УДК 004.89:336

С. О. САВЧЕНКО
Хмельницький національний університет

МЕТОДОЛОГІЧНІ ПІДХОДИ ДО ПЕРСОНАЛІЗАЦІЇ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

У статті систематизовано методологічні підходи до персоналізації великих мовних моделей (LLM) в контексті їх практичного застосування у галузі фінансових технологій. Актуальність дослідження зумовлена активним впровадженням використання LLM у реальних продуктах, таких як державні ШІ-асистенти, банківські чат-боти, системи автоматизованого фінансового консультування та автоматизовані агенти з обробки фінансових документів. При цьому базові LLM загального призначення не можуть бути ефективно використані для вузько-галузевих задач через відсутність доступу до актуальних корпоративних даних, незнання специфічної нормативно-правової бази та неможливість виконання дій у зовнішніх системах. У даній роботі систематизовано методи персоналізації, які дозволяють подолати ці обмеження. Проаналізовано п'ять ключових методів адаптації LLM до конкретних бізнес-задач: управління контекстом та системний промпт – найпростіший і найменш затратний спосіб задати роль, стиль та обмеження поведінки моделі без додаткового навчання; інжиніринг промптів, включаючи техніки ланцюжка думок (CoT) та дерева думок (ToT), що суттєво підвищують точність логічного міркування при вирішенні складних аналітичних фінансових задач; доповнене пошуком генерування (RAG), що представляє собою архітектурне рішення для динамічного доступу до актуальних даних із зовнішніх джерел; тонке налаштування (Fine-tuning), зокрема параметрично-ефективний метод LoRA (Low-Rank Adaptation), який при мінімальних обчислювальних витратах дозволяє адаптувати LLM до вузькоспеціалізованих задач, таких як класифікації фінансових документів, аналізу тональності новин, кредитного скорингу тощо; виклик функцій (Function Calling) та ШІ-агенти, що забезпечують двосторонню інтеграцію LLM із зовнішніми API, CRM- та ERP-системами. Проведено порівняльний аналіз зазначених методів за п'ятьма критеріями: вартість впровадження, вимоги до навчальних даних, актуальність знань, можливості інтеграції із зовнішніми системами та прозорість сформованих відповідей.

За результатами проведеного аналізу визначено, що для інтеграції ШІ-систем у фінансові продукти оптимальною є гібридна архітектура, яка поєднує використання RAG та Fine-tuning. Емпіричні дані свідчать, що кожен із цих методів позитивно впливає на точність відповіді, а їх поєднання є найбільш ефективним. Сформульовано конкретні рекомендації щодо вибору методу персоналізації залежно від типу прикладної задачі.

Ключові слова: *великі мовні моделі, персоналізація LLM, RAG, тонке налаштування, LoRA, інжиніринг промптів, FinTech, ШІ-агенти.*

S. O. SAVCHENKO
Khmelnytskyi National University

METHODOLOGICAL APPROACHES TO PERSONALIZATION IN LARGE LANGUAGE MODELS

The article systematizes methodological approaches to the personalization of large language models (LLMs) within the context of their practical application in the field of financial technology. The relevance of this research is driven by the active implementation of LLMs in real-world products, such as government AI assistants, banking chatbots, automated financial advisory systems, and automated agents for processing financial documents. At the same time, base general-purpose LLMs cannot be effectively utilized for niche industry tasks due to a lack of access to up-to-date corporate data, ignorance of specific regulatory frameworks, and an inability to perform actions in external systems. This paper systematizes personalization methods that allow these limitations to be overcome. Five key methods for adapting LLMs to specific business tasks are analyzed: context management and the system prompt – the simplest and most cost-effective way to define the model's role, style, and behavioral constraints without additional training; prompt engineering, including Chain-of-Thought (CoT) and Tree-of-Thought (ToT) techniques, which significantly improve the accuracy of logical reasoning when solving complex analytical financial tasks; Retrieval-Augmented Generation (RAG), which represents an architectural solution for dynamic access to current data from external sources; Fine-tuning, specifically the parameter-efficient LoRA (Low-Rank Adaptation) method, which allows LLMs to be adapted to highly specialized tasks such as financial document classification, news sentiment analysis, and credit scoring with minimal computational costs; and Function Calling and AI agents, which provide two-way integration of LLMs with external APIs, CRM, and ERP systems. A comparative analysis of these methods was conducted based on five criteria: implementation cost, training data requirements, knowledge currency, integration capabilities with external systems, and the transparency of the generated responses.

