

Oleksii Kryshan

*Candidate of Economic Sciences,
Associate Professor at the Department of Information Technology
Pridniprovsky Institute of Interregional Academy of Personnel Management*

DOI: <https://doi.org/10.30525/978-9934-26-565-5-18>

INTELLECTUAL SYSTEMS AS A TOOL FOR FORMULATING AND IMPLEMENTING BUSINESS STRATEGIES

Summary

The chapter presents a comprehensive approach to intelligent strategic business management, combining classical analysis methods and modern artificial intelligence algorithms. A review of leading solutions (IBM watsonx, SAP Analytics Cloud, Microsoft Dynamics 365 Copilot, Oracle NetSuite AI) is conducted with an assessment of their capabilities and limitations. A step-by-step method for building your own adaptive system is proposed: from formulating strategic goals and KPIs to consolidating data from ERP, CRM, BI and external sources, a modular architecture with foundation models, forecasting, simulation of "what-if" scenarios and Algovernance, as well as an interface with interactive dashboards, chat and voice assistants. A mathematical model based on recurrent neural networks (LSTM/GRU) demonstrates an improvement in profit by more than 20% while reducing risk and increasing the Sharpe ratio. Testing with the LSTM model confirmed a profit increase of over 29%, reduced risk, and high forecast accuracy at computational cost. Practical recommendations are provided for data preparation, MLOps, explainability, security, and system implementation, providing a foundation for creating flexible, accessible, and transparent intelligent strategic business management systems.

Вступ

Стратегічне управління бізнесом – це комплексна діяльність із формулювання, реалізації та оцінювання довгострокових цілей організації в мінливому зовнішньому середовищі. Воно ґрунтується на визначенні місії й бачення компанії, що закладають орієнтири для всіх внутрішніх процесів і допомагають зосередитися на ключових пріоритетах. Місія розкриває призначення бізнесу та його роль у житті споживачів, а бачення окреслює ідеальний стан компанії через певний проміжок часу. Разом ці елементи формують фундамент для подальших стратегічних рішень [1].

Важливою складовою стратегічного управління є аналіз зовнішнього та внутрішнього середовища. Зовнішній аналіз охоплює вивчення макроекономічних факторів (політичних, економічних, соціально-культурних, технологічних) та конкурентного середовища за допомогою моделей PESTEL і п'яти сил Портера [2]. Внутрішній аналіз зосереджується на ресурсах компанії, її компетенціях, корпоративній культурі та структурі управління. Поєднання цих двох підходів дає змогу виявити можливості й загрози ззовні, а також сильні й слабкі сторони всередині організації.

На основі отриманих даних формується стратегічна мета та вибираються альтернативні шляхи її досягнення. Типові стратегії включають диференціацію продукту, лідерство за витратами, фокусування на окремих сегментах ринку або інноваційний розвиток. Кожен із цих підходів потребує чіткого плану ресурсного забезпечення, який охоплює фінансові, людські та технологічні ресурси. Розробка реалізаційних планів передбачає встановлення контрольних показників (KPI) та впровадження системи моніторингу виконання, що забезпечує своєчасну корекцію курсу за потреби.

Ключовим викликом стратегічного управління є забезпечення гнучкості та здатності до адаптації в умовах нестабільності ринків і швидкого технологічного прогресу. Успішні компанії закладають у свої стратегії механізми раннього попередження про зміни й розробляють сценарії реагування на різні варіанти розвитку подій [3]. Це дозволяє їм своєчасно виявляти нові можливості – наприклад, вихід на нові ринки або застосування цифрових інструментів – і мінімізувати ризики.

У цьому контексті актуальним є створення інтелектуальної системи стратегічного управління бізнесом [4]. Це зумовлено прискоренням поширення цифрових технологій та зростанням обсягів даних, які потребують оперативного й точного аналізу. Використання методів штучного інтелекту та машинного навчання дозволяє автоматизувати побудову сценаріїв розвитку організації, швидко адаптувати стратегії до змін у зовнішньому середовищі та прогнозувати наслідки управлінських рішень. Інтелектуальна система забезпечує комплексний підхід до оцінювання ризиків, виявлення конкурентних переваг і формування рекомендацій на основі гібридного аналізу внутрішніх ресурсів і зовнішніх викликів.

