. Дослідження показують, що коефіцієнти схвалення платежів значно варіюються залежно від каналу обробки, що вказує на неоптимальність розподілу транзакцій між шлюзами. Результатом є невдалі платежі, які можна було б уникнути, якби система систематично аналізувала минулі дані й обирала оптимальний маршрут. Для платіжної системи, що обробляє мільярди транзакцій щорічно, це означає значні економічні втрати.

Наявні підходи до управління маршрутизацією характеризуються нижчим рівнем автоматизації й адаптивності, ніж це необхідно для сучасних вимог. Управління маршрутизацією часто здійснюється вручну, командами, що

розслідують причини невдач та постійно коригують правила. Крім того, операційні витрати залишаються неоптимальними через надлишкові трудові ресурси та відсутність швидких механізмів адаптації до змін у роботі платіжних шлюзів. Методологічна розрив полягає в тому, що більшість сучасних підходів не враховують можливість автоматичного аналізу великих обсягів даних трансакцій задля оптимізації вибору шлюзу в реальному часі.

Необхідно розробити формалізований підхід до оптимізації маршрутизації, який би враховував динамічність умов обробки платежів, обмеження на часові та обчислювальні ресурси, а також можливість інтеграції з сучасними методами машинного навчання. Українські фінтех-компанії активно розвиваються, але їхні інформаційні системи маршрутизації часто залишаються на рівні технічних рішень попередніх років. Це створює можливість для впровадження інноваційних рішень на основі машинного навчання.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області фінтех-систем обробки платежів

1.1.1 Поняття та особливості платіжних систем у фінтех-секторі

Платіжна система у фінтехі включає програмні компоненти, API (Application Programming Interface), інтегровані платіжні шлюзи, процесингові модулі і сховища даних, які реалізують повний життєвий цикл безготівкової транзакції: від створення платіжного запиту на стороні клієнта до остаточного підтвердження чи відхилення з боку банківської інфраструктури. У типовій архітектурі можна виділити принаймні такі підсистеми: фронтний платіжний віджет або SDK (Software Development Kit), процесингове ядро, модуль маршрутизації на платіжні шлюзи, модулі боротьби з шахрайством, білінгу та звітності, а також інтеграційний шар з банками, агрегаторами й альтернативними платіжними методами. Для великих платформ характерна інтеграція одразу з кількома провайдерами (“payment orchestrator”), що дозволяє розподіляти трафік між десятками шлюзів і платіжних інструментів, зберігаючи єдиний інтерфейс для мерчанта [2].

Фінтех-платіжні системи відрізняються від банківського процесингу орієнтацією на високі пікові навантаження та можливістю швидкого експериментування з логікою маршрутизації.. Саме на рівні фінтех-платформ сьогодні впроваджуються машинне навчання для прийняття рішень у реальному часі, адаптивні алгоритми маршрутизації та контекстні моделі оцінки ймовірності успіху транзакції на різних шлюзах.

Сучасні платіжні системи характеризуються багатоканальністю: одна й та сама платформа має підтримувати карткові операції, банківські перекази, локальні платіжні схеми, гаманці, миттєві платежі й інші інструменти, причому з різними вимогами до авторизації, відкатів, повернень та чарджбеків. При цьому платформа повинна одночасно забезпечувати відповідність суворим стандартам

безпеки, таких як PCI DSS (Payment Card Industry Data Security Standard), реалізація шифрування та токенизації даних, обмежувати доступ до платіжних даних і не створювати надмірних затримок у ланцюжку обробки. Додатковою складністю є нерівномірний розподіл навантаження: у пікові періоди (розпродажі, матчі, ігрові івенти) потік транзакцій може зростати у 5–20 разів, що вимагає динамічного масштабування сервісів та стійкості до відмов окремих шлюзів [3].

Показники успішності та латентності платіжних шляхів мають стохастичний характер. Дослідження показують, що ймовірність успіху сильно залежить від часу доби, типу інструмента, конкретного шлюзу і навіть короткострокових збоїв на стороні банків або мереж. Це робить неефективними суто статичні схеми маршрутизації “одна країна – один шлюз” і створює запит на алгоритмічні системи, які постійно перевчаються на основі відповідей шлюзів, сигналів про затримки та інформації про збої. Унаслідок цього в сучасних фінтех-системах з’являється окремий клас компонентів – decision engines, які поєднують правила бізнесу, ML моделі, багаторукі бандити й механізми зворотного зв’язку для оптимізації маршрутизації, рівня відмов та економічних показників [4].

Фінтех-системи обробки платежів мають відповідати регуляторним вимогам і водночас забезпечувати гнучкість. Поняття платіжної системи в цьому контексті виходить далеко за межі “платіжного шлюзу” і включає оркестрацію множини провайдерів, методів та бізнес-правил у єдиному інформаційному середовищі. Надалі, при побудові моделі маршрутизації на основі машинного навчання, доцільно спиратися саме на ці особливості: багатоканальність, високі вимоги до латентності, несталий характер показників успішності та наявність багатих потоків історичних транзакційних даних для навчання й адаптації моделей [5].

1.1.2 Роль маршрутизації у забезпеченні успішності транзакцій

Маршрутизація є ключовим механізмом забезпечення високої частки успішно авторизованих транзакцій у платіжних системах, де задіяно багато провайдерів, оскільки саме вона визначає канал проведення операції за умов наявності декількох еквайрів та платіжних провайдерів.

Інтелектуальна маршрутизація дає змогу спрямовувати транзакції до провайдерів із вищими показниками авторизації з урахуванням історичних значень success/approval rate, локальної присутності та поточного стану інфраструктури (латентність, доступність). У практиці великих онлайн-мерчантів це пов'язується із приростом частки успішних платежів на кілька відсоткових пунктів і зменшенням частки технічних відмов.

Використання механізмів розумної маршрутизації дає можливість автоматично переключати транзакцію на альтернативні маршрути в разі збоїв, перевищення тайм-аутів або деградації показників конкретного провайдера, що прямо впливає на стабільність успішності оброблення [6].

Сучасні платіжні платформи реалізують маршрутизацію з урахуванням широкого спектра параметрів транзакції (BIN та тип платіжного інструмента, країна емітента й акцептанта, валюта, сума, спосіб аутентифікації, часові характеристики, профіль ризику), що дає змогу підбирати провайдера з оптимальними локальними коефіцієнтами схвалення для даного сегмента операцій.

Зростання успішності оброблення платежів безпосередньо корелює з підвищенням конверсії на етапі оплати та скороченням кількості покинутих кошиків, що особливо суттєво для електронної комерції.

Оптимізація маршрутів дозволяє поєднати мету максимізації success rate з мінімізацією транзакційних витрат, оскільки вибір провайдера може ґрунтуватися одночасно на очікуваній імовірності схвалення та структурі комісій [7].

1.2 Опис організаційної структури об'єкта автоматизації

Компанія Corefy організована як матрична структура з трьома основними напрямками, що звітують генеральному директору. Така структура дозволяє балансувати між розвитком продукту, технічною реалізацією та комерційною операцією, забезпечуючи при цьому ефективне управління платежами та маршрутизацією. На рисунку 1.1 представлена організаційна структура компанії.

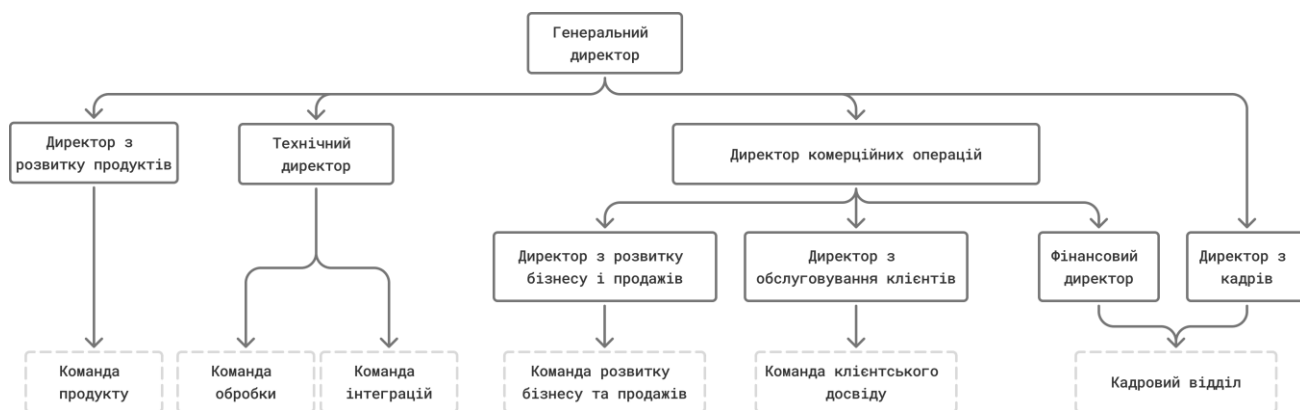


Рисунок 1.1 – Організаційна структура Corefy

Генеральний директор координує три основні функціональні напрямки.

Директор з розвитку продукту – відповідає за стратегію продукту, визначення функціональності та встановлення пріоритетів розробки. Цей напрямок включає Команду продукту, яка займається аналізом вимог мерчантів та визначенням технічних специфікацій.

Технічний директор – управляє всіма аспектами технічної реалізації та операційної надійності системи. Цей напрямок розділяється на кілька команд, кожна з яких має своїх спеціалістів.

Директор комерційних операцій – управляє взаємовідносинами з мерчантами, комерційною стратегією та розширенням ринку. Цей напрямок

включає команди, що займаються обслуговуванням клієнтів та розвитком бізнесу.

Крім того, до структури входять Кадровий відділ (управління персоналом) та Фінансовий директор (управління фінансами).

Архітектура платформи Corefy складається з кількох ключових шарів-компонентів, які взаємодіють для забезпечення обробки платіжних транзакцій та маршрутизації. Це сервісно-орієнтована архітектура з виділенням клієнтських застосунків, ядра та розширень.

Client services: застосунки мерчанта (Checkout App, Merchant Portal, Organization Dashboard), які через публічні та whitelabel-домени звертаються до facade/cashier API.

Core services: центральні сервіси оплати – payment gateway, provider hub, providers gate, payout gateway, IPN handler, що маршрутизують транзакції до зовнішніх платіжних провайдерів.

Common services: ratex, firewall, vault, ledger, які надають спільні функції курсів валют, безпеки, зберігання секретів та бухгалтерського обліку.

Extensions: додаткові модулі (batch payout, subscriptions, reports, інтеграції з месенджерами), що працюють поверх ядра та використовують ті ж API.

Уся взаємодія побудована як набір внутрішніх та зовнішніх API, де facade/cashier приймає запити клієнтів, а далі транзакції проходять через ядро до платіжних провайдерів, використовують спільні сервіси та можуть доповнюватися розширеннями. Рисунок 1.2 ілюструє архітектуру компонентів та сервісів Corefy.