https://doi.org/10.32782/mathematical-modelling/2026-9-1-31 ISSN 2618-0332 (print), 2618-0340 (online)

Based on the analysis conducted, it has been determined that a hybrid architecture combining the use of RAG and Fine-tuning is optimal for integrating AI systems into financial products. Empirical data indicate that each of these methods has a positive impact on response accuracy, and their combination proves to be the most effective. Specific recommendations have been formulated regarding the choice of personalization method depending on the type of application task. Based on the analysis conducted, it has been determined that a hybrid architecture combining the use of RAG and Fine-tuning is optimal for integrating AI systems into financial products. Empirical data indicate that each of these methods has a positive impact on response accuracy, and their combination proves to be the most effective. Specific recommendations have been formulated regarding the choice of personalization method depending on the type of application task.

Keywords: *large language models, LLM personalization, RAG, fine-tuning, LoRA, prompt engineering, FinTech, AI agents.*

Постановка проблеми

Упродовж останніх років великі мовні моделі, або ж LLM (Large Language Model), перетворилися з суто академічного об'єкта дослідження на практичний інструмент, що активно впроваджується у комерційні та державні продукти. Приватні компанії та державні установи приступили до активного впровадження чат-ботів та ШІ-агентів у своїх продуктах. Наприклад, банки інтегрують інтелектуальних консультантів для обслуговування клієнтів, медичні заклади впроваджують ШІ-асистентів для підтримки пацієнтів, страхові компанії будують автоматизовані системи оцінки ризику, а ритейлери – персоналізовані системи рекомендацій товарів та послуг.

У розрізі впровадження ШІ в державному секторі показовим є досвід України. В 2025 році у застосунку «Дія» реалізовано Дія.AI – першого у світі державного ШІ-асистента, визнаного переможцем конкурсу SEMIC Best Cases Award 2025 у категорії «Government-to-Citizens» Європейської Комісії [1]. Система демонструє показник задоволеності CSAT понад 80 %, здатна скоротити час вирішення питань громадян до 3 хвилин та, за прогнозами, зекономить понад 1 мільйон годин громадян на рік [1]. Дія.AI побудована на основі комерційної LLM від Google (Gemini 2.0 Flash) у поєднанні з власними фільтрами безпеки та інтеграцією з державними реєстрами [2]. Наразі схожі ініціативи реалізуються й у інших країнах. Уряд США активно використовує чат-боти в державних агенціях для взаємодії з громадянами [3]. Банківська галузь впроваджує ШІ для персоналізованого фінансового обслуговування [4].

Ключовою технологічною проблемою залишається те, що готові LLM-моделі загального призначення не можуть безпосередньо застосовуватися для вирішення специфічних задач конкретних бізнес-областей. Базові моделі не мають доступу до актуальних внутрішніх даних компаній, незнайомі зі специфічними регуляторними вимогами галузі (наприклад, міжнародними стандартами фінансової звітності (МСФЗ) або ж нормативами Національного банку України) і нездатні виконувати дії у зовнішніх системах (CRM, ERP, банківська інфраструктура, тощо). Саме тому для якісної інтеграції ШІ у промислові продукти (перш за все мова іде про чат-боти та агенти, побудовані на основі LLM) необхідно реалізувати взаємодію LLM із зовнішніми джерелами інформації та сервісами, застосовуючи методи персоналізації. Систематизація таких методів набуває особливої актуальності у юридичній, медичній та фінансовій (FinTech) сферах.

Аналіз останніх досліджень і публікацій

Теоретичним підґрунтям більшості сучасних LLM слугує архітектура Transformer, представлена розробниками компанії Google у 2017 році [5]. Вони запропонували механізм уваги (attention), який дозволив масштабувати мовні моделі до мільярдів параметрів без принципових архітектурних змін. Подальший розвиток мовних моделей представила робота [6], в якій було запропоновано механізм BERT (Bidirectional Encoder Representations from Transformers). Механізм моделі BERT полягає у двонаправленому контекстному аналізі тексту за допомогою архітектури трансформера, де кожне слово інтерпретується з урахуванням усіх слів у реченні одночасно. В результаті було встановлено парадигму «попереднє навчання – тонке налаштування». Наступний якісний та кількісний крок у розвитку великих мовних моделей було представлено у дослідженні [7], яке присвячено моделі GPT-3 зі 175 мільярдами параметрів. GPT-3

https://doi.org/10.32782/mathematical-modelling/2026-9-1-31 ISSN 2618-0332 (print), 2618-0340 (online)