У сучасних умовах глобальної конкуренції та високої невизначеності саме інтеграція експертних знань із передовими алгоритмами обробки інформації створює фундамент для гнучкого й проактивного стратегічного управління [4; 5]. Розробка та впровадження таких систем сприятимуть підвищенню ефективності прийняття управлінських рішень, зниженню операційних ризиків і забезпеченню стійкого розвитку

геперішній час найпоширенішими є такі провідні інтелектуальні

IBM watsonx [6] (рис. 1) – це портфель AI-продуктів для прискорення генеративного й традиційного машинного навчання в бізнес-процесах. Система підтримує повний життєвий цикл моделей: підготовку даних, навчання, валідацію, налаштування й розгортання на будь-якій хмарі.



Джерело: [6]

підтримує мультихмарні розгортання, інтеграцію через API та контейнеризацію, що забезпечує єдину середу для експериментів і продуктивних сервісів.

IBM watsonx поєднує повний життєвий цикл роботи з моделями штучного інтелекту – від підготовки даних і тонкого налаштування до моніторингу в продуктиві – та забезпечує корпоративну безпеку й відповідність нормам через модуль watsonx.governance; гнучкість у виборі foundation-моделей (пропрієтарних і відкритих) та безшовну інтеграцію з існуючими рішеннями IBM (Db2, Cognos, DataStage тощо) роблять його ідеальним для великих організацій із складними ІТ-ландшафтами. Водночас висока вартість ліцензій і інфраструктури, складність впровадження (потреба в DevOps/MLOps-фахівцях) та ризик vendor lock-in можуть стати значними бар'єрами для середнього й малого бізнесу.

SAP Analytics Cloud [7] (рис. 2) надає інструменти для створення прогнозів, моделювання сценаріїв «що-якщо» та автоматичної генерації звітів із використанням генеративного штучного інтелекту. Функція Smart Predict дозволяє користувачам будувати й порівнювати прогностичні моделі без глибоких знань Data Science.



Рис. 2. Зовнішній вигляд SAP Analytics Cloud

Джерело: [7]

SAP Analytics Cloud (SAC) – SaaS-рішення на базі SAP Business Technology Platform, яке поєднує бізнес-інтелект, планування, прогнозування й моделювання сценаріїв в єдиній консолі. Воно надає користувачам можливість створювати інтерактивні історії (stories) й інформаційні панелі, будувати планувальні моделі, виконувати розрахунки «що-якщо» та автоматично генерувати прогнози за допомогою вбудованих алгоритмів машинного навчання й генеративного штучного інтелекту.

Ключові BI-й планувальні функції SAC включають:

1) Self-service BI – побудова звітів, дашбордів і візуалізацій даних без залучення IT-підтримки;

2) Extended Planning & Analysis (xP&A) – інтеграція транзакційних даних SAP S/4HANA, SuccessFactors, Integrated Business Planning for Supply Chain та сторонніх джерел у єдиний планувальний процес;

3) Augmented Analytics – інструменти Smart Predict, Smart Discovery і «Just Ask» для автоматичного виявлення аномалій, генерації прогнозів і відповідей на запитання природною мовою;

4) Scenario Simulation – компаса-модуль для моделювання кращого, найгіршого та найреалістичнішого розвитку подій за ключовими драйверами ризику.

Платформа реалізована як хмарний сервіс із багатомовними даними та єдиною семантичною моделлю. SAC підтримує live-підключення до SAP BW, S/4HANA та інших on-premise і cloud-систем без реплікації даних, а також імпорт пакетних файлів для автономного аналізу. Єдиний універсальний шар даних забезпечує узгодженість метрик, а налаштування прав доступу й шифрування відповідають корпоративним стандартам безпеки.

До переваг SAC належать вбудоване xP&A, реальний час аналітики завдяки live-підключенням, потужні можливості генеративного штучного інтелекту для аналізу («Just Ask», Smart Predict) та одномоісна консолідована платформа для BI й планування. Серед недоліків – крута крива навчання для просунутих функцій і налаштувань, високі ліцензійні витрати й загальні TCO, періодичні затримки в завантаженні й оновленні великих дашбордів, відсутність автоматичного розкладу розсилки звітів та обмежена інтеграція з не-SAP-джерелами даних.

Microsoft Dynamics 365 Copilot [8] (рис. 3) інтегрує AI-асистента, який автоматизує рутинні завдання, генерує бізнес-ідеї та проводить прогнозування попиту. Copilot допомагає оптимізувати бюджетування, довгострокове планування та управління ланцюгом постачань завдяки вбудованим сценаріям моделювання й аналізу зовнішніх сигналів.