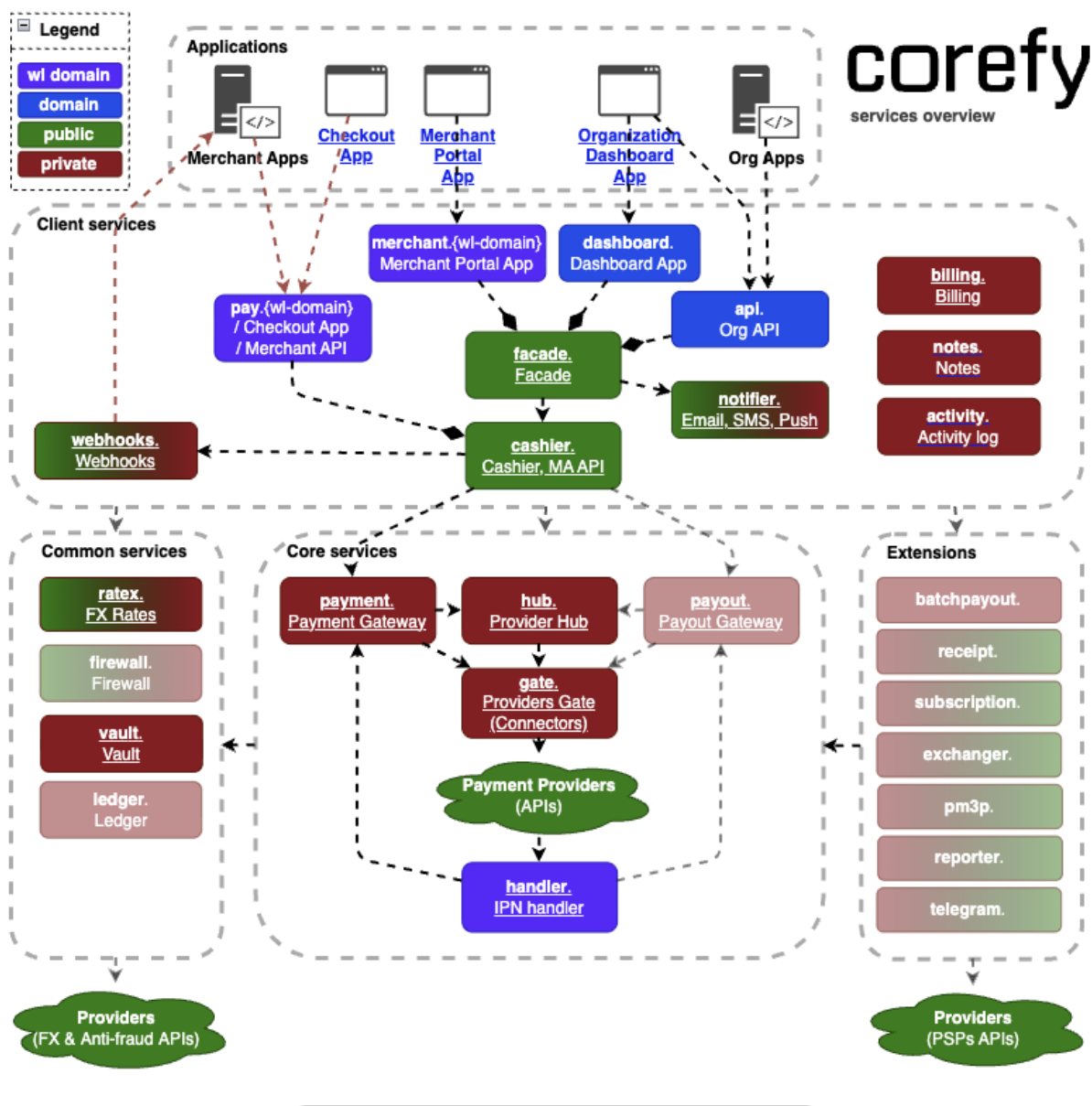


Рисунок 1.2 – Архітектура компонентів та сервісів Corefy

1.3 Життєвий цикл платіжної транзакції на об'єкті автоматизації

Corefy є платформою управління платіжними маршрутами, яка забезпечує обробку платіжних транзакцій через інтеграцію з численними платіжними шлюзами та провайдерами. Архітектура системи побудована на послідовній обробці трьох основних платіжних сутностей: Платіжний Запит (Payment Request), Підтвердження Платежу (Payment Commit) та Платіжна Транзакція

(Payment Transaction). Кожна сутність представляє окремий етап життєвого циклу платежу та накопичує дані про параметри та результати обробки.

Платіжний Запит є найвищорівневою сутністю, яка створюється мерчантом через систему та відображає платіж у системі мерчанта. Платіжний Запит містить основні атрибути платежу: суму та валюту, посилання на операції мерчанта через їхні ідентифікатори, а також опціонально дані про платника та обрані сервіси. Основна задача полягає в інкапсуляції складного процесу проведення платежу та забезпеченні зв'язку з системою мерчанта.

Підтвердження Платежу є операцією, яка відображає оплату, виконану користувачем, та містить інформацію про обраний сервіс та його параметри. На цьому етапі встановлюється зв'язок з рахунком мерчанта та валютним рахунком, вибирається сервіс платежу та встановлюються комерційні умови. Підтвердження Платежу відповідає за облік коштів через систему реєстрації та управління комерційними умовами платежу.

Платіжна Транзакція є сутністю, яка відображає платіж на стороні провайдера. Основна задача полягає у контролі процесу оплати, включаючи послідовність платіжних операцій провайдера, переходи за статусами та обробку сум. Платіжна Транзакція містить ідентифікатор платежу у провайдера, інформацію про успішність обробки та деталі про взаємодію з провайдером.

1.4 Маршрутизація як центральна функція системи

Маршрутизація платіжних транзакцій у Corefy реалізується на етапі Підтвердження Платежу через механізм Маршрутної Схеми. Маршрутна Схема є набором налаштувань, які визначають способи проведення платежу, включаючи комерційні умови та послідовність обробки. Основна задача полягає у забезпеченні гнучкої маршрутизації платежів з можливістю резервного переходу на альтернативні шляхи, динамічного управління параметрами

платежу та встановлення комісій. На рисунку 1.3 представлено контекстну діаграму процесу маршрутизації.



Рисунок 1.3 – Контекстна діаграма процесу маршрутизації

Маршрутна Схема визначає проведення платежу за допомогою кількох компонентів. Правила маршрутизації встановлюють вибір шлюзу для конкретного типу платежу та забезпечують переведення на резервні шлюзи при недоступності основного. Правила конвертації встановлюють способи обміну валют та розрахунку додаткових комерційних надбавок. Конфігурація полів управляє обов'язковістю внесення даних та способами їх заповнення. Механізм версій контролює зміни в налаштуваннях та дозволяє швидко повернутися до попередніх версій.

На практиці мерчанти задають правила маршрутизації через веб-інтерфейс платформи або через конфігураційні файли. Типовий приклад правила: якщо сума платежу більша за 1000 доларів та валюта долари США, то обрати основний шлюз зі спеціальною тарифікацією, а резервні шлюзи встановити як альтернативу. Стратегія маршрутизації може бути однією з таких: фіксована

(один шлюз для всіх), розподілена (розподіл платежів між шлюзами за вагами), оптимальна (з врахуванням поточних метрик), балансована (рівномірний розподіл), за пріоритетом, послідовна (резервні варіанти при невдачі), припинення (заборона обробки) [8]. На рисунку 1.4 представлена декомпозиція процесу маршрутизації.

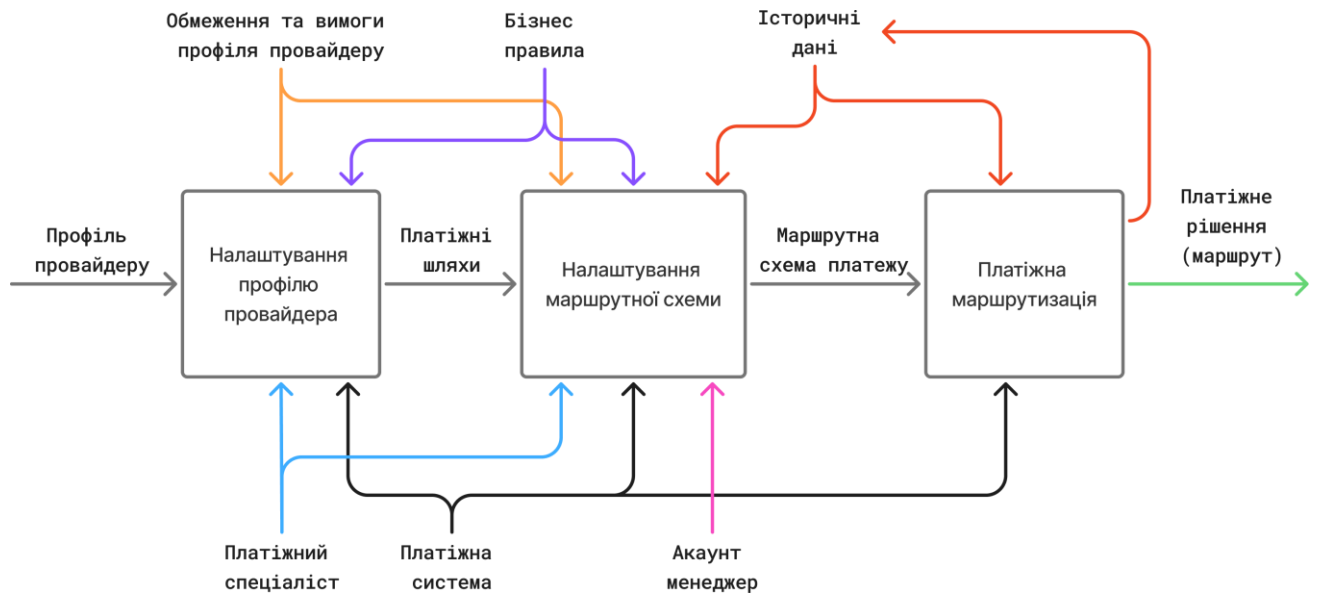


Рисунок 1.4 – Декомпозиція процесу маршрутизації

1.5 Виявлені потреби та можливості покращення маршрутизації

На основі аналізу організаційної структури та архітектури Corefy виявлено кілька критичних потреб у покращенні маршрутизації. Перша потреба полягає у підвищенні успішності обробки платежів. Поточна система залежить від експертної конфігурації та не адаптується до змін умов, географічних різниць та типів платежів. Потрібна система, здатна вивчити складні залежності та досягти кращих результатів.

Друга потреба полягає у мінімізації комісійних видатків при збереженні успішності. Мерчанти готові йти на компроміс між гарантією успіху та

вартістю. Потрібна система, що балансує ці фактори відповідно до потреб мерчанта.

Третя потреба полягає у динамічній адаптації до змін ринку. Шлюзи змінюють комісійні ставки, їх доступність змінюється. Потрібна система, що безперервно адаптується без ручного втручання.

Четверта потреба полягає у вдосконаленні виявлення шахрайства. Поточна система використовує неадаптивні правила. Потрібна система, здатна виявляти складні залежності та нові типи аномалій [9].

П'ята потреба полягає у зменшенні навантаження на команду підтримки. Потрібна система, що надає автоматичні рекомендації та здатна до саморегулювання без ручного втручання.

Шоста потреба полягає у підвищенні конкурентоспроможності платформи. Конкуренти впроваджують подібні технології. Впровадження машинного навчання позиціонує Corefy як інноваційну платформу.

1.6 Структура даних для машинного навчання при маршрутизації

Для впровадження машинного навчання у систему маршрутизації необхідно визначити набір характеристик, які дозволяють прогнозувати результати обробки платежу. На відміну від поточної системи, яка працює з явними правилами, система машинного навчання накопичує та аналізує історичні дані про результати маршрутизації для вивчення оптимальних рішень

Характеристики платежу включають параметри з Платіжного Запиту та Підтвердження Платежу: сума платежу, валюта, тип сервісу платежу, ідентифікатор платника, ідентифікатор мерчанта, тип платіжної карти, країна видання карти, країна мерчанта, країна платника.

Характеристики часу включають час платежу, час доби, день тижня, вихідний день чи ні, місяць року. Ці характеристики враховують часові

закономірності у успішності платежів.

Характеристики шлюзу включають поточний стан та історичну роботу: ідентифікатор шлюзу, доступність у відсотках за останню годину, успішність за останню годину, успішність за останній день, середній час відповіді, комісійна ставка. Ці характеристики змінюються у реальному часі на основі поточних метрик.

Характеристики платника включають загальну вартість для компанії, кількість попередніх операцій, його особисту успішність, час з останньої операції, середню суму його платежів. Ці характеристики відображають поведінку та надійність платника.

Ознаки ризику включають загальну оцінку ризику, перевищення нормальної швидкості операцій, географічну аномалію (країна платника не збігається з країною карти), аномалію суми (платіж значно більший за звичні суми), потреба у додатковій аутентифікації. Ці ознаки дозволяють моделі враховувати потенційні ризики.

Цільова змінна означає успішність обробки на обраному шлюзі: успіх (так) або невдача (ні). Типи невдач включають: перевищення часу очікування, помилка з'єднання, недоступність шлюзу, невірні дані карти, неправильний код, невдача додаткової аутентифікації, відхилення видавцем карти, недостатньо коштів, блокування видавцем [10].

1.6.1 Процес накопичення та обробки даних

Дані для машинного навчання накопичуються через реєстрацію кожної успішної та невдалої спроби маршрутизації. На етапі Підтвердження Платежу система записує початкові параметри платежу та обраний шлюз. На етапі Транзакції Платежу система записує результат на шлюзі разом з деталями про час та причини помилок.

Історія спроб маршрутизації логується таким чином: якщо платіж не пройде на основному шлюзі, система реєструє цю невдачу та переходить до альтернативного шлюзу. Кожна спроба записується з номером спроби, обраним шлюзом, результатом та кодом помилки.

Обробка даних включає кілька етапів. На етапі очищення видаляються дублікати та невірні записи, обробляються пропущені значення. На етапі трансформації категоріальні характеристики з великою кількістю значень кодуються за допомогою методу середнього значення на основі успішності, а характеристики з малою кількістю значень кодуються через розширення категорій.