довела здатність LLM до виконання завдань у режимі «кількох прикладів» (few-shot learning) без оновлення параметрів. Подальший розвиток у придатності до практичного застосування здійснено дослідниками [8], які запровадили навчання на основі зворотного зв'язку від людей – RLHF (Reinforcement Learning from Human Feedback) та представили модель InstructGPT. Відкриті моделі серії LLaMA, описані у роботі [9], значно спрощують прикладні дослідження у цій сфері, адже моделі сімейства LLaMA, розроблені компанією Meta, мають відкриті ваги та можуть бути розгорнутими на власній інфраструктурі.

У питанні адаптації LLM до конкретних прикладних задач однією з ключових є робота [10], в якій систематизовано методи інжинірингу промптів. У дослідженні [11] представлено техніку ланцюжка думок (Chain-of-Thought, CoT), що суттєво покращує логічне «міркування» моделі. Доповнене пошуком генерування (Retrieval-Augmented Generation, RAG) вперше було запропоноване в роботі [12]. У сфері тонкого налаштування моделей (fine-tuning) важливою є робота [13], в якій автори представили метод донавчання великих моделей LoRA, який замість зміни всіх ваг моделі додає невеликі низькорангові матриці до окремих її шарів і навчає лише їх.

Специфіку застосування LLM у фінансовій галузі було досліджено в роботі [14] (представлено модель BloombergGPT) та у комплексному огляді [15]. Незважаючи на значний обсяг наявних досліджень, у літературі бракує систематизованого підходу до вибору методу персоналізації LLM залежно від типу прикладної задачі, зокрема для FinTech галузі.

Мета дослідження

Метою даного дослідження є систематизація методологічних підходів до персоналізації великих мовних моделей, аналіз їх архітектурних особливостей та практичних характеристик, а також формулювання рекомендацій щодо вибору оптимального методу персоналізації LLM залежно від типу задачі у сфері FinTech.

Виклад основного матеріалу дослідження

1. Архітектура та еволюція великих мовних моделей.

Сучасні великі мовні моделі базуються на архітектурі Transformer, запропонованій у 2017 році [5]. Ключовим аспектом даної архітектури є механізм самоуваги (self-attention), що дозволяє моделі динамічно зважувати відносну важливість кожного токена (частини слова) у послідовності відносно всіх інших при формуванні контекстного представлення. На відміну від рекурентних нейронних мереж, Transformer обробляє всю послідовність паралельно, що відкрило можливості для масштабного навчання на великих корпусах даних.

Архітектура Transformer у базовому вигляді включає два основні компоненти: кодувальник (encoder) та декодувальник (decoder). Моделі типу BERT [6] використовують лише кодувальник (encoder-only), що забезпечує контекстуалізоване двонаправлене представлення тексту, тоді як GPT [7] та LLaMA [9] базуються виключно на декодувальнику (decoder-only) і реалізують авторегресивну генерацію. Моделі декодувального типу наразі домінують у задачах генерації природної мови завдяки здатності формувати зв'язний текст довільної довжини.

Якісний стрибок у практичній придатності LLM відбувся із запровадженням механізму навчання з підкріпленням на основі зворотного зв'язку від людини (RLHF), описаного в роботі [8]. Модель InstructGPT розміром 1,3 мільярди параметрів перевершила GPT-3 (175 мільярдів параметрів) за оцінками користувачів. Це свідчить, що узгодження моделі з людськими перевагами є не менш важливим, ніж кількість параметрів. Відкриті моделі серії LLaMA [9] суттєво розширили доступ до досліджень і прикладних розробок. Вони зробили можливим тонке налаштування LLM навіть за обмежених обчислювальних ресурсів без обов'язкового використання масштабної хмарної інфраструктури.