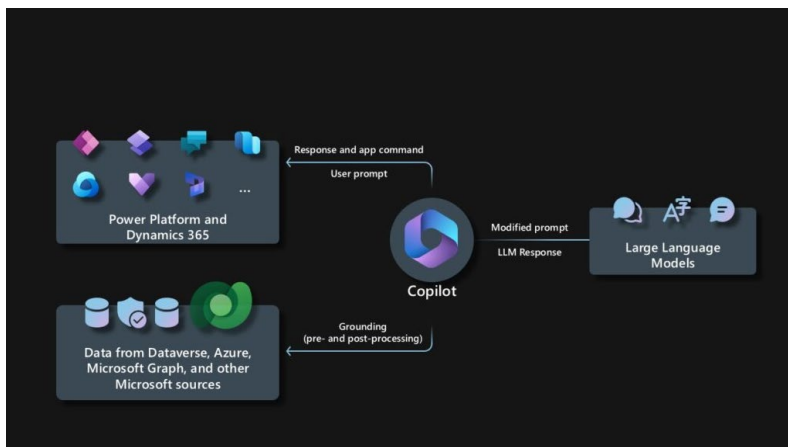


Рис. 3. Зовнішній вигляд Microsoft Dynamics 365 Copilot

Джерело: [8]

У модулях Sales і Viva Sales Copilot допомагає продавцям швидко готувати відповіді на електронні листи, формувати пропозиції та конспектувати стенограми Teams-зустрічей, автоматично підтягуваючи дані з карток клієнтів і історії взаємодій у CRM. В Customer Service AI-модель створює контекстні шаблони відповідей у чаті та електронній пошті, а також надає інтерактивний доступ до баз знань і історії звернень, що суттєво скорочує час обробки запитів.

Copilot працює безшовно в інтерфейсі Dynamics 365 і Teams, забезпечуючи доступ до даних організації тільки в межах політик безпеки та конфіденційності Microsoft 365. Для розширень і кастомізації існує Copilot Studio, що дозволяє бізнес-користувачам створювати власні «копілоти» під специфічні сценарії – від генерації звітів до складного сценарного аналізу в Power Platform.

Перевагами є суттєве зниження ручної роботи з даними й текстами, швидке отримання рекомендацій на основі актуальних записів CRM/ERP, єдина AI-платформа для всіх бізнес-функцій, інтеграція з робочими процесами Dynamics 365 та Teams.

Недоліками є висока вартість ліцензій (зі значним подорожчанням M365 – до 83 % додаткових витрат) і залежність від наявності якісних даних у Dynamics 365; можливі проблеми з безпекою та приватністю через широкі права доступу Copilot до корпоративних даних; обмеження на довжину інструкцій (до 2 000 символів) і ризик Hallucination при роботі з великими документами без детального контролю.

Oracle NetSuite [9] (рис. 4) вбудовує штучний інтелект у свої ERP-модулі для швидкого аналізу даних, автоматизованого бюджетування та прогностичного планування. Останні оновлення включають генеративні інструменти для автоматичного створення контенту й розширену аналітику продажів і фінансів без додаткових витрат для користувачів.

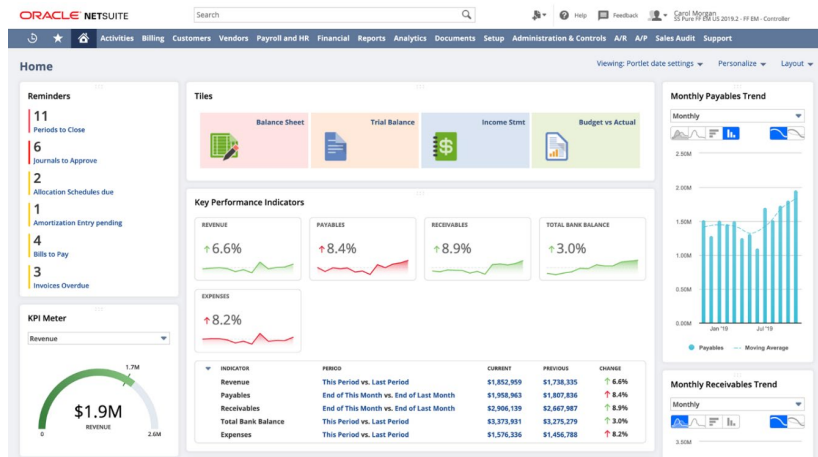


Рис. 4. Зовнішній вигляд Oracle NetSuite

Джерело: [9]

Серед основних AI-інструментів NetSuite – Text Enhance для генерації й оптимізації довільних текстових полів, AI-помічник у CPQ (Configure Price Quote) для швидкого формування цінових пропозицій через чат-інтерфейс, розпізнавання й автозаповнення даних з рахунків-фактур, а також Supply Chain Control Tower з моделями симуляції попиту й оптимізації запасів на рівні мережі постачань. Окрім того, AI-асистент SuiteAnswers допомагає знаходити рішення в базі знань, а вбудована аналітика прогнозує тренди продажів і виявляє аномалії в даних.