Масштабування числових характеристик проводиться для глибокого навчання, нормалізуючи їх до середнього значення нуль та стандартного відхилення один. Для моделей, заснованих на деревах рішень, часто використовується масштабування до діапазону від нуля до одиниці.

Створення нових характеристик включає перетворення часу через циклічні функції для врахування циклічної природи часу доби та дня тижня. Також створюються комбіновані характеристики, такі як високий ризик та потреба у додатковій аутентифікації. Обчислюються сумарні характеристики за останніми часовими періодами: кількість операцій платника за останню годину, відсоток невдач шлюзу за останню годину.

Розділення на набори проводиться за часовою послідовністю для запобігання витоку інформації: перші 70 відсотків часово впорядкованих даних для навчання, наступні 15 відсотків для налаштування параметрів моделей, останні 15 відсотків для перевірки якості моделей на невидимих раніше даних.

1.6.2 Обсяг та характеристики доступних даних у Corefy

Платформа Corefy накопичує великі обсяги платіжних даних за рахунок

обробки операцій від численних мерчантів у різних географічних регіонах та з різноманітними типами платежів.

Система обробляє приблизно 1-5 мільйонів платіжних операцій на день, по консервативним оцінкам. При річному обчисленні це складає приблизно 365-1825 мільйонів транзакцій на рік.

Кожна транзакція логується з детальною інформацією: параметри Платіжного Запиту, параметри Підтвердження Платежу, результати від Транзакції Платежу та метадані про взаємодію з провайдером. Загальна кількість параметрів на кожну транзакцію складає 50-100 залежно від деталізації.

Розрахунок обсягу даних: при 1 мільярді транзакцій на рік, 75 параметрів на транзакцію, загальний річний обсяг складає приблизно 600 терабайт. При консервативній оцінці 365 мільйонів транзакцій обсяг складає приблизно 220 терабайт на рік.

Успішність залежить від якості конфігурації. Типовий показник успішності при налаштованій системі складає 85-90 відсотків. Для нових мерчантів без історії успішність може бути нижче 80 відсотків, а для досвідчених мерчантів з великим обсягом транзакцій та оптимальною конфігурацією успішність може досягати навіть 95 відсотків, у рідких випадках [11].

Характеристики даних включають географічну розподіленість, часову динаміку (різні часові зони, циклічна активність за часом доби та днями тижня), різноманітність типів платежів та різноманітність шлюзів з різною комісійною структурою.

Успішні платежі складають від 85 до 95 відсотків, невдалі платежі складають 5-15 відсотків. Це вимагає спеціальних методів обробки при навчанні моделей.

1.7 Інтеграція ML в існуючу архітектуру маршрутизації

Запропонована архітектура передбачає одночасне функціонування поточної системи правил та нової системи машинного навчання. На етапі Підтвердження Платежу система може застосовувати обидва підходи: явні правила для встановлення початкових умов та модель машинного навчання для оптимізації вибору шлюзу.

Модель машинного навчання отримує на вхід характеристики платежу та шлюзу, формує вектор характеристик розміром 50-60 елементів та видає на виході для кожного доступного шлюзу ймовірність успіху або оцінку якості. Система обирає шлюз з найвищою оцінкою, дотримуючись обмежень мерчанта через Маршрутну Схему.

Зворотний зв'язок для системи машинного навчання надходить через реєстрацію результатів Транзакції Платежу. Якщо транзакція успішна, система одержує позитивний сигнал. Якщо невдала, система одержує негативний сигнал разом з причиною. Ці дані накопичуються та використовуються для поліпшення моделей машинного навчання.

1.8 Постановка задач дослідження

Для розробки гібридного підходу до маршрутизації платіжних операцій машинним навчанням у платіжних системах та розроблення практичних рекомендацій для впровадження у фінтех-компаніях поставлено наступні п'ять науково-практичних завдань.

Завдання 1 полягає у системному аналізі архітектури платіжних систем та методів маршрутизації.

Завдання 2 полягає у розробці класифікації та порівняльному аналізі методів ML з позицій придатності для маршрутизації. Також необхідно оцінити параметри, які впливають на успішність маршрутизації для кожного з методів.

Завдання 3 полягає у розробці математичної моделі задачі маршрутизації та формалізації набору ознак (features).

Завдання 4 полягає у розробці архітектури гібридної ML-системи з механізмами адаптації.

Завдання 5 полягає в реалізації прототипу системи ML-маршрутизації та проведенні експериментальної оцінки на синтетичних даних.

Завдання 6 полягає у порівняльному аналізі запропонованого підходу з базовими методами.

Завдання 7 полягає у розробці рекомендацій для впровадження ML маршрутизації на об'єкті автоматизації.

2. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ВИРІШЕННЯ ПРОБЛЕМИ МАРШРУТИЗАЦІЇ

Проблема вибору оптимального платіжного шлюзу для кожної транзакції є критично важливою для сучасних платіжних систем. За останнє десятиліття з'явилося кілька підходів до її вирішення – від простих статичних правил до складних моделей машинного навчання. У цьому підрозділі проаналізовано основні методи маршрутизації, їхні переваги та обмеження на основі реального досвіду провідних платіжних компаній.

2.1 Статичні правила маршрутизації

Статичні правила представляють собою найпростіший підхід до маршрутизації платежів, що базується на заздалегідь визначених конфігураціях розподілу трафіку. Типовий приклад такого правила: «Для всіх транзакцій VISA з США направляти 60% через шлюз А та 40% через шлюз В». Ці правила зазвичай формуються на основі історичних даних про успішність транзакцій та договірних зобов'язань із платіжними провайдерами.

Основними перевагами статичних правил є їхня простота імплементації та надзвичайно висока швидкодія – час прийняття рішення становить менше 1 мілісекунди. Бізнес-команди мають повний контроль над розподілом трафіку та можуть швидко вносити зміни без залучення технічних фахівців. Це особливо важливо в ситуаціях, коли потрібна негайна реакція на зміни в роботі певного шлюзу або при укладанні нових комерційних угод.

Проте статичні правила мають суттєві обмеження. Вони не адаптуються до змін у реальному часі – якщо певний шлюз починає працювати гірше, система продовжує направляти на нього заздалегідь визначений відсоток трафіку. Це

призводить до неоптимального розподілу транзакцій та втрати потенційних успішних платежів. За даними індустрії, системи на основі статичних правил досягають середнього рівня успішності близько 92%, що є базовим показником (baseline) для порівняння з більш складними методами [11].

2.2 Логістична регресія

Логістична регресія (LR) стала першим кроком у напрямку використання машинного навчання для маршрутизації платежів. Модель використовує сигмоїдну функцію для оцінки ймовірності успіху транзакції:

$$P(\text{success}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}} \quad (2.1)$$

де x_i – ознаки транзакції (сума, тип картки, країна, час доби) ;

β_i ваги, що навчаються на історичних даних.

Логістична регресія функціонує як дискримінативна модель для бінарної класифікації, де для кожного доступного шлюзу обчислюється ймовірність успіху, і транзакція направляється на шлюз з найвищою прогнозованою ймовірністю. Основними перевагами цього підходу є швидкість інференсу (близько 5 мілісекунд), математична стабільність та інтерпретованість – ваги ознак дозволяють зрозуміти, які фактори найбільше впливають на вибір шлюзу.

Однак логістична регресія має суттєве обмеження – вона може моделювати лише лінійні залежності та не враховує складні взаємодії між ознаками. Наприклад, модель не може автоматично виявити, що транзакції з певної країни краще проходять через конкретний шлюз лише в певні години доби. У результаті, незважаючи на покращення порівняно зі статичними правилами, загальна успішність рідко перевищує 89%, а в деяких випадках може бути навіть нижчою за добре налаштовані статичні правила [10].

2.3 Random Forest

Random Forest (RF) представляє собою ансамблевий метод, що складається з множини дерев рішень (зазвичай 100-500), кожне з яких навчається на випадковій підвибірці даних та ознак. Фінальне рішення приймається через голосування всіх дерев, що забезпечує високу стабільність та точність прогнозів.

Математично, для задачі регресії прогноз Random Forest визначається як:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^B T_b(x) \quad (2.2)$$

де B – кількість дерев у лісі;

$T_b(x)$ – функція прогнозу b -го дерева;

x – вектор ознак транзакції.

Для бінарної класифікації (успіх/невдача) кожне дерево прогнозує ймовірність, а остаточне рішення приймається як середня ймовірність всіх дерев.

Ключовими перевагами Random Forest є здатність моделювати нелінійні залежності, природна обробка категоріальних ознак без необхідності їх кодування, та стійкість до викидів у даних. Метод також надає корисну метрику важливості ознак, що дозволяє зрозуміти, які фактори найбільше впливають на успішність маршрутизації.

Основним обмеженням Random Forest є відносно висока латентність – 15-25 мілісекунд на прийняття рішення, що може бути критичним для високонавантажених систем. Також існує ризик перенавчання при великій кількості дерев та недостатньому обсязі даних. Проте, з огляду на доведену ефективність та надійність, Random Forest став де-факто індустріальним стандартом для інтелектуальної маршрутизації платежів [10].

2.4 Градієнтний бустинг: XGBoost та LightGBM

Методи градієнтного бустингу, зокрема XGBoost та LightGBM, представляють еволюцію ансамблевих методів, де дерева будуються послідовно, а кожне наступне дерево намагається виправити помилки попередніх.

XGBoost використовує цільову функцію з регуляризацією:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (2.3)$$

де перший член – це сума втрат на всіх прикладах;

другий член – регуляризація ансамблю.

Регуляризаційний член складається з L1 та L2 штрафів:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (2.4)$$

де γ – штраф за кількість листків дерева;

(T), λ – коефіцієнт L2 регуляризації ваг листків (w_j), що запобігає переоснащенню.

Такий підхід забезпечує баланс між точністю моделі та її складністю.

LightGBM відрізняється стратегією побудови дерев – замість рівномірного розбиття по глибині (level-wise), використовується розбиття по листках (leaf-wise), що дозволяє досягти кращої точності при меншій кількості ітерацій. Крім того, LightGBM використовує техніки Gradient-based One-Side Sampling (GOSS) та Exclusive Feature Bundling (EFB) для прискорення навчання на великих датасетах [5].

За результатами тестування на реальних платіжних даних, XGBoost досягає ROC-AUC (Receiver Operating Characteristic Area Under the Curve) 0.88-0.92, що на 3-5% краще за Random Forest. При цьому LightGBM демонструє

особливо вражаючі результати в швидкодії – середня латентність становить 12 мілісекунд, що майже вдвічі швидше за Random Forest при порівнянній точності [11].

Основною складністю використання градієнтного бустингу є необхідність тонкого налаштування гіперпараметрів – learning rate, максимальна глибина дерев, коефіцієнти регуляризації. Неправильне налаштування може призвести до перенавчання або недонавчання моделі. Також існує проблема холодного старту для нових типів транзакцій або шлюзів, де модель не має достатньо історичних даних.

2.5 Нейронні мережі

Нейронні мережі (NN) представляють найбільш гнучкий підхід до маршрутизації платежів, здатний виявляти складні нелінійні патерни в даних. Типова архітектура для цієї задачі включає:

- вхідний шар з нормалізацією ознак;
- 2-3 приховані шари з 128-256 нейронами кожен;
- функції активації ReLU або LeakyReLU;
- dropout для регуляризації (0.2-0.3);
- вихідний шар з softmax для вибору між шлюзами.

Процес навчання нейронної мережі використовує зворотне поширення помилки (backpropagation) з оптимізатором Adam та categorical cross-entropy як функцію втрат:

$$L = - \sum_{i=1}^N \sum_{j=1}^K y_{ij} \log(\hat{y}_{ij}) \quad (2.5)$$

де N – кількість прикладів у батчі;

K – кількість класів (доступних платіжних шлюзів);

u_{ij} – індикаторна змінна (1, якщо приклад i належить класу j , 0 в іншому випадку);

\hat{u}_{ij} – прогнозована ймовірність для класу j .

Оптимізатор Adam налаштовує темп навчання для кожного параметра, що прискорює конвергенцію порівняно з простим градієнтним спуском.

Незважаючи на високу потенційну точність (93-94% успішності), нейронні мережі мають серйозні обмеження для production-використання. Головною проблемою є латентність – 30-50 мілісекунд на інференс, що в 3-4 рази більше за LightGBM. Це критично для високонавантажених систем, де кожна додаткова мілісекунда затримки може призвести до втрати конверсії.