Проте жодна з базових LLM не може бути використана для вузько-спеціалізованих задач. Ключові обмеження можна об'єднати у наступні категорії: «галюцинації» (генерація неіснуючих

https://doi.org/10.32782/mathematical-modelling/2026-9-1-31 ISSN 2618-0332 (print), 2618-0340 (online)

фактів), застарілість знань (навчальні дані обмежені часом навчання моделі), відсутність доступу до приватних корпоративних даних, неможливість виконання дій у зовнішніх системах. Подолання цих обмежень є основною метою методів персоналізації, розглянутих у наступному розділі.

2. Методи персоналізації великих мовних моделей.

2.1. Управління контекстом та системний промпт. Найпростішим і найменш затратним методом персоналізації є управління вхідним контекстом моделі. Кожен запит до LLM складається із системного промпту – набору інструкцій, що задають роль, стиль та обмеження поведінки моделі, та контексту розмови. Якісно підібраний системний промпт дозволяє без будь-якого додаткового навчання налаштувати модель як спеціалізованого фінансового консультанта, юридичного асистента або оператора служби підтримки клієнтів банку.

Управління контекстом у ширшому розумінні включає формування мало-прикладного навчання (few-shot learning) безпосередньо у промпті: розробник включає у запит кілька прикладів бажаних пар «вхід–вихід», спрямовуючи модель до певного формату відповіді без оновлення її ваг. У роботі [7] показано, що модель здатна виконувати нові задачі за наявності від одного до кількох прикладів, не потребуючи жодного градієнтного оновлення.

Перевагами цього методу є нульова вартість навчання, миттєве розгортання та висока гнучкість. Обмеження ж полягають у залежності від розміру контекстного вікна моделі (від 8 тисяч до 1 мільйона токенів у різних моделей), ймовірність «забування» системних інструкцій при надто довгих розмовах та відсутність стійкого засвоєння нових предметних знань.

2.2. Інжиніринг промптів (Prompt Engineering) є систематичним підходом до конструювання вхідних запитів з метою максимізації якості відповідей LLM без зміни параметрів моделі. Таксономія сучасних технік інжинірингу промптів включає: нульовий приклад (zero-shot), кілька прикладів (few-shot), ланцюжок думок (Chain-of-Thought, CoT), дерево думок (Tree-of-Thought, ToT), ланцюжок перевірки (Chain-of-Verification, CoVe) тощо.

Ключовою технікою для фінансових застосунків є CoT [11]. Включення у промпт прикладів із покроковими міркуваннями значно підвищує точність відповіді при вирішенні складних аналітичних задач. Наприклад, при оцінці ризику кредитної заявки CoT дозволяє моделі явно сформувати аналітичний ланцюжок: аналіз кредитної історії позичальника → розрахунок боргового навантаження → аналіз галузевих ризиків → формування рекомендації. Огляд [10] додатково класифікує техніки «м'якого» промптингу (Prefix Tuning, Prompt Tuning), що навчають лише невеликий набір «м'яких токенів» зі збереженням ваг основної моделі. Такий підхід є проміжним між виключно налаштуванням промпту та повноцінним тонким налаштуванням.

2.3. Доповнене пошуком генерування (RAG) є архітектурним рішенням, яке допомагає подолати проблему застарілості знань та відсутності доступу до приватних корпоративних даних. В оригінальній роботі [12] RAG визначається як гібридна система: нейронний пошуковий компонент (retriever) динамічно знаходить релевантні фрагменти з зовнішньої бази знань, які далі передаються разом із запитом у LLM-генератор (generator). Схема роботи інформаційної системи з RAG представлена на рисунку 1.

Типова RAG-система для FinTech включає два ключові етапи. Перший етап – офлайн-індексування: корпоративні документи (фінансові звіти, нормативні акти НБУ, продуктові описи) перетворюються на векторні представлення (embeddings) та зберігаються у векторній базі даних. Другий етап – обробка запиту в режимі реального часу. Запит користувача векторизується, виконується семантичний пошук у базі, знайдені фрагменти доповнюються оригінальним запитом та передаються в LLM для генерації відповіді з явним посиланням на джерела.