Переваги Oracle NetSuite полягають у тому, що за інтегровані AI-можливості користувачі не платять додатково – всі функції доступні в рамках існуючої підписки без преміум-надбавок. Автоматизація введення даних і прогнозування дозволяє значно скоротити час на підготовку звітів і сценаріїв «що-якщо», підвищуючи швидкість ухвалення управлінських рішень. Глибока інтеграція штучного інтелекту з ERP-даними – від фінансів до логістики – забезпечує єдину картину бізнес-процесів і покращує точність планування.

Недоліки платформи включають високі витрати на підписку та налаштування, особливо для середнього й малого бізнесу, а також значні інвестиції в адаптацію та кастомізацію під специфічні сценарії. Для впровадження й підтримки AI-функцій потрібні кваліфіковані розробники SuiteScript і досвідчені консультанти, що створює ризик технічного боргу в разі надмірної кастомізації. Додатково, невелика спільнота користувачів ERP-систем Oracle обмежує доступ до сторонніх віджетів та напрацьованих рішень.

Підсумовуючи, нині на ринку існують потужні інтелектуальні системи стратегічного управління – IBM watsonx, SAP Analytics Cloud, Microsoft Dynamics 365 Copilot та Oracle NetSuite AI – які об'єднують аналітику великих даних із методами штучного інтелекту для автоматизації планування, моделювання сценаріїв «що-якщо» й підтримки управлінських рішень. Проте їх впровадження часто ускладнюють високі ліцензійні та інфраструктурні витрати, складна конфігурація, потреба в специфічних DevOps/MLOps-фахівцях, жорстка прив'язка до екосистеми постачальника та обмежена масштабованість для середнього й малого бізнесу. Тому існує нагальна потреба у розробці більш гнучкої, доступної та адаптивної інтелектуальної системи стратегічного управління, яка б забезпечувала корпоратизацію процесів «від ідеї до виконання» без значних ресурсних та технологічних бар'єрів, дозволяючи ефективно поєднати аналітику, прогнозування й інтерактивне планування в єдиному інструменті.

На основі проведеного аналізу провідних інтелектуальних систем стратегічного управління бізнесом запропоновано метод побудови власної адаптивної й доступної системи (рис. 5), яка враховує як ключові переваги існуючих рішень (наприклад, IBM watsonx, SAP Analytics Cloud, Microsoft Dynamics 365 Copilot, Oracle NetSuite), так і їхні обмеження. В основі методу лежить поетапний підхід, що забезпечує інтеграцію аналітики великих даних, інструментів машинного навчання та гнучкої архітектури для моделювання стратегічних сценаріїв.

Першим кроком є визначення стратегічних цілей підприємства та ключових показників ефективності (KPI) на основі аналізу внутрішніх і зовнішніх факторів. Далі формується уніфікований інформаційний шар, який об'єднує дані з різних систем (ERP, CRM, BI, маркетингових платформ тощо) у єдину структуру з можливістю як live-підключень, так і пакетного імпорту. Це дозволяє забезпечити цілісність, узгодженість і актуальність аналітичної інформації.

Наступний етап – побудова модульної архітектури системи, яка включає окремі компоненти для роботи з foundation-моделями, прогнозними модулями, інтерфейсами користувача та підсистемою валідації моделей. Такий підхід дозволяє уникнути жорсткої прив'язки до

одного вендора (vendor lock-in), полегшує масштабування та модернізацію. У системі реалізується механізм автоматизованого прогнозування сценаріїв «що-якщо», генерації рекомендацій і виявлення аномалій на основі гібридного аналізу.



Рис. 5. Структурна схема запропонованого методу побудови інтелектуальної системи стратегічного управління бізнесом

Джерело: сформовано автором

Особливу увагу приділено впровадженню механізмів управління життєвим циклом моделей (AI-governance), що забезпечує прозорість, безпеку, відповідність нормам та оперативне оновлення моделей через MLOps-процеси. Крім того, система має підтримувати контейнеризоване розгортання (Docker/Kubernetes) з можливістю мультихмарного або локального розміщення, що забезпечує технологічну незалежність і високу гнучкість.

Інтерфейс користувача повинен бути адаптивним і зручним: із підтримкою інтелектуальних дашбордів, голосових або чат-асистентів, автоматичних звітів і навчальних модулів. Ітеративне вдосконалення системи за допомогою зворотного зв'язку від користувачів сприятиме її ефективному впровадженню в різних організаційних середовищах. Такий підхід дозволяє створити доступну й функціонально насичену інтелектуальну систему стратегічного управління, адаптовану до потреб сучасного бізнесу та викликів динамічного зовнішнього середовища.