Іншою важливою проблемою є інтерпретованість – нейронні мережі функціонують як «чорна скринька», що ускладнює діагностику помилок та пояснення рішень регуляторам [12].

2.6 Порівняльний аналіз методів

Для систематичного порівняння розглянутих методів наведено таблицю 2.1 з ключовими характеристиками, які отримані з [7, 8, 10, 13, 15].

Додатково варто розглянути економічну ефективність кожного методу. За розрахунками, кожен додатковий процентний пункт успішності платежів приносить додаткові \$2-5 мільйонів річного доходу на кожен мільярд доларів обороту. При цьому витрати на впровадження та підтримку ML-моделей становлять:

- Random Forest: \$50-100К на рік (інфраструктура + команда);
- LightGBM: \$60-120К на рік;
- нейронні мережі: \$150-300К на рік [14].

Таблиця 2.1– Порівняння методів маршрутизації платежів

Метод	Точність успішності	Латентність	Адаптивність	Інтерпретованість	Швидкість інтеграції
Статичні правила [8]	85-90%	<1 мс	Х	✓✓✓	✓✓✓
Логістична регресія [10]	89%	5 мс	Х	✓✓✓	✓✓✓
Random Forest [10]	92%	20 мс	Х	✓✓	✓✓✓
XGBoost [13]	93%	25 мс	Х	✓	✓✓
LightGBM [14]	92%	12 мс	Х	✓	✓✓✓
Нейронні мережі [7]	93-94%	40 мс	✓	Х	✓

Примітка: ✓✓✓ – відмінно, ✓✓ – добре, ✓ – задовільно, Х – незадовільно

Таким чином, ROI від впровадження інтелектуальної маршрутизації може досягати 20-50х протягом першого року [7].

2.7 Висновок по другому розділу

Аналіз існуючих методів маршрутизації платежів показує чітку еволюцію від простих правил до складних моделей машинного навчання. Статичні

правила, попри свою простоту та швидкість, не забезпечують достатню ефективність для сучасних платіжних систем з їх складними патернами та динамічними змінами.

Логістична регресія покращує базовий підхід, але її лінійність обмежує потенціал оптимізації. Random Forest довів свою ефективність на практиці, ставши стандартом індустрії завдяки балансу між точністю та надійністю – кейс Razorpay з покращенням на 4.1% є переконливим доказом.

LightGBM виявився оптимальним вибором за співвідношенням швидкість/якість, забезпечуючи високу точність при латентності 12 мілісекунд. Це робить його ідеальним кандидатом для високонавантажених production-систем [11].

Нейронні мережі демонструють найвищий потенціал точності, але їхня висока латентність (40+ мілісекунд) та низька інтерпретовність роблять їх ризикованим вибором для критичних платіжних систем. Проте, з розвитком апаратного забезпечення та методів оптимізації інференсу, цей підхід може стати більш практичним у майбутньому.

На основі проведеного аналізу, оптимальним рішенням для сучасної платіжної системи є гібридна архітектура, що поєднує:

- LightGBM як основну модель для швидкого прийняття рішень;
- Random Forest як резервну модель для забезпечення надійності;
- елементи адаптивності для реагування на зміни в реальному часі;
- A/B-тестування для безпечного впровадження нових моделей.

3. РОЗРОБКА МЕТОДУ ІНТЕГРАЦІЇ МАШИННОГО НАВЧАННЯ В СИСТЕМУ ОБ'ЄКТА АВТОМАТИЗАЦІЇ

3.1 Обґрунтування комбінованого підходу до оптимізації маршрутизації

Аналіз архітектури платіжної системи Corefy виявив, що інтеграція машинного навчання повинна враховувати як статичні бізнес-правила мерчантів, так і динамічні характеристики роботи платіжних провайдерів у реальному часі.

Системи на основі правил не адаптуються до змін умов обробки платежів протягом дня, тоді як машинне навчання самостійно виявляє закономірності в успішності різних комбінацій (платіжний метод, географія, шлюз, час обробки).

На противагу поточним системам, що використовують винятково ручне конфігурування правил, запропонований метод поєднує чотири послідовні етапи фільтрації та ранжування шлюзів.

3.2 Архітектура гібридної системи маршрутизації

Запропонована архітектура складається з п'яти послідовних компонентів, кожен з яких виконує специфічну роль у процесі вибору оптимального маршруту:

Модуль 1: Статична фільтрація на основі бізнес-правил

На першому етапі система застосовує явні обмеження мерчанта, визначені у Маршрутній Схемі: набір дозволених шлюзів, географічні обмеження, ліміти по типам платіжних інструментів. Цей крок необхідний для дотримання контрактних зобов'язань та управління ризиками. Вихідний список терміналів передається на наступний модуль.

Модуль 2: Виявлення простоїв провайдерів

На цьому етапі застосовується логістична регресія для прогнозування потенційних простоїв шлюзів на основі реальних метрик:

$$P(\text{downtime}_g) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot \text{SR}_g + \beta_2 \cdot \text{latency}_g + \beta_3 \cdot \text{volume}_g)}} \quad (3.1)$$

де SR_g – коефіцієнт успішності шлюзу g за останню годину;

latency_g – середня затримка відповіді;

volume_g – поточне навантаження.

Термінали з прогнозованою ймовірністю простою вище порога 0,7 виключаються з подальшої обробки.

Модуль 3: Розрахунок адаптивних ознак в реальному часі

Для кожного із залишившихся терміналів система обчислює набір динамічних ознак, що оновлюються після кожної транзакції. Ознаки розділяються на три категорії:

Часові ознаки (time-window-based): успішність за останні 5 сек, 30 сек, 5 хв; обсяг транзакцій за часові вікна; середня затримка. Ці ознаки оновлюються з експоненціальним загасанням за формулою:

$$f_{\text{decay}}(t) = f_{\text{prev}} \cdot 2^{-(t-t_{\text{prev}})/\text{half-life}} \quad (3.2)$$

де half-life для часових ознак становить 30–60 сек залежно від волатильності провайдера.

Подійні ознаки (event-based): успішність за останні 10 та 100 транзакцій; кількість послідовних невдач; часова дистанція від останньої невдачі. Цей тип ознак краще відображає миттєві зміни в роботі провайдера.

Глобальні ознаки: сумарна успішність системи, відсоток трафіку на кожного провайдера, стан всіх терміналів мерчанта. Вони допомагають моделі враховувати контекст операцій.

Модуль 4: Ансамблева класифікація та ранжування

На цьому етапі для кожного терміналу обчислюється ймовірність успіху за допомогою LightGBM (Light Gradient Boosting Machine) – сучасної реалізації методу градієнтного бустингу. На відміну від простих методів (логістична регресія) або басового ансамблю (Random Forest), LightGBM будує послідовність дерев рішень, де кожне наступне дерево фокусується на виправленні помилок попередніх. Модель навчена на історичних даних останніх 6–8 тижнів, виключаючи транзакції з явними помилками платника (невірний ОТР, недостатньо коштів). Вихідні ймовірності служать базисом для ранжування терміналів.

LightGBM розв'язує задачу регресії мінімізацією об'єктивної функції::

$$L(\theta) = \sum_{i=0}^n l(y_i, \hat{y}_i) + \sum_{k=0}^m \Omega(f_k) \quad (3.3)$$

де $y_i \in \{0,1\}$ – фактичний результат i -ї транзакції (0 – невдача, 1 – успіх);;

\hat{y}_i – прогнозоване значення моделлю;

$l(y_i, \hat{y}_i)$ – функція втрат (для бінарної класифікації використовується лог-втрата);

$\Omega(f_k)$ – член регуляризації для k -го дерева;

m – загальна кількість послідовно доданих дерев.

Прогноз для терміналу t на платежі P обчислюється як сума прогнозів всіх дерев у послідовності:

$$p_t = \sigma(\sum_{k=0}^m f_k(x_t)) \quad (3.4)$$

де $f_k(x_t)$ – прогноз k -го дерева на векторі ознак x_t ;

$\sigma(\cdot)$ – сигмоїдна функція, що нормалізує суму в діапазон $[0, 1]$;

$m \in [100, 200]$ – глибина ансамблю, оптимізована на валідаційному наборі.

Модуль 5: Каскадна обробка та механізм зворотного зв'язку

Система спочатку спрямовує платіж на термінал з найвищою прогнозованою ймовірністю успіху. Якщо платіж не пройде (помилка з'єднання, timeout, відхилення провайдером), система автоматично переводить його на наступний за рейтингом термінал.

При цьому:

- кожна невдала спроба реєструється з часовою міткою та кодом помилки;
- після отримання результату (упродовж 15–30 сек) ознаки терміналу оновлюються у базі даних в режимі реального часу;
- невдала спроба знижує оцінку терміналу через експоненціальне загасання; успішна спроба на резервному терміналі підвищує його оцінку.

3.3 Формалізація задачі оптимізації маршруту

Постановка задачі може бути сформульована як задача навчання з учителем з елементами адаптивного управління. На кожен момент часу t система спостерігає стан шлюзів:

$$S_t = \{\mu_1(t), \mu_2(t), \dots, \mu_m(t)\} \quad (3.5)$$

де $\mu_i(t)$ – поточна оцінка успішності провайдера i .

Ці оцінки виводяться з реальних даних, але в системі вважаються невідомими до спостереження результатів.

Для кожної транзакції система обирає впорядкований список терміналів:

$$A_t = [a_1, a_2, \dots, a_k] \quad (3.6)$$

де a_1 – первинний термінал;

a_2 – перший резервний, і т.д.

Система отримує сигнал $r_{t \setminus in 0,1}$ через 15–30 сек після спроби маршрутизації. При успіху $r_t = 1$, при невдачі $r_t = 0$.

Мета оптимізації: максимізувати кумулятивну успішність за період T :

$$\max_{A_1, A_2, \dots, A_T} \sum_{t=1}^T r_t(A_t), \quad (3.7)$$

при обмеженнях:

- кожен термінал має максимальну пропускну спроможність TPS (транзакцій на секунду);
- система повинна мати латентність < 5 мс на прийняття рішення;
- всі транзакції платника мають залишатися в межах установлених торговцем правил маршрутизації.

Ця задача належить до класу non-stationary multi-armed bandits із затримкою зворотного зв'язку, оскільки умови роботи провайдерів змінюються протягом дня (час пік, обслуговування, географічні зміни), а інформація про результат надходить із затримкою [4].

3.4 Алгоритм вибору оптимального маршруту

Запропонований алгоритм HYBRID-ROUTE є центральним елементом методу інтеграції машинного навчання в систему маршрутизації платіжних транзакцій об'єкта автоматизації. Його призначення – формувати для кожного платежу впорядкований список терміналів (платіжних маршрутів), відсортованих за прогнозованою ймовірністю успішної обробки, з урахуванням як бізнес-обмежень мерчанта, так і поточного технічного стану платіжних провайдерів.

Ключова ідея алгоритму полягає в поєднанні статичних та динамічних компонентів у єдиному процесі прийняття рішень. Спочатку застосовуються статичні правила (Маршрутна Схеми мерчанта, обмеження по країні, валюті, типу інструменту тощо), далі – модуль виявлення простоїв на основі логістичної регресії, після чого використовуються динамічні ознаки в реальному часі та модель LightGBM для оцінки ймовірності успіху на кожному терміналі. Такий підхід є природним продовженням існуючої архітектури Corefy: алгоритм вбудовується між фасадними сервісами та шаром інтеграції з провайдерами, не змінюючи бізнес-інтерфейсів, але радикально покращуючи якість вибору маршруту.

Таблиця 3.1 – Таблиця позначень

Позначення	Опис
1	2
P	Об'єкт «платіж» з параметрами: сума, валюта, ідентифікатор мерчанта, тип карти, географія, час створення.
S	Маршрутна Схеми (Routing Scheme) для мерчанта: набір бізнес-правил та обмежень маршрутизації.
T	Множина всіх доступних терміналів/маршрутів у системі.
$t \in T$	Окремий термінал (маршрут) до конкретного платіжного провайдера.
M	Навчена модель LightGBM, що оцінює ймовірність успіху транзакції на терміналі.

Продовження таблиці 3.1

1	2
F	База онлайн-ознак (feature store) з поточними оцінками стану терміналів.
candidates	Підмножина терміналів, які задовольняють статичним бізнес-правилам для платежу P
viable	Підмножина терміналів, що пройшли фільтр за ймовірністю простою (downtime).
$\mu_{\text{down}}(t)$	Прогнозована ймовірність простою / деградації терміналу t (вихід логістичної регресії).
\mathbf{x}_t	Вектор ознак, що описує стан терміналу t з урахуванням контексту платежу P.
\hat{p}_t	Прогнозована ймовірність успіху платежу P на терміналі t, обчислена моделлю M.
scored_terminals	Список пар (t, \hat{p}_t) для всіх життєздатних терміналів.
k	Кількість топових терміналів, які обираються для каскадної обробки (зазвичай $k = 3-5$).
attempt_order	Впорядкований за \hat{p}_t список з k рекомендованих терміналів.
Δt	Час, що минув з моменту останнього оновлення ознак терміналу.