У дослідженій літературі виокремлюються три покоління RAG: Naive RAG (базовий пошук і генерація), Advanced RAG (оптимізація до- та після-пошуку: перевпорядкування результатів, переформулювання запиту, гіпотетичні ембедінги HyDE) та Modular RAG (гнучка композиція спеціалізованих модулів). Для фінансових застосунків Advanced RAG є мінімально прийнятним рівнем, адже базовий пошук дає незадовільну точність при роботі зі складними

https://doi.org/10.32782/mathematical-modelling/2026-9-1-31 ISSN 2618-0332 (print), 2618-0340 (online)

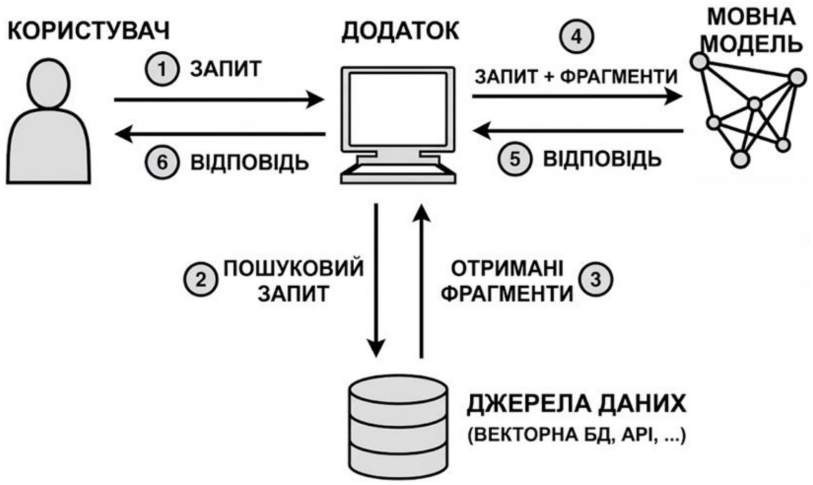

Рис. 1. Архітектура системи доповненого пошуком генерування (RAG)

фінансовими документами. Ключові переваги використання RAG у фінансовій галузі полягають у можливості роботи з актуальними ринковими даними без перенавчання моделі, прозорості відповідей через явне зазначення джерел та відповідності регуляторним вимогам.

2.4. Тонке налаштування (Fine-tuning) передбачає додаткове навчання LLM на спеціалізованому наборі даних із частковим або повним оновленням параметрів моделі. Повне тонке налаштування вимагає значних обчислювальних ресурсів (для моделей розміром 7 мільярдів параметрів і більше необхідно щонайменше кілька GPU класу NVIDIA A100), тому на практиці частіше використовуються параметрично-ефективні методи тонкого налаштування (Parameter-Efficient Fine-Tuning).

Найпоширенішим PEFT-методом є LoRA (Low-Rank Adaptation), вперше запропонований у роботі [13]. LoRA заморожує ваги базової моделі та додає до цільових шарів пари матриць-адаптерів низького рангу: $\Delta W = BA$, де $B \in \mathbb{R}d \times r$, $A \in \mathbb{R}r \times k$, $r \ll \min(d, k)$. Це значно скорочує кількість параметрів, які донавчаються, порівняно з повним налаштуванням при практично ідентичній якості. PEFT-методи класифікуються за трьома категоріями: адитивні (Adapters, Prefix Tuning), специфікаційні (Sparse Fine-Tuning) та рефакторингові (LoRA та його варіанти QLoRA, AdaLoRA) [13].

Спеціалізовані мовні моделі, що розроблені для використання у фінансовій галузі, такі як BloombergGPT та FinGPT використовують кардинально різні архітектурні підходи навчання [14]. Модель BloombergGPT передбачає навчання 50-мільярдної моделі з нуля на 363 мільярдах фінансових токенів (витрати оцінюються у мільйони доларів США). Модель FinGPT використовує LoRA-адаптацію загальнодоступних моделей для досягнення подібних результатів при витратах у кілька сотень доларів. Тонке налаштування є оптимальним, коли потрібно навчити модель специфічному стилю документів, засвоїти спеціалізовані галузеві терміни або стабільно виконувати вузьку класифікаційну задачу.

2.5. Виклик функцій та ШІ-агенти. Виклик функцій (Function Calling) – це здатність LLM генерувати структуровані запити для взаємодії із зовнішніми інструментами (функціями, API) відповідно до запиту користувача. Сучасні підходи демонструють можливість навчання моделей самостійно визначати, коли і як доцільно викликати зовнішні сервіси, такі як пошукові системи, перекладачі або системи запитань-відповідей, без потреби у масштабній ручній розмітці. Подальший розвиток цього напрямку призвів до створення моделей, оптимізованих для точного виклику великої кількості API, що дозволяє суттєво зменшити рівень галюцинацій і підвищити надійність взаємодії із зовнішніми системами.