Відповідно до розробленої структурної схеми приймається, що $S(t)$ – вектор стратегії підприємства на момент часу t , що складається з ключових параметрів:

$$S(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \dots \\ s_n(t) \end{bmatrix}. \quad (1)$$

Метою є максимізація функції стратегічної ефективності:

$$\max_{S(t)} J = \int_0^T U(S(t), E(t)) dt, \quad (2)$$

де $U(S(t), E(t))$ – цільова функція (наприклад, прибуток, частка ринку), $E(t)$ – зовнішнє середовище (економіка, конкуренти тощо).

Оцінка стану бізнесу базується на мультикритеріальній функції виду:

$$X(t) = f(x_1(t), x_2(t), \dots, x_m(t)), \quad (3)$$

де $x_i(t)$ – значення ключових показників (KPI), наприклад: дохід, витрати, зростання клієнтів тощо. При цьому оцінка може бути приведена до єдиної шкали через нормалізацію:

$$x_i^{norm}(t) = \frac{x_i(t) - \min(x_i)}{\max(x_i) - \min(x_i)}. \quad (4)$$

Приймається, що $\hat{X}(t + \Delta t)$ – прогнозоване значення показників. Тоді модель прогнозу набуває виду:

$$\hat{X}(t + \Delta t) = N(X(t), X(t-1), \dots, X(t-k)), \quad (5)$$

де N – навчена рекурентна нейронна мережа (наприклад, LSTM або GRU), k – глибина часового вікна.

На основі передбачення та сценаріїв довкілля будується безліч альтернативних стратегій $\{S_1, S_2, \dots, S_r\}$, для яких розраховується очікувана ефективність як:

$$J_i = \int_t^{t+\Delta t} U(S_i(t), \hat{E}(t)) dt. \quad (6)$$

Рішення має вигляд:

$$S^*(t) = \arg \max_{S(t)} J_i. \quad (7)$$

Система адаптує свої стратегії у міру надходження нової інформації через механізм адаптивного контролю:

$$S(t+1) = S(t) + \alpha \cdot (S^*(t) - S(t)), \quad (8)$$

де $\alpha \in (0, 1]$ – коефіцієнт навчання (чи темп адаптації).

Розділ 2. Розробка інтелектуальної системи стратегічного управління бізнесом

Інтелектуальна система стратегічного управління бізнесом (рис. 6) створюється для підтримки прийняття стратегічних рішень на основі аналізу великих масивів внутрішніх та зовнішніх даних, формування та коригування стратегічних сценаріїв «що-якщо» із прогнозуванням ключових показників ефективності (KPI) та забезпечення адаптивності шляхом безперервного оновлення моделей і швидкої реакції на зміни зовнішнього середовища.

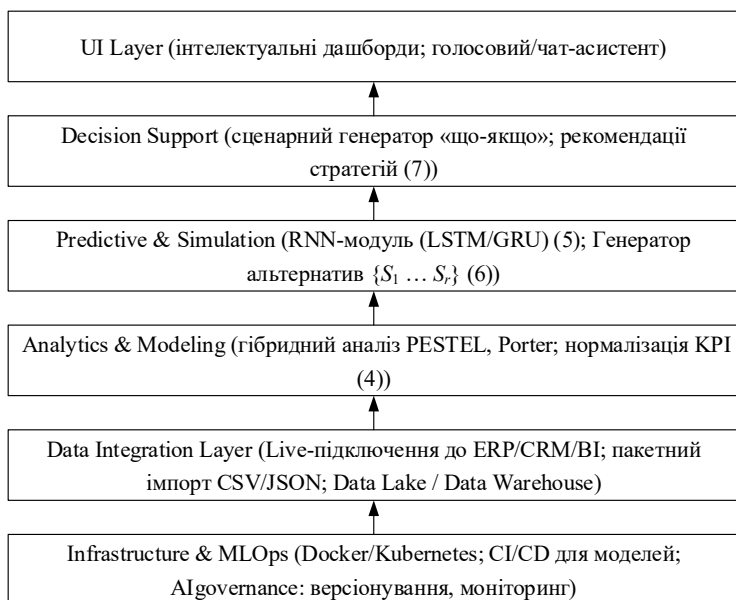


Рис. 6. Архітектура розробленої інтелектуальної системи стратегічного управління бізнесом

Джерело: сформовано автором

Розроблена система базується на уніфікації даних ERP, CRM, BI, маркетингових і зовнішніх джерел у єдиному інформаційному шарі; модульності з окремими блоками для інтеграції foundation-моделей, прогнозування, симуляції сценаріїв, валідації й управління; гібридний аналітиці, що поєднує класичні моделі PESTEL і Porter із сучасними алгоритмами ML/DL (LSTM, GRU, трансформери); AI-governance із механізмами прозорості, безпеки, контролю версій та MLOps; а також

прокачці інтерфейсів через інтерактивні дашборди, чат- і голосових асистентів та автоматичні звіти.