Алгоритм HYBRID-ROUTE

Вхідні дані:

- платіж P з параметрами (сума, валюта, мерчант, карта, географія, час);
- Маршрутна Схема S з правилами мерчанта;
- модель LightGBM M , навчена на історії останніх 6 тижнів;
- база ознак F з онлайн-оцінками терміналів.

Вихідні дані – `attempt_order` – впорядкований список рекомендованих терміналів для каскадної обробки.

Лістинг 3.1 – Програмний код для функції гібридної маршрутизації

```

1: function HYBRID_ROUTE(P, S, M, F)
2:   candidates ← GET_TERMINALS(S, P)
3:   viable ← ∅
4:   for each t in candidates do
5:     metrics_t ← GET_RUNTIME_METRICS(t)
6:     downtime_prob ← LOGISTIC_REGRESSION(metrics_t)
7:     if downtime_prob < 0.7 then
8:       viable ← viable ∪ {t}
9:     end if
10:  end for

11:  scored_terminals ← ∅
12:  for each t in viable do
13:    features ← EXTRACT_FEATURES(t, F, P)
14:    timeElapsed ← CURRENT_TIME() -
features.last_update_time
15:    features ← APPLY_DECAY(features, timeElapsed)
16:    prob_success ← M.PREDICT(features)
17:    scored_terminals ← scored_terminals ∪ {(t,
prob_success)}
18:  end for

19:  scored_terminals ← SORT_DESC(scored_terminals by
prob_success)
20:  attempt_order ← TOP_K(scored_terminals, k)
21:  return attempt_order
22: end function

```

3.5 Пояснення етапів алгоритму

Функція `GET_TERMINALS(S, P)` реалізує перший рівень відбору маршрутів:

- враховує обмеження Маршрутної Схеми S (дозволені провайдери для даного мерчанта, типи карт, країни, валюти);
- відфільтровує термінали, які принципово не підходять для даного платежу (наприклад, локальні методи, недоступні у країні платника).

У результаті отримуємо множину `candidates`, яка містить тільки ті термінали, що узгоджені з бізнес-логікою та контрактами мерчанта. На цьому етапі не враховується поточний технічний стан провайдерів.

Наступний цикл по `candidates` реалізує модуль стійкості до відмов:

- `GET_RUNTIME_METRICS(t)` отримує поточні агреговані метрики терміналу t : недавній `success rate`, середню затримку, частоту помилок, індикатори технічних збоїв;
- `LOGISTIC_REGRESSION(metrics_t)` обчислює ймовірність того, що термінал перебуває в стані простою або деградації – $\mu_{\text{down}}(t)$;
- якщо $\mu_{\text{down}}(t) \geq 0.7$, термінал вважається «ризиковим» та не потрапляє до множини `viable`.

На виході цього етапу маємо `viable` – набір технічно працездатних терміналів, які одночасно відповідають і бізнес-правилам, і поточним вимогам до надійності.

Третій блок реалізує основну ML-компоненту алгоритму:

а) `EXTRACT_FEATURES(t, F, P)` витягує з бази ознак F поточний вектор \mathbf{x}_t , що описує:

- 1) історичну успішність терміналу за короткі часові вікна (наприклад, останні 5 секунд, 30 секунд, 10 подій);
- 2) агреговані характеристики по платнику, мерчанту, BIN-коду карти, країні;

3) глобальні показники навантаження та латентності.

Додатково до \mathbf{x}_t можуть бути включені контекстні характеристики платежу P (сума, тип інструменту, географія);

б) `timeElapsed` фіксує, скільки часу минуло з моменту останнього оновлення ознак для цього терміналу;

в) `APPLY_DECAY(features, timeElapsed)` застосовує експоненціальне загасання до динамічних ознак, щоб надати більшої ваги свіжим даним та зменшити вплив застарілих спостережень;

г) `M.PREDICT(features)` повертає $\hat{p}_t \in [0,1]$ – оцінку ймовірності успішної авторизації для платежу P на терміналі t , згідно з моделлю LightGBM.

Пара (t, \hat{p}_t) додається до до списку `scored_terminals`. На цьому етапі гібридний підхід завершується персоналізованим ML-ранжуванням для кожного доступного маршруту.

Ранжування та формування списку для каскадної обробки:

а) `SORT_DESC(scored_terminals by prob_success)` впорядковує всі пари (t, \hat{p}_t) у порядку спадання \hat{p}_t . Верхні елементи списку відповідають терміналам з максимальною прогнозованою ймовірністю успіху;

б) `TOP_K(scored_terminals, k)` обирає перші k терміналів. Це і є маршрутна послідовність для каскадної обробки:

- 1) перший термінал використовується як основний маршрут;
- 2) у разі технічної або логічної невдачі платіж може бути автоматично перекинутий на наступний у списку, без повторного звернення до ML-модуля.

Повернутий список `attempt_order` безпосередньо передається в модуль, що керує каскадними повторними спробами проведення платежу та фактичною взаємодією з платіжними провайдерами.

Алгоритм HYBRID-ROUTE виконує роль `decision engine` між бізнес-рівнем (Маршрутні Схеми, налаштування мерчантів) та технічним рівнем (Provider Hub, Providers Gate). Його використання дозволяє:

- автоматизувати перехід від ручних статичних правил до адаптивної

маршрутизації;

- підвищити частку успішних транзакцій за рахунок контекстного вибору найкращого терміналу;
- зберегти сумісність з поточною логікою маршрутизації Corefy за рахунок повернення списку терміналів у вже прийнятному для системи форматі;
- закласти основу для подальшого розширення (врахування комісійних витрат, пріоритезація ROI, використання бандит-алгоритмів тощо).

Таким чином, алгоритм HYBRID-ROUTE формалізує запропонований метод інтеграції машинного навчання у систему маршрутизації та може розглядатися як ядро проєктованого ML-компонента платформи Corefy.

3.6 Механізми адаптації в режимі реального часу

Після отримання результату система оновлює ознаки терміналу в базі даних (Redis або подібне сховище з низькою латентністю):

$$SR_t^{new} = SR_{t-1} \cdot 2^{-1/\tau} + (1 - 2^{-1/\tau}) \cdot outcome \quad (3.8)$$

де τ – параметр часового вікна в секундах (зазвичай 60–300 сек залежно від волатильності провайдера);

$outcome \in \{0,1\}$ – результат спроби маршрутизації платежу через конкретного провайдера, де 0 – відхилення платежу, 1 – успішна обробка.

Цей метод забезпечує, що недавні результати важать більше, ніж давні, що критично важливо в середовищі платіжних систем, де умови змінюються швидко.

Модель LightGBM переоцінюється щотижня на основі накопичених за тиждень даних. Для запобігання катастрофічному забуванню та забезпечення стійкості до концептуального дрейфу система:

- зберігає множину моделей (поточна, попередня, версія для А/В тестів);
- застосовує техніку stratified time-series split, розділяючи дані на навчальний (70%), валідаційний (15%) та тестовий (15%) набори по часовій послідовності без витоку інформації;
- моніторить метрику precision на тестовому наборі; якщо вона падає більш ніж на 2%, модель не розгортається.

3.7 Інтеграція з архітектурою Corefy

Запропонований метод інтегрується в Corefy на рівні компонента Provider Hub таким чином:

- а) на вхід до ML-модуля надходять параметри платежу з компоненту Facade API та поточні ознаки терміналів з Redis;
- б) на виході модуль видає впорядкований список терміналів з прогнозованими ймовірностями успіху;
- в) компонента Providers Gate використовує цей список для послідовної обробки терміналів;
- г) IPN Handler реєструє результати та оновлює ознаки у Redis без блокування основного потоку обробки платежів;
- д) перебудова моделі проводиться на окремому сервісі, не впливаючи на production систему.

Розроблений метод передбачає валідацію на двох рівнях:

- тестування на синтетичному датасеті (наборі даних) з контрольованою динамікою. Генерація даних, що відтворюють реальні закономірності (часові цикли, раптові зміни успішності), для верифікації механізму адаптації;
- у майбутньому А/В тестування на реальному трафіку Corefy з розділенням користувачів на контрольну групу (поточна система) та експериментальну групу (ML-система) протягом мінімум 4 тижнів.

3.8 Вибір архітектури ML-моделі

На основі порівняльного аналізу методів маршрутизації обрано LightGBM як основну модель інференсу для етапу ансамблевої класифікації та ранжування. Цей вибір ґрунтується на наступних причинах:

- оптимальне співвідношення латентності та точності. LightGBM демонструє латентність інференсу ≈ 12 мс на сучасному обладнанні, що є майже вдвічі швидшим за Random Forest (15–20 мс) та майже в 3–4 рази швидшим за нейронні мережі (40+ мс) [13]. Така латентність дозволяє інтегрувати модель у критичний шлях обробки платежів Corefy без збільшення загального часу прийняття рішення;

- висока точність на реальних даних платіжних систем. За даними досліджень Razorpay та інших платіжних агрегаторів, LightGBM досягає ROC-AUC близько 0.7949–0.8050 та precision близько 0.9430–0.9469. Ці показники перевищують базові методи (статичні правила – 90%, логістична регресія – 89%) і конкурують з Random Forest, при цьому забезпечуючи істотно вищу швидкість інференсу [10];

- стійкість до перенавчання та адекватна інтерпретованість. На відміну від нейронних мереж, які часто функціонують як «чорна скринька», LightGBM забезпечує feature importance – метрику, що показує, які ознаки найбільше впливають на прогноз. Це дозволить команді Corefy діагностувати причини змін у маршрутизації, виявляти аномалії роботи провайдерів та пояснювати рішення регуляторам та мерчантам.

LightGBM також менш сприйнятлива до перенавчання порівняно з глибокими нейронними мережами завдяки вбудованим механізмам регуляризації (L1/L2 штрафи), при цьому здатна виявляти складні нелінійні залежності між ознаками терміналу та ймовірністю успіху.

3.9 Висновок по третьому розділу

Розроблений метод інтеграції машинного навчання у систему маршрутизації Corefy формалізовано та підготовлено до реалізації. Його гібридність дозволяє балансувати між бізнес-контролем, технічною надійністю та алгоритмічною ефективністю, що є критично важливим для сучасних фінтех-платформ. Наступний крок – прототипування та експериментальна валідація на синтетичних даних.

4. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

4.1 Опис експериментального середовища

4.1.1 Датасет: структура, обсяг, підготовка даних

Експериментальні дослідження проводились на синтетичному наборі даних платіжних транзакцій, що моделює реальні умови роботи платіжної системи. Цей набір даних складається з навчального набору обсягом 1 мільйон транзакцій та тестового набору обсягом 200 тисяч транзакцій.

Датасети охоплюють період з 1 січня по 31 грудня 2025 року, моделюючи повний річний цикл роботи платіжної системи. Загалом датасети містять 2000 унікальних мерчантів, близько 100 000 унікальних клієнтів, 30 платіжних шлюзів, 32 країни Європейського регіону (включаючи Україну та Грузію) та 13 валют.

Кожна транзакція описується 45 ознаками, які згруповано у 12 категорій (табл. 4.1).

Таблиця 4.1– Категорії ознак датасету

Категорія	Кількість	Приклади ознак
1	2	3
Ідентифікатори	3	transaction_id, merchant_id, customer_id
Транзакційні	3	amount, currency, payment_method
Карткові	4	card_type, card_brand, card_country, bin
Географічні	3	merchant_country, customer_country, is_cross_border
МСС	3	mcc, mcc_category, mcc_risk_level
Часові	9	hour_of_day, day_of_week, is_payday_period, is_holiday_period
Аутентифікація	2	is_3ds, 3ds_outcome

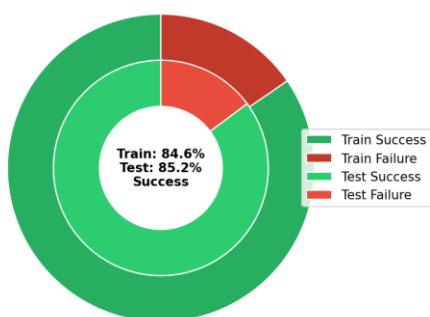
Продовження таблиці 4.1

1	2	3
Ризик та верифікація	4	risk_score, velocity_1h, avs_result, cvv_result
Профіль клієнта	3	customer_is_returning, customer_historical_success_rate
Характеристики шлюзу	5	gateway_chosen, gateway_fee_percent, gateway_success_1h
Результат	5	gateway_success, decline_code, decline_reason, is_soft_decline

Після застосування конструювання ознак та кодування їх кількості збільшилась – 52 для LightGBM (кодування мітками) та 147 для Random Forest (цифрове кодування). Цільовою змінною є бінарна ознака gateway_success, що відображає успішність обробки транзакції обраним платіжним шлюзом.