Концепція доповнених мовних моделей передбачає інтеграцію трьох основних типів інструментів: інструментів міркування (наприклад, інтерпретаторів коду), інструментів отримання

https://doi.org/10.32782/mathematical-modelling/2026-9-1-31 ISSN 2618-0332 (print), 2618-0340 (online)

інформації (пошукових систем або векторних баз даних) та інструментів дій (запис у CRM, ініціювання фінансових операцій тощо). Подальший розвиток цієї ідеї реалізується у вигляді ШІ-агентів на базі LLM, які включають модулі профілювання користувача, пам'яті, планування та виконання дій. У FinTech це відкриває можливості для створення повноцінних інтелектуальних асистентів. Така система може не лише пояснювати фінансові продукти, а й автоматично перевіряти актуальні умови через API банків, ініціювати операції у внутрішніх системах та забезпечувати завершення повного користувацького сценарію в межах єдиного діалогу.

3. Порівняльний аналіз та рекомендації щодо вибору методу.

Емпіричні дослідження демонструють, що підходи Retrieval-Augmented Generation (RAG) та тонкого налаштування добре доповнюють одне одного. Зокрема, застосування LoRA-адаптації забезпечує приріст точності на рівні близько 6 відсоткових пунктів, тоді як використання RAG додатково підвищує цей показник приблизно на 5 пунктів. Таким чином, комбіноване використання RAG та fine-tuning доцільно розглядати як оптимальний підхід для виробничих FinTech-застосунків, де критично важливими є як актуальність знань, так і стабільна якість відповідей. Для систематизованого порівняння методів персоналізації у таблиці 1 наведено їх характеристики за п'ятьма критеріями, що є ключовими при проєктуванні FinTech-систем.

Таблиця 1

Порівняльна характеристика методів персоналізації LLM

Метод	Вартість впровадження	Вимоги до даних	Актуальність знань	Сторонні інтеграції	Прозорість відповіді
Системний промпт	Мінімальна	Не потрібні	Обмежена контекстним вікном	Відсутні	Середня
Інжиніринг промптів	Мінімальна	Приклади для few-shot	Обмежена контекстним вікном	Відсутні	Середня
RAG	Середня	Корпус документів	Висока (динамічне оновлення)	Частково (читання)	Висока (посилання на джерела)
Fine-tuning (LoRA)	Висока	Тисячі навчальних прикладів	Статична (на момент навчання)	Відсутні	Низька
Function Calling	Середня–висока	Специфікації API	Висока (реальний час)	Присутні (читання і запис)	Середня

На основі проведеного аналізу, в таблиці 2 сформульовано рекомендації щодо вибору методу персоналізації для конкретних типів задач у сфері FinTech. Оглядове дослідження [15] підтверджує, що жодна з фінансових LLM-систем у реальному виробничому середовищі не спирається лише на один метод персоналізації. Домінуючим підходом є гібридні архітектури.

Досвід реальних впроваджень підтверджує запропоновані рекомендації. Дія.AI успішно застосовує гібридну архітектуру: готова LLM (Gemini 2.0 Flash) у поєднанні з системним промптом, фільтрами безпеки та інтеграцією з державними реєстрами через виклик функцій [1; 2]. Такий підхід дозволяє мінімізувати потребу у витратному тонкому налаштуванні, зберігаючи при цьому високу надійність і контрольованість відповідей. Спеціалізовані моделі BloombergGPT та FinGPT [14] демонструють різні підходи до тонкого налаштування у фінансовій сфері: BloombergGPT орієнтується на масштаб і використання власних даних, тоді як FinGPT – на ефективність донавчання за допомогою механізму LoRA та відкритих джерел.