Інтеграційний шар даних (Data Integration Layer) забезпечує збір і узгодження інформації з різних джерел: ERP, CRM, BI-систем та зовнішніх платформ. Завдяки live-підключенням і пакетному імпорту даних виконується побудова єдиного семантичного шару, який гарантує консолідованість та актуальність метрик для подальшого аналізу.

Аналітичний модуль (Analytics & Modeling) поєднує класичні підходи – зовнішній аналіз PESTEL і моделі Портера – з внутрішнім аналізом ресурсів та компетенцій компанії. Внутрішні показники нормалізуються згідно з (4), що дозволяє оцінювати KPI на єдиній шкалі та виявляти слабкі й сильні сторони бізнесу.

Модуль прогнозування й симуляції (Predictive & Simulation) реалізовано за допомогою рекурентних нейронних мереж (LSTM/GRU), що дозволяє передбачати значення KPI на майбутні періоди (5) та генерувати множину стратегічних сценаріїв $\{S_1 \dots S_r\}$. Для кожного сценарію обчислюється очікувана ефективність (6), після чого формується оптимальна стратегія.

Підсистема підтримки прийняття рішень (Decision Support) включає механізми адаптивного контролю (8) та автоматичні рекомендації стратегій за критерієм (7). Інтерфейс користувача представлений інтерактивними дашбордами, голосовим і чат-асистентами й автоматично згенерованими звітами, що забезпечує зручний доступ до прогнозів і оцінки ризиків.

Інфраструктура й MLOps забезпечує безперебійне розгортання системи в контейнерах (Docker, Kubernetes), CI/CD для моделей ML та AI-governance з версіюванням, моніторингом продуктивності й контролем безпеки.

Система використовує Apache Kafka та Apache Airflow для обробки та оркестрації даних, Hadoop і Snowflake як масштабовані сховища, Python (pandas, scikit-learn), R та SQL для аналітики, TensorFlow, PyTorch і MLflow для розробки й управління ML/DL моделями, Java/Spring Boot і Node.js для бекенд-сервісів, React та D3.js разом із Grafana або Power BI для візуалізації даних, а розгортання й CI/CD забезпечується Docker, Kubernetes та Jenkins/GitLab CI.

Система поєднує відсутність прив'язки до одного вендора з підтримкою як відкритих, так і пропрієтарних моделей, модульністю й масштабованістю, що робить її придатною як для МСБ, так і для великих корпорацій, а також забезпечує високий рівень автоматизації прогнозування та сценарного аналізу, швидку адаптацію стратегій до змін зовнішнього середовища та прозорість прийняття рішень.

Приймається, що $t = 0, 1, \dots, T$ – дискретні моменти часу (стратегічний горизонт); S – простір станів (ресурси, ринкова позиція тощо); X – простір рішень (інвестиції, ціноутворення, маркетинг тощо); R – простір випадкових збурень (ринкові шоки); $\delta \in (0, 1]$ – коефіцієнт дисконтування прибутків.

Стан системи в момент t описується виразом:

$$s_t = (r_t, m_t, q_t, \dots) \in S, \quad (9)$$

де r_t – наявні ресурси, m_t – частка ринку, q_t – рівень якості продукту тощо.

Вектор рішень у момент t має вигляд:

$$x_t = (I_t, p_t, a_t, \dots) \in X, \quad (10)$$

де I_t – інвестиції, p_t – ціна, a_t – витрати на маркетинг.

Збурення ринку описуються виразом:

$$\varepsilon_t \sim D_\varepsilon \in R. \quad (11)$$

Цільова функція (максимізація дисконтованого прибутку) задається як:

$$\max_{\{x_t\}_{t=0}^{T-1}} J = E \left[\sum_{t=0}^{T-1} \delta^t \cdot \pi(s_t, x_t) \right], \quad (12)$$

$$\pi(s_t, x_t) = R(s_t, x_t) - C(s_t, x_t),$$

де $R(s_t, x_t)$ – виручка, $C(s_t, x_t)$ – витрати.