Розподіл цільової змінної характеризується дисбалансом: частка успішних транзакцій становить 84,59% у навчальному та 85,17% у тестовому наборах. Цей набір даних охоплює 30 унікальних платіжних провайдерів, що дозволяє змоделювати реалістичні сценарії вибору оптимального маршруту. На рисунку 4.1 представлено порівняння розподілу в обох датасетах.

Порівняння розподілу: Train vs Test



Розподіл цільової змінної: Train vs Test

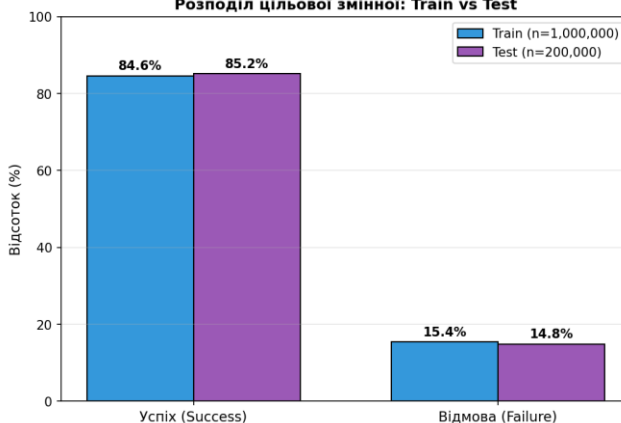


Рисунок 4.1 – Розподіл цільової змінної у датасетах

Датасет реалізує спрощену версію структури ознак, визначеної у розділі 1.6, з фокусом на ознаках, що мають найбільший вплив на успішність маршрутизації згідно з галузевими дослідженнями та включає п'ять основних категорій. На рисунку 4.2 представлено гістограму розподілу ключових ознак.

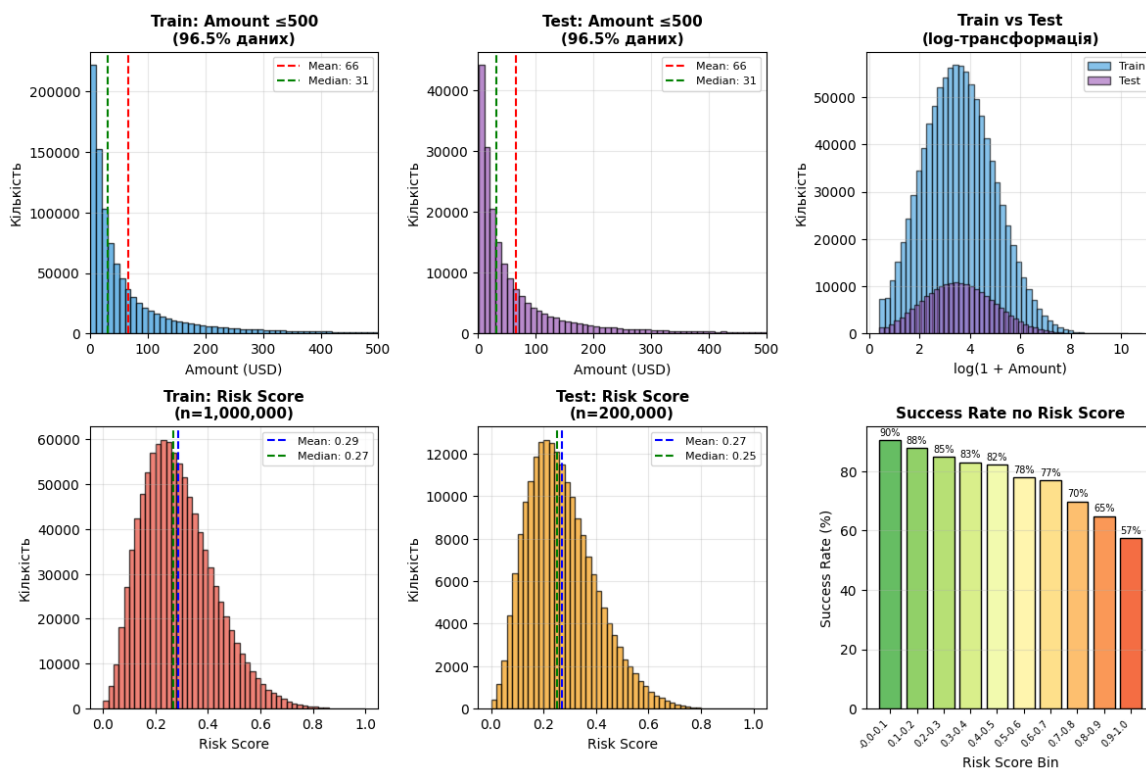


Рисунок 4.2 – Гістограми розподілу ключових ознак (amount, risk_score).

Характеристики платежу охоплюють суму транзакції, валюту, метод оплати, BIN-код картки, тип та бренд картки, країну видачі картки, країну мерчанта, код категорії мерчанта (MCC) та відповідний рівень ризику. Часові характеристики включають час транзакції, годину доби, день тижня, ознаки вихідного дня, періоду виплати зарплат та святкового періоду.

Характеристики клієнта містять ідентифікатор, країну, ознаку повторного клієнта, історичний показник успішності та кількість попередніх транзакцій.

Характеристики шлюзу включають ідентифікатор обраного провайдера, комісію та історичні показники успішності за останню годину та добу.

Ознаки безпеки охоплюють індикатор крос-бордер транзакції, використання 3D Secure, токенизацію картки, оцінку ризику та результати верифікації AVS/CVV.

Для запобігання витоку інформації з цільової змінної з навчання виключено ознаки, що є наслідками результату транзакції: час відповіді шлюзу, код та причина відмови, тип відмови та ймовірність успіху при повторній спробі.

На основі аналізу досліджень у галузі оптимізації платіжної маршрутизації реалізовано комплекс покращень при конструюванні ознак. Для суми транзакції застосовано логарифмічну трансформацію (\log_{1p}) для обробки важких хвостів розподілу, бінування за розміром (micro, small, medium, large, high) та виділення ознаки округленої суми. Часові ознаки закодовано циклічно за допомогою синус-косинус трансформації для години доби, дня тижня, дня місяця та місяця, що дозволяє моделі коректно враховувати циклічність часових патернів.

Для характеристик шлюзу створено додаткові ознаки: співвідношення короткострокового та довгострокового success rate (gateway_sr_ratio) та їх різницю (gateway_sr_diff), що дозволяє виявляти тренди у продуктивності шлюзів. Для оцінки ризику створено категоріальне групування (risk_bin) та крос-ознаку is_3ds_crossborder.

Для категоріальних ознак з високою кардинальністю (MCC, merchant_country, card_country) застосовано Target Encoding з Bayesian smoothing за формулою

$$\text{encoded} = \frac{n}{n+m} \text{category}_{\text{mean}} + \frac{m}{n+m} \text{global}_{\text{mean}} \quad (4.1)$$

де $m=20$ – параметр згладжування.

Цей підхід зменшує ризик перенавчання на рідкісних категоріях та забезпечує кращу генералізацію порівняно з one-hot encoding.

Реалізовано двоетапну процедуру відбору ознак. На першому етапі видалено 8 ознак з низькою дисперсією (threshold=0.01). На другому етапі

виключено 9 ознак з високою кореляцією ($r > 0.95$) для уникнення мультиколінеарності та покращення інтерпретованості моделі.

4.1.2 Baseline: поточна rule-based система Corefy

Для коректного порівняння ефективності ML-моделей визначено кілька базових підходів (baseline). Majority Class Baseline передбачає завжди класифікувати транзакцію як успішну, що забезпечує точність 85,17%, яка відповідає частці позитивного класу в даних. Проте такий підхід має ROC-AUC рівний 0.5, що свідчить про відсутність дискримінаційної здатності.

Базовий показник історичної ставки за шлюзом використовує історичний показник успішності кожного шлюзу для прогнозування. Цей підхід імітує прості статичні правила маршрутизації, де рішення приймається на основі загальної статистики провайдера.

Найбільш релевантним baseline є Оптимальний шлюз на основі правил, що моделює оптимізовану систему статичних правил. Аналіз показав, що найкращим шлюзом за показником успішності є Trustly із $SR = 90,17\%$. Симуляція маршрутизації всіх транзакцій через цей шлюз дає успішність 90,06%, що є майже верхньою межею для простих rule-based систем.

4.1.3 Метрики оцінки

Для оцінки якості моделей використано розширений комплекс метрик. Класифікаційні метрики включають Accuracy (загальна точність), Precision (точність позитивних передбачень), Recall (повнота виявлення успішних транзакцій), F1-Score (гармонійне середнє Precision та Recall), ROC-AUC (площа

під ROC-кривою), PR-AUC (площа під Precision-Recall кривою, особливо інформативна для незбалансованих даних) та Matthews Correlation Coefficient (MCC, збалансована метрика для бінарної класифікації).

Операційні метрики охоплюють латентність inference (середнє значення, 50-й, 95-й та 99-й перцентилі) та час навчання моделі. Згідно з вимогами, латентність Random Forest не повинна перевищувати 20 мс, а LightGBM – 12 мс.

Бізнес-метрики включають Success Rate Uplift (приріст успішності відносно baseline) та відсоток транзакцій, для яких модель рекомендує інший шлях, що відображає потенціал оптимізації маршрутизації.

Для інтерпретації моделей використано SHAP (SHapley Additive exPlanations) – метод, що забезпечує теоретично обґрунтовані оцінки важливості ознак з урахуванням їх взаємодій.

4.1.4 Технічне середовище

Експерименти проводились на комп'ютері з процесором Apple M4 та 24 ГБ оперативної пам'яті під управлінням операційної системи macOS. Програмне середовище базувалось на Python 3.10 із використанням бібліотек pandas, numpy, scikit-learn для Random Forest, lightgbm для градієнтного бустингу та shap для інтерпретації моделей. Візуалізація результатів виконана засобами matplotlib.

Оптимальні гіперпараметри моделей визначено за допомогою RandomizedSearchCV з 10 ітераціями та 3-кратною крос-валідацією на підвибірці 50 тисяч записів. Для Random Forest оптимальною виявилась конфігурація з 400 деревами, максимальною глибиною 30 та мінімумом 5 зразків у листі. Для LightGBM – 200 ітерацій бустингу, 31 лист, максимальна глибина 15, швидкість навчання 0.05, subsample 0.8 та colsample_bytree 0.7.

Процес експерименту включав послідовне виконання етапів: завантаження даних, конструювання ознак, target encoding, підбір гіперпараметрів, навчання

моделей (300К записів для RF, 1М для LightGBM), оцінка метрик на тестовому наборі, вимірювання латентності на 1000 одиночних записах та SHAP-аналіз важливості ознак.

4.2 Результати експериментів

Експериментальна оцінка показала значне покращення метрик після застосування конструювання ознак. Random Forest досяг Accuracy 89,7%, Precision 91,1%, Recall 97,4%, F1-Score 94,2%, ROC-AUC 86,9% та PR-AUC 97,4%. Латентність становила 19,2 мс (mean), що відповідає вимозі <20 мс. MCC склав 0,53, що свідчить про добру збалансовану якість класифікації.

LightGBM на повному наборі даних (1М записів) досяг Accuracy 77,1%, Precision 94,7%, Recall 77,5%, F1-Score 85,2%, ROC-AUC 87,3% та PR-AUC 97,5%. Латентність становила 0,39 мс (mean), що значно краще за вимогу <12 мс. MCC склав 0,41.