Висновки

У роботі систематизовано п'ять ключових методологічних підходів до персоналізації великих мовних моделей для застосування у сфері FinTech: управління контекстом та

Таблиця 2

Рекомендації щодо вибору методу персоналізації залежно від типу задачі

Тип задачі	Рекомендований метод	Обґрунтування
Чат-бот підтримки клієнтів банку (FAQ, умови продуктів)	Системний промпт + RAG	Актуальна база знань, контрольовані відповіді з посиланням на джерела
Класифікація фінансових новин/аналіз тональності	Fine-tuning (LoRA)	Вузька задача зі стабільним форматом, потреба у засвоєнні галузевої термінології
Robo-advisor (персональний інвестиційний консультант)	RAG + Function Calling	Актуальні ринкові дані + можливість дій у торговому портфелі
Генерація фінансових звітів (МСФЗ, управлінська звітність)	Fine-tuning + системний промпт	Специфічний стиль та структура документів, регуляторна відповідність
ШІ-агент для банківських операцій (платежі, заявки)	Function Calling + RAG	Двостороння інтеграція з банківськими API, доступ до актуальних даних рахунків
Відповіді на регуляторні запити (НБУ, FATF, AML)	RAG	Актуальна нормативно-правова база, обов'язкова прозорість джерел
Загальний фінансовий асистент (перший контакт, broad queries)	Системний промпт + CoT	Мінімальні витрати на впровадження, достатня якість для загальних питань

системний промпт, інжиніринг промптів, доповнена пошуком генерація (RAG), тонке налаштування (Fine-tuning з акцентом на LoRA/PEFT) та виклик функцій (Function Calling). Кожен із методів має чітко визначені переваги, обмеження та сферу застосування.

На основі проведеного дослідження сформульовано наступні висновки. По-перше, жоден з розглянутих методів не є універсальним. Вибір методики персоналізації моделі визначається типом задачі, вимогами до актуальності даних, наявністю навчальних прикладів та допустимими витратами на розробку. Визначено, що для виробничих FinTech-систем оптимальним є комбінований підхід з використанням RAG та Fine-tuning, що забезпечує актуальність знань та стабільну якість відповідей на вузькоспеціалізованих задачах. Механізм виклику зовнішніх функцій є необхідною умовою для побудови повноцінних ШІ-агентів, здатних не лише відповідати на питання, а й виконувати транзакційні дії у банківських та корпоративних системах. Реальний досвід впроваджень (Дія.AI, BloombergGPT, FinGPT) підтверджує, що системи, засновані виключно на готових LLM без методів персоналізації, не підходять для практичного застосування у регульованих галузях.

До перспективних напрямів подальших досліджень належать: розроблення уніфікованих підходів до оцінювання LLM-систем, орієнтованих на застосування у FinTech; удосконалення методів зниження рівня галюцинацій у критично важливих фінансових сценаріях; а також дослідження ефективних підходів до інтеграції RAG і fine-tuning в межах єдиної обчислювально-оптимізованої архітектури, придатної для використання малими та середніми фінансовими установами.

Список використаної літератури

1. European Commission. Diia.AI: Ukraine's national AI agent for government services. Public Sector Tech Watch, 2025. URL: https://interoperable-europe.ec.europa.eu/collection/public-sector-tech-watch/diiaai-ukraines-national-ai-agent-government-services (дата звернення: 25.03.2026)
2. Organisation for Economic Co-operation and Development. Diia AI Assistant. OECD.AI Policy Observatory, 2025. URL: https://oecd.ai/en/dashboards/policy-initiatives/diia-ai-assistant (дата звернення: 25.03.2026)
3. Chen T., Gasco-Hernandez M. Uncovering the results of AI chatbot use in the public sector: Evidence from US state governments. *Public Performance & Management Review*. 2025. Vol. 48, No. 6. P. 1331–1356. DOI: https://doi.org/10.1080/15309576.2024.2389864