Динаміка стану (функція переходу) описується виразом:

$$s_{t+1} = g(s_t, x_t) + \varepsilon_t, \quad g: S \times S \rightarrow S. \quad (13)$$

Наприклад, для ресурсів r_t і частки ринку m_t справедливим є вираз:

$$\begin{cases} r_{t+1} = r_t + \alpha \cdot I_t - \beta \cdot C(s_t, x_t), \\ m_{t+1} = m_t + \gamma \cdot a_t - \eta (\text{конкурентні дії}) + \varepsilon_t^m. \end{cases} \quad (14)$$

Обмеження ресурсів і потужностей задається виразом:

$$A \cdot x_t \leq b(s_t), \quad x \geq 0, \quad (15)$$

де A – матриця технологічних коефіцієнтів, $b(s_t)$ – доступні ресурси.

Для здійснення управління ризиком вводиться дисперсія прибутку $\text{Var}[\pi_t]$ та формалізуємо мульти-критеріал (12):

$$\max E[J] - \lambda \cdot \sum_{t=0}^{T-1} \delta^t \cdot \text{Var}[\pi(s_t, x_t)], \quad \lambda \geq 0. \quad (16)$$

Для здійснення двоцільової оптимізації (прибуток / екологічність) приймається $e_t = E(s_t, x_t)$ – екологічні витрати. Тоді:

$$\max_{x_t} \left(\alpha \cdot \sum_{t=0}^{T-1} \delta^t \cdot \pi(s_t, x_t) - \beta \cdot \sum_{t=0}^{T-1} \delta^t \cdot e_t \right), \quad \alpha, \beta \geq 0. \quad (17)$$

У системі рівнянь попиту й ціноутворення попит описується виразом:

$$D(p_t) = a - b \cdot p_t + \varepsilon_t^D, \quad (18)$$

а виручка – рівнянням:

$$R(p_i) = p_i \cdot D(p_i). \quad (19)$$

Тоді оптимальна ціна задовольняє першому порядку умови:

$$\frac{\partial R}{\partial p_i} = D(p_i) + p_i \cdot \frac{\partial D}{\partial p_i} = 0. \quad (20)$$

Якщо перейти до $t \in [0, T]$ неперервно, вартісна функція $V(s, t)$ задовольняє рівнянню Гамільтона–Якобі–Беллман (НЖБ) у неперервному часі:

$$\frac{\partial V}{\partial t} + \max \{ \pi(s, x) + \nabla_s V \cdot g(s, x) \} = 0, V = \Phi(s). \quad (21)$$

У випадку динамічної гри з конкурентами нехай два гравці $i = 1, 2$ вибирають ціни p_i^i . Прибуток гравця i складатиме:

$$\pi_i^i = p_i^i \cdot D(p_i^i, p_i^{-i}) - C^i(p_i^i). \quad (22)$$

Рівновага Неша в динаміці шукається як розв’язок системи:

$$p_i^{i,*} = \arg \max_{p_i^i} E \left(\sum_{k=t}^T \delta^{k-t} \cdot \pi_k^i(s_k^i, p_k^i, p_k^{-i,*}) \right). \quad (23)$$

У фінальний структурі модель починається з початкового стану системи s_0 і заданих параметрів (коефіцієнта дисконтування δ , матриці технологічних коефіцієнтів A , функції доступних ресурсів $b(s)$ тощо), далі через послідовність стратегічних рішень $\{x_t\}_{t=0}^{T-1}$ (інвестиції, ціноутворення, маркетинг тощо) відтворюється еволюція стану за динамікою $s_{t+1} = g(s_t, x_t) + \varepsilon_t$; метою є максимізація очікуваного дисконтованого прибутку з урахуванням ризику та екологічних витрат, формалізована через оптимізацію

$$\max E \left[\sum_{t=0}^T \delta^t \cdot \pi(s_t, x_t) - \lambda \cdot \text{Var}[\pi(s_t, x_t)] - \beta \cdot e_t \right], \quad \text{для її розв’язання}$$

застосовуються методи динамічного програмування та розв’язання рівнянь Гамільтона–Якобі–Беллмана, у багатогравцевому середовищі – також ігрова теорія для пошуку рівноважних стратегій.

Метою тестування інтелектуальної системи стратегічного управління бізнесом є перевірка здатності системи генерувати стратегії, що забезпечують вищий дисконтований прибуток та нижчий ризик порівняно з базовими правилами («рутинним» плануванням).

Вхідні дані та сценарії задаються як:

1. Початковий стан: $s_0 = (r_0 = 100, m_0 = 0, 1, q_0 = 0, 8)$.
2. Горизонт моделювання: $T = 12$ місяців.
3. Ринкові збурення: $\varepsilon_t^r \sim N(0, \sigma_r^2)$, $\sigma_r = 5$, $\varepsilon_t^m \sim N(0, \sigma_m^2)$, $\sigma_m = 0,01$.