Детальний аналіз розподілу ймовірностей виявив, що LightGBM потребує калібрування порогу класифікації. При стандартному порозі $t=0.50$ модель демонструє низький Recall (77,5%), оскільки розподіл передбачених ймовірностей зміщений вліво. Оптимальний поріг $t=0.40$ забезпечує значно кращий баланс метрик:

При оптимальному порозі LightGBM досягає порівнянних з Random Forest показників Recall (91,3% порівнянно з 97,4%) при збереженні переваг у латентності. Для production рекомендовано використовувати $t=0.40$

Random Forest демонструє значно вищу Accuracy (89,7% проти 86,2%) та Recall (97,4% проти 91,3%), що робить його кращим вибором для задач, де критично важливо не пропустити успішні транзакції. LightGBM з оптимальним порогом характеризується вищим ROC-AUC (87,3% проти 86,9%) та у 49 разів

меншою латентністю (0.39 мс проти 19.2 мс), що робить його оптимальним для впровадження.

На основі отриманих результатів рекомендовано використовувати LightGBM як основну модель для виробничого-середовища завдяки оптимальному балансу між якістю та швидкістю, а Random Forest – для сценаріїв, де критично важливий високий Recall.

4.2.1 Порівняння ML-моделей (Random Forest, та LightGBM)

Для вирішення задачі прогнозування успішності транзакції досліджено дві моделі машинного навчання: Random Forest та LightGBM. Вибір цих алгоритмів обумовлений їх високою ефективністю на табличних даних, стійкістю до перенавчання та здатністю працювати з категоріальними ознаками.

Random Forest є ансамблевим методом, що базується на побудові множини дерев рішень з подальшим усередненням їх передбачень. Кожне дерево навчається на випадковій підвибірці даних (bootstrap) та випадковій підмножині ознак, що забезпечує декореляцію дерев та зменшує дисперсію моделі. У проведених експериментах оптимальна конфігурація включала 400 дерев з максимальною глибиною 30. Модель досягла ROC-AUC 86,9%, Precision 91,1%, Recall 97,4% та PR-AUC 97,4%. Середня латентність inference становила 19,2 мс при часі навчання 70 секунд на 300К записів.

LightGBM є реалізацією градієнтного бустингу з оптимізаціями для швидкості та ефективності використання пам'яті. Алгоритм використовує leaf-wise стратегію росту дерев та гістограмний підхід до знаходження оптимальних точок розбиття. Оптимальна конфігурація включала 200 ітерацій бустингу з 31 листом, максимальною глибиною 15 та швидкістю навчання 0,05. Модель досягла ROC-AUC 87,3%, Precision 94,7%, Recall 77,5% та PR-AUC 97,5%.

Середня латентність становила 0,39 мс при часі навчання 8,3 секунди на 1М записів.

Важливим відкриттям стала необхідність калібрації порогу класифікації для LightGBM. Аналіз розподілу передбачених ймовірностей показав, що модель видає консервативні оцінки, тому стандартний поріг 0.50 є занадто високим. При оптимальному порозі $t=0.40$ LightGBM досягає значно кращих результатів: Accuracy 86,2%, Recall 91,3%, F1-Score 91,9% при незначному зниженні Precision до 92,4%.

Порівняльний аналіз виявив значні відмінності у характеристиках моделей. Random Forest демонструє найвищий Recall (97.4%), що означає кращу здатність виявляти успішні транзакції з мінімумом пропусків. LightGBM з оптимальним порогом досягає порівняного Recall (91.3%) при вищому ROC-AUC (87.3% проти 86.9%) та значно меншій латентності.

За критерієм придатності до production-середовища LightGBM з порогом $t=0.40$ має значні переваги: латентність 0.39 мс дозволяє виконувати inference у реальному часі без впливу на загальний час обробки транзакції, а час перенавчання 8.3 секунди на повному датасеті уможлиблює часте оновлення моделі при надходженні нових даних. Random Forest рекомендовано для сценаріїв з підвищеними вимогами до Recall (VIP-клієнти, великі суми) або як резервну модель.

Таблиця 4.2– Зведена таблиця результатів показників тестування моделей

Метрика	Random Forest	LighGBM	Вимога
1	2	3	4
ROC-AUC	86,9%	87,3%	85-92%
PR-AUC	97,4%	97,5%	-

Продовження таблиці 4.2

1	2	3	4
Accuracy	89,7%	86,2%	-
Precision	91,1%	92,4%	-
Recall	97,4%	91,3%	-
F1-Score	94,2%	91,9%	-
MCC	0,53	0,41	-
Latency	19,2 мс	0,39 мс	<20 / <12 мс

4.2.2 Аналіз впливу параметрів та SHAP-інтерпретація

Ефективність системи маршрутизації визначається не лише якістю ML-моделі, але й параметрами системи прийняття рішень та правильним розумінням факторів, що впливають на успішність транзакцій.

SHAP-аналіз виявив домінуючу роль ознаки `3ds_outcome` (SHAP importance 4.40), що відображає результат 3D Secure верифікації. Це підтверджує критичну важливість процесу автентифікації для успішності платежу. Друга за важливістю ознака – `is_3ds` (0.74), що вказує на загальний факт використання 3D Secure. Разом ці дві ознаки формують понад 80% сумарної SHAP importance.

Серед інших значущих факторів виявлено: `mcc_target_encoded` (0.12), що підтверджує ефективність target encoding для категорії мерчанта; `is_cross_border` (0.12), що відображає підвищений ризик міжнародних транзакцій; `is_tokenized` (0.07), що свідчить про важливість токенизації для безпеки; `card_country_encoded` (0.07) та `gateway_fee_percent` (0.06).

Розмір вікна агрегації визначає період, за який обчислюються історичні показники успішності шлюзів. У датасеті використано два вікна: 1 година

(gateway_success_1h) та 24 години (gateway_success_24h). Конструювання ознак додатково створює gateway_sr_ratio як співвідношення цих показників. SHAP-аналіз показав помірну важливість velocity_1h (0.03), що підтверджує цінність короткострокових сигналів.

Коефіцієнт згасання визначає швидкість зменшення впливу старих спостережень на поточні оцінки. При експоненціальному згасанні з коефіцієнтом λ вплив спостереження, зробленого t періодів тому, зменшується як $\exp(-\lambda t)$. У контексті платіжних шлюзів рекомендовано значення $\lambda = 0.1-0.3$, що відповідає напівперіоду 2-7 годин.

Рівень дослідження (exploration rate, ϵ) визначає частку транзакцій, що направляються на неоптимальні шлюзи для збору інформації. Рекомендовано $\epsilon = 5-10\%$ з поступовим зменшенням у міру накопичення даних.

Додатково досліджено вплив гіперпараметрів моделей. Для Random Forest критичними є кількість дерев (400 оптимально) та максимальна глибина (30). Для LightGBM – швидкість навчання (0.05), кількість листів (31), subsample (0.8) та параметри регуляризації (reg_lambda = 0.1). Крос-валідація підтвердила стабільність результатів: ROC-AUC на валідації становив 86.4% для RF та 86.2% для LightGBM, що близько до тестових показників [14].

4.2.3 A/B тестування: ML vs Rule-based

Фінальна валідація ефективності ML-моделі потребує проведення A/B тестування на реальному трафіку. Офлайн-оцінка на історичних даних має обмеження: неможливо достовірно оцінити результат транзакції через альтернативний шлюз, оскільки кожна транзакція оброблялась лише одним провайдером.

Дизайн A/B тесту передбачає випадковий розподіл вхідного трафіку на дві групи: контрольну (A) з rule-based маршрутизацією та експериментальну (B) з

ML-маршрутизацією. Рекомендований розподіл – 50/50 для максимальної статистичної потужності або 90/10 як рекомендовано у [4] для мінімізації ризику на початковому етапі.

Метрики оцінки включають первинну метрику (Success Rate) та вторинні метрики: середній час обробки, розподіл по шлюзах, загальну вартість комісій та частку soft decline. Статистична значущість оцінюється за допомогою z-тесту для порівняння пропорцій з рівнем значущості $\alpha = 0.05$.

Мінімальний розмір вибірки для виявлення ефекту 0.5 п.п. (з 85% до 85.5%) при потужності 80% становить приблизно 150 000 транзакцій на групу. За типового трафіку 100 000 транзакцій на день тест потребуватиме 3-4 дні для досягнення статистичної значущості.

Симуляція на тестовому наборі показала, що ML-моделі рекомендують шлюз, відмінний від поточного, у 94% випадків. Це свідчить про значний потенціал оптимізації, проте реальний uplift може відрізнятись через зміну поведінки шлюзів при зміні розподілу трафіку. Rule-based baseline (завжди Trustly) забезпечує SR 90.06%, що є цільовим показником для перевершення ML-системою [6].

Рекомендована стратегія впровадження передбачає поетапний підхід. На першому етапі ML-модель працює у shadow mode, формуючи рекомендації без впливу на реальну маршрутизацію. Порівняння рекомендацій з фактичними рішеннями дозволяє виявити систематичні відмінності. На другому етапі проводиться A/B тест з малою часткою трафіку (5-10%). На третьому етапі, за умови позитивних результатів, частка ML-маршрутизації поступово збільшується до 100% з постійним моніторингом метрик [8].

Критерії зупинки тесту включають: статистично значуще зниження SR більше ніж на 0.5 п.п., аномальне зростання часу відповіді або концентрація трафіку на одному шлюзі понад 80%. За наявності таких індикаторів система автоматично повертається до rule-based маршрутизації.

4.3 Оцінка ефективності для IT-проєкту

Розроблена система демонструє високу технічну ефективність за ключовими показниками. Латентність inference LightGBM (0,39 мс) на два порядки менша за типовий час обробки транзакції платіжним шлюзом (100-500 мс), що робить ML-компонент практично непомітним у загальному потоці обробки. Стабільність латентності підтверджується низьким стандартним відхиленням (0,06 мс) та прийнятним 99-м перцентилем (0,70 мс).

Час перенавчання моделі (8,3 секунди для LightGBM на 1М записів) дозволяє реалізувати онлайн-навчання з оновленням моделі кожні 5-15 хвилин без впливу на продуктивність системи. Це критично важливо для адаптації до змін у поведінці платіжних шлюзів [13].

Конструювання ознак та target encoding забезпечили компактну репрезентацію даних (52 ознаки для LightGBM замість 171 при one-hot encoding), що зменшує споживання пам'яті та прискорює inference.

Економічний ефект від впровадження ML-маршрутизації формується з кількох джерел. Підвищення Success Rate безпосередньо збільшує виручку, оскільки кожна успішна транзакція генерує дохід від комісії. За консервативною оцінкою, підвищення SR на 1 процентний пункт при обсязі 1 мільйон транзакцій на місяць та середній комісії 2% від суми транзакції 50 USD дає додатковий дохід приблизно 10 000 USD на місяць.

Зменшення кількості повторних спроб (retry) знижує навантаження на інфраструктуру та зменшує витрати на комісії шлюзів за невдалі транзакції. Типова комісія за невдалу транзакцію становить 0,05-0,15 USD, що при 15% невдалих транзакцій та 1 мільйоні спроб складає 7 500-22 500 USD щомісячних витрат

Оптимізація розподілу трафіку між шлюзами дозволяє отримати кращі умови від провайдерів та зменшити загальну вартість обробки. Різниця в

комісіях між шлюзами може сягати 0,5-1%, що при оптимальному розподілі дає економію 2 500-5 000 USD на мільйон транзакцій.

Сукупний економічний ефект оцінюється у 20 000-37 500 USD на місяць на мільйон транзакцій, що забезпечує окупність витрат на розробку та впровадження протягом 2-4 місяців [7].

Автоматизація вибору шлюзу зменшує потребу в ручному налаштуванні правил маршрутизації. Традиційний підхід потребує 2-4 години роботи аналітика щотижня для аналізу статистики та коригування правил. ML-система виконує цю роботу автоматично з вищою точністю.

Система забезпечує швидку адаптацію до інцидентів: при деградації показників шлюзу модель автоматично зменшує трафік на нього протягом 1-2 годин (залежно від вікна агрегації), тоді як ручне реагування зазвичай потребує 4-8 годин.

SHAP-аналіз забезпечує прозорість прийняття рішень, що важливо для аудиту та відповідності регуляторним вимогам (PCI DSS, GDPR). Можливість пояснити, чому конкретна транзакція була направлена на певний шлюз, є критичною для відповідності.

Основні ризики впровадження включають: можливість деградації моделі при зміні розподілу даних (дрейф), залежність від якості вхідних даних та потенційні проблеми з інтерпретованістю рішень. Ці ризики мінімізуються завдяки моніторингу метрик у реальному часі, механізму автоматичного відкату на rule-based систему та регулярному перенавчанню моделі.