4. Sheth J. N., Jain V., Roy G., Chakraborty A. AI-driven banking services: The next frontier for a personalised experience in the emerging market. *International Journal of Bank Marketing*. 2022. Vol. 40, No. 6. P. 1248–1271. DOI: https://doi.org/10.1108/IJBM-09-2021-0449
5. Vaswani A. та ін. Attention is all you need. *Advances in Neural Information Processing Systems*. 2017. Vol. 30. P. 5998–6008. DOI: https://doi.org/10.48550/arXiv.1706.03762
6. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*. 2019. P. 4171–4186. DOI: https://doi.org/10.18653/v1/N19-1423
7. Brown T. B. et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 1877–1901. DOI: https://doi.org/10.48550/arXiv.2005.14165
8. Ouyang L. et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*. 2022. Vol. 35. P. 27730–27744. DOI: https://doi.org/10.48550/arXiv.2203.02155
9. Touvron H. et al. LLaMA: Open and efficient foundation language models. *arXiv*, 2023. DOI: https://doi.org/10.48550/arXiv.2302.13971
10. Liu P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*. 2023. Vol. 55, No. 9. Article 195. DOI: https://doi.org/10.1145/3560815
11. Wei J. et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*. 2022. Vol. 35. P. 24824–24837. DOI: https://doi.org/10.48550/arXiv.2201.11903
12. Lewis P. et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 9459–9474. DOI: https://doi.org/10.48550/arXiv.2005.11401
13. Hu E. J. et al. LoRA: Low-rank adaptation of large language models. *Proceedings of ICLR*. 2022. DOI: https://doi.org/10.48550/arXiv.2106.09685
14. Wu S. et al. BloombergGPT: A large language model for finance. *arXiv*, 2023. DOI: https://doi.org/10.48550/arXiv.2303.17564
15. Nie Y. et al. A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv*, 2024. DOI: https://doi.org/10.48550/arXiv.2406.11903

References

1. European Commission, Interoperable Europe Portal. (2025). Diia.AI: Ukraine's national AI agent for government services. Public Sector Tech Watch. https://interoperable-europe.ec.europa.eu/collection/public-sector-tech-watch/diiaai-ukraines-national-ai-agent-government-services [in English].
2. Organisation for Economic Co-operation and Development. (2025). Diia AI Assistant. OECD.AI Policy Observatory. https://oecd.ai/en/dashboards/policy-initiatives/diia-ai-assistant [in English].
3. Chen, T., & Gasco-Hernandez, M. (2025). Uncovering the results of AI chatbot use in the public sector: Evidence from US state governments. *Public Performance & Management Review*, *48*(6), 1331–1356. https://doi.org/10.1080/15309576.2024.2389864 [in English].
4. Sheth, J. N., Jain, V., Roy, G., & Chakraborty, A. (2022). AI-driven banking services: The next frontier for a personalised experience in the emerging market. *International Journal of Bank Marketing*, *40*(6), 1248–1271. https://doi.org/10.1108/IJBM-09-2021-0449 [in English].
5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5998–6008. https://doi.org/10.48550/arXiv.1706.03762 [in English].
6. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, *1*, 4171–4186. https://doi.org/10.18653/v1/N19-1423 [in English].
7. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., …, Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. https://doi.org/10.48550/arXiv.2005.14165 [in English].
8. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., …, Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744. https://doi.org/10.48550/arXiv.2203.02155 [in English].
9. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., …, Lample, G. (2023). LLaMA: Open and efficient foundation language models. *arXiv preprint* https://doi.org/10.48550/arXiv.2302.13971 [in English].
10. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, *55*(9), Article 195. https://doi.org/10.1145/3560815 [in English].
11. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, *35*, 24824–24837. https://doi.org/10.48550/arXiv.2201.11903 [in English].
12. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., …, Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474. https://doi.org/10.48550/arXiv.2005.11401 [in English].
13. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*. https://doi.org/10.48550/arXiv.2106.09685 [in English].
14. Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., …, Mann, G. (2023). BloombergGPT: A large language model for finance. *arXiv preprint* https://doi.org/10.48550/arXiv.2303.17564 [in English].
15. Nie, Y., Kong, Y., Dong, X., Mulvey, J. M., Poor, H. V., Wen, Q., & Zohren, S. (2024). A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv preprint* https://doi.org/10.48550/arXiv.2406.11903 [in English].

Савченко Сергій Олександрович – доктор філософії з галузі знань 12 «Інформаційні технології», докторант факультету інформаційних технологій Хмельницького національного університету. E-mail: savchenkoso@khmnu.edu.ua, ORCID: 0000-0001-9706-5334.

Savchenko Serhii Oleksandrovych – PhD in Information Technologies, Doctoral Student at the Faculty of Information Technologies of the Khmelnytskyi National University. E-mail: savchenkoso@khmnu.edu.ua, ORCID: 0000-0001-9706-5334.

Дата першого надходження статті до видання: 31.03.2026
Дата прийняття статті до друку після рецензування: 04.05.2026
Дата публікації (оприлюднення) статті: 01.07.2026