4.4 Рекомендації щодо впровадження у production

Для production-середовища рекомендовано LightGBM з порогом класифікації $t=0.40$ як основну модель завдяки оптимальному балансу якості та швидкодії. Латентність 0.39 мс дозволяє виконувати inference синхронно у

потоці обробки транзакції без помітного впливу на загальний час відповіді. При оптимальному порозі модель досягає Accuracy 86,2%, Recall 91,3%, F1-Score 91,9%, ROC-AUC 87,3% та PR-AUC 97,5%. [11,13].

Random Forest рекомендовано для сценаріїв з підвищеними вимогами до Recall (97.4%) – наприклад, для VIP-клієнтів або транзакцій з високими сумами, де пропуск успішної транзакції особливо критичний.

Для критичних сегментів трафіку доцільно використовувати ensemble з обох моделей з агрегацією передбачень методом усереднення ймовірностей.

Система моніторингу повинна відстежувати три категорії метрик. Операційні метрики включають латентність inference (mean, p50, p95, p99), throughput (requests per second), error rate та доступність сервісу. Метрики якості моделі охоплюють розподіл передбачених ймовірностей, частку позитивних передбачень, реальний Success Rate по когортах та SHAP importance для виявлення drift. Метрики data drift відстежують зміни у розподілі вхідних ознак порівняно з навчальними даними [4,11].

Рекомендовано налаштувати алерти при: збільшенні латентності p99 понад 5 мс, зміні частки позитивних передбачень більше ніж на 5 п.п., виявленні значущого data drift (KL-divergence > 0.1) або зниженні реального SR більше ніж на 1 п.п.

Перенавчання моделі рекомендовано виконувати щоденно на даних за останні 30 днів з експоненціальним зважуванням (більша вага новішим спостереженням). Автоматичне А/В тестування нової моделі проти поточної дозволяє валідувати покращення перед повним впровадженням.

Впровадження рекомендовано здійснювати поетапно протягом 8-12 тижнів. На першому етапі (тижні 1-2) виконується інтеграція ML-сервісу у shadow mode з логуванням передбачень без впливу на реальну маршрутизацію. Аналіз розбіжностей між ML-рекомендаціями та поточними рішеннями дозволяє виявити потенційні проблеми.

На другому етапі (тижні 3-4) запускається A/B тест з 5-10% трафіку. Моніторинг Success Rate, латентності та розподілу по шлюзах. За умови позитивних результатів частка трафіку збільшується до 25%.

На третьому етапі (тижні 5-8) частка ML-маршрутизації поступово збільшується до 50%, потім до 75% з постійним моніторингом метрик. Кожне збільшення супроводжується аналізом статистичної значущості результатів.

На фінальному етапі (тижні 9-12) ML-система стає основною з rule-based fallback. Документується runbook для операційної підтримки, налаштовуються автоматичні процеси перенавчання та моніторингу.

Мінімальні вимоги для розгортання у систему включають: 2-4 vCPU та 4-8 GB RAM на екземпляр ML-сервісу, SSD-сховище для моделі та логів, мережевий канал з латентністю до основного сервісу менше 1 мс. Для обробки пікового навантаження 10 000 транзакцій на секунду достатньо 4-6 екземплярів сервісу з горизонтальним масштабуванням [15].

Рекомендовано використання контейнеризації (Docker) та оркестрації (Kubernetes) для спрощення розгортання та масштабування. CI/CD pipeline повинен включати автоматичне тестування моделі на валідаційному наборі перед розгортанням.

4.5 Висновок по четвертому розділу

Проведені експериментальні дослідження продемонстрували високу ефективність розробленої ML-системи маршрутизації платіжних транзакцій як з технічної, так і з економічної точок зору.

На основі синтетичного набору даних з 1 мільйона транзакцій моделей Random Forest та LightGBM було реалізовано та оцінено. LightGBM показав виняткові результати для production-середовища: ROC-AUC 87,3%, PR-AUC 97,5% при латентності inference всього 0,39 мс, що на два порядки нижче, ніж

типовий час обробки платіжним шлюзом (100-500 мс). Random Forest, у свою чергу, продемонстрував найвищий Recall 97,4%, що робить його ідеальним для критичних сегментів, де неприйнятна втрата потенційно успішних транзакцій.

Ключовий вклад у підвищення якості моделей внесли конструювання ознак, Target Encoding з Bayesian smoothing, двоетапна процедура відбору ознак та циклічне кодування часових параметрів. SHAP-аналіз виявив домінуючу роль результату 3D Secure верифікації (SHAP importance 4,40), що підтверджує важливість процесу автентифікації для успішності платежу.

Економічна оцінка свідчить про значний потенціал впровадження. Консервативні розрахунки показують накопичений економічний ефект 20 000–37 500 USD на місяць при обсязі 1 мільйон транзакцій, що забезпечує окупність витрат на розробку та впровадження протягом 2–4 місяців. Додатково система забезпечує автоматизацію вибору шлюзу, що розвантажує аналітиків від щотижневого ручного налаштування правил, та швидку адаптацію до інцидентів через автоматичне переспрямування трафіку за 1–2 години замість традиційних 4–8 годин.

Розроблений метод інтеграції повністю відповідає операційним вимогам з точки зору латентності та пропускнуої спроможності. Механізм автоматичного відкату на rule-based систему, поряд з постійним моніторингом метрик та регулярним перенавчанням моделей, мінімізує ризики від data drift та зміни поведінки платіжних шлюзів.

Рекомендована поетапна стратегія впровадження (shadow mode → A/B тест → поступове масштабування → full rollout) протягом 8–12 тижнів дозволяє валідувати гіпотези та мінімізувати виробничі ризики. SHAP-аналіз забезпечує необхідну прозорість прийняття рішень для відповідності регуляторним вимогам (PCI DSS, GDPR), що є критичним для фінтех-проектів.

ВИСНОВКИ

У ході кваліфікаційної роботи проведено комплексне дослідження методів інтеграції машинного навчання в інформаційні системи обробки платіжних транзакцій у фінтех-компаніях. Основну увагу зосереджено на розробленні та обґрунтуванні підходу до оптимізації маршрутизації платежів за допомогою сучасних алгоритмів ML у контексті специфіки та обмежень існуючих платіжних платформ.

Виявлено, що існуючі підходи до маршрутизації у платіжних системах базуються переважно на статичних правилах з показником успішності близько 92%, що не адаптуються до змін умов обробки платежів у реальному часі. Це створює методологічний розрив між можливостями сучасних аналітичних методів та фактичним рівнем автоматизації платіжних систем у галузі.

Розроблено класифікацію існуючих методів машинного навчання за критеріями придатності для маршрутизації платежів. Порівняльний аналіз охопив п'ять основних підходів:

- статичні правила (базовий рівень, ROC-AUC 0.5, латентність < 1 мс);
- логістична регресія (ROC-AUC близько 0.89, латентність 5 мс);
- Random Forest (ROC-AUC 0.869, латентність 19.2 мс, Recall 97.4%);
- градієнтний бустинг (XGBoost, LightGBM) (ROC-AUC 0.87–0.92, латентність 0.39 мс для LightGBM);
- нейронні мережі (ROC-AUC 0.93–0.94, латентність 30–50 мс, проблеми з production-готовністю).

На основі аналізу обрано LightGBM як оптимальний алгоритм для production-середовища завдяки найменшій латентності при високій точності та можливості онлайн-адаптації.

Було зроблено формалізацію задачі маршрутизації у контексті non-stationary multi-armed bandits із затримкою зворотного зв'язку. Розроблено математичне описання ймовірності успіху платежу на кожному провайдері через

композицію статичних та динамічних компонентів (формули 3.1–3.4).

Запропонована архітектура HYBRID-ROUTE інтегрує статичні бізнес-правила з динамічними ML-прогнозами. Експериментальна валідація проведена на синтетичному наборі даних обсягом 1 млн транзакцій з стратифікованим часовим розділенням (70/15/15%)

За результатами кваліфікаційної роботи підготовлено до друку статтю «Fintech Transaction Routing via a Dynamically Adapted LightGBM» авторів Mukha S., Mikhnova O.

Пояснювальна записка до кваліфікаційної роботи оформлена згідно ДСТУ та методичними вказівками щодо розробки та оформлення кваліфікаційної роботи [16-18].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Розвиток фінтеху в Україні [Електронний ресурс] / Національний банк України. URL: <https://bank.gov.ua/ua/about/develop-strategy/fintech2025> (дата звернення: 07.12.2025).
2. Cao L., Yang Q., Yu P. S. Data science and AI in FinTech: An overview. *International Journal of Data Science and Analytics*. 2021. Vol. 12, No. 2. P. 81–99. DOI: 10.1007/s41060-021-00278-w.
3. Thomas G. Breakthrough Advancements in Real-time Payment Processing Optimization. *European Journal of Computer Science and Information Technology*. 2025. Vol. 13, No. 17. P. 39–47. DOI: 10.37745/ejcsit.2013/vol13n173947.
4. Chaudhary A., Rai A., Gupta A. Maximizing Success Rate of Payment Routing using Non-stationary Bandits. *Proceedings of the 3rd International Conference on AI-ML Systems (AIMLSystems '23)*. New York : ACM, 2023. Article 27. P. 1-7. DOI: 10.1145/3639856.3639883.
5. Penaganti R. Building a Universal Payment Platform: Breaking Free from Payment Gateway Lock-in. *Journal of International Crisis and Risk Communication Research*. 2025. P. 90–107. DOI: 10.63278/jicrcr.vi.3416.
6. Ciobotea C. Improving payment authorization rates by intelligently routing transactions. Master's thesis. Enschede : University of Twente, 2016. – URL: <http://essay.utwente.nl/70977/> (дата звернення: 07.12.2025).
7. Saride K. C. AI and Machine Learning in Payment Systems: Unlocking Higher Approval Rates and Lower Fees. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2025. Vol. 11, No. 2. P. 858–877. DOI: 10.32628/CSEIT25112433.
8. Agrawal A., Patil H. A Control-Theoretic Approach to Dynamic Payment Routing for Success Rate Optimization. *Proceedings of the 3rd International Conference on AI-ML Systems (AIMLSystems '23)*. New York : ACM, 2023. Article 8. P. 1–6. DOI: 10.48550/arXiv.2510.16735.

9. Ramesh G., Vamsi K. Enhancing Payment Ecosystems with AI/ML: Real-Time Analytics for Fraud Prevention and User Insights. *World Journal of Advanced Research and Reviews*. 2025. Vol. 25, No. 1. P. 1273–1281. DOI: 10.30574/wjarr.2025.26.1.1273.

10. Bygari R., Gupta A., Raghuvanshi S., Bapna A., Sahu B. An AI-powered Smart Routing Solution for Payment Systems. 2021 IEEE International Conference on Big Data (Big Data). Orlando : IEEE, 2021. P. 2026–2033. DOI:10.48550/arXiv.2111.00783.

11. Makin Y., Gondhi P. K. Adaptive Payment Routing in E-Commerce: A Machine Learning Approach for Higher Purchase Success Rates. *International Journal of Applied Mathematics*. 2025. Vol. 38, No. 2. P. 109–122. DOI: 10.12732/ijam.v38i2s.73.

12. Yeh C.-C. M., Zhuang Z., Wang J., Zheng Y., Ebrahimi J., Mercer R., Wang L., Zhang W. Online Multi-horizon Transaction Metric Estimation with Multi-modal Learning in Payment Networks. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21)*. New York : ACM, 2021. P. 4331–4340. DOI: 10.1145/3459637.3481942.

13. Shahapurkar A., Patil R. Concept drift and machine learning model for detecting fraudulent transactions in streaming environment. *International Journal of Electrical and Computer Engineering (IJECE)*. 2023. Vol. 13, No. 5. P. 5560–5568. DOI: 10.11591/ijece.v13i5.pp5560-5568.

14. Lin, H., Chung, J. W., Lao, Y., Zhao, W. Online Gradient Boosting Decision Tree: In-Place Updates for Efficient Adding/Deleting Data. *arXiv*. 2025. DOI: 10.48550/arXiv.2502.01634.

15. Singh, P. Modernizing Legacy FinTech Systems through Cloud-Native Microservices Architecture. *International Journal on Science and Technology*. Vol. 15, no. 1, pp. 1–15. 2024. DOI: 10.71097/IJSAT.v15.i1.6070.

16. ДСТУ 3008:2015 «Звіти у сфері науки і техніки. Структура та правила оформлення». Київ: Держстандарт України, 2017. 31 с.

17. ДСТУ 8302:2015 «Бібліографічне посилання. Загальні положення та

правила складання». Київ: Держстандарт України, 2017. 20 с.

18. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня програми «Інформаційні управляючі системи та технології») / Упоряд.: Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. Харків: ХНУРЕ, 2021. 30 с.