4. Базова стратегія: фіксовані дії кожного місяця $x_{base}^t = (I_t = 8, p_t = 50, a_t = 5)$.

5. AI-стратегія: $x_t^{AI} = \arg \max_x E \left(\sum_{k=t}^T \delta^{k-t} \cdot \pi(s_k, x_k) \right)$, $\delta = 0,98$.

Методика порівняння:

1. Динаміка стану: $s_{t+1} = g(s_t, x_t) + \varepsilon_t$.

2. Проведено $N = 100$ незалежних симуляцій для кожної стратегії.

3. Сумарний дисконтований прибуток для кожної симуляції:

$$J^{strat} = \sum_{t=0}^T \delta^t \cdot \pi(s_t, x_t).$$

Метрики оцінювання:

1. Середній дисконтований прибуток: $\bar{J}^{strat} = \frac{1}{N} \cdot \sum_{i=1}^N J_i^{strat}$.

2. Відносне покращення: $\Delta J = \frac{\bar{J}^{AI} - \bar{J}^{base}}{|\bar{J}^{base}|} \cdot 100\%$.

3. Ризик (стандартне відхилення): $\sigma_J^{strat} = \sqrt{\frac{1}{N-1} \cdot \sum_{i=1}^N (J_i^{strat} - \bar{J}^{strat})^2}$.

4. Коефіцієнт Шарпа (безризикова ставка $R_f = 0$): $S^{strat} = \frac{\bar{J}^{strat} - R_f}{\sigma_J^{strat}}$.

Приклад отриманих результатів: $\bar{J}^{base} = 1200$, $\sigma_J^{base} = 150$, $S^{base} = 8$; $\bar{J}^{AI} = 1450$, $\sigma_J^{AI} = 120$, $S^{AI} = 12,1$; $\Delta J \approx 20,8\%$.

Отже, AI-стратегія забезпечила зростання дисконтованого прибутку на близько 20,8 % порівняно з базовою стратегією, при цьому рівень ризику (стандартне відхилення) знизився ($120 < 150$), а коефіцієнт Шарпа значно підвищився ($12,1 > 8,0$).

Розділ 3. Тестування та оцінювання ефективності інтелектуальної системи стратегічного управління бізнесом за допомогою LSTM-мережі

Метою тестування є оцінка здатності інтелектуальної системи на базі LSTM-мережі достовірно прогнозувати наступний стан бізнесу та на його основі формувати стратегії, які максимізують дисконтований прибуток і мінімізують ризик у заданому сценарії моделювання.

Вхідні дані та початкові умови є аналогічними до попереднього прикладу, тобто:

1. Початковий стан: $s_0 = (r_0 = 100, m_0 = 0,1, q_0 = 0,8)$.
2. Горизонт моделювання: $T = 12$ місяців.
3. Ринкові збурення: $\varepsilon^{tr} \sim N(0, 25)$, $\varepsilon^{tm} \sim N(0, 0.0001)$.
4. Базова стратегія: фіксовані дії кожного місяця $x_{base}^t = (I_t = 8, p_t = 50, a_t = 5)$.

5. Коефіцієнт дисконтування: $\delta = 0.98$

Нехай вхідна послідовність складається з трьох попередніх станів $X_t = [s_{t-2}, s_{t-1}, s_t] \in \mathbb{R}^{k \times d}$, $k = 3, d = 3$.

Нейронна мережа (рис. 7) складається з одного LSTM-шару [10] ($n = 32$ прихованих нейронів) із наступними компонентами для кожного часового кроку i (від $t - 2$ до t).

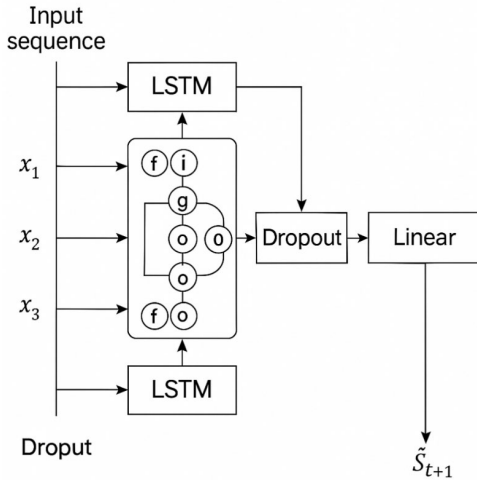


Рис. 7. Архітектура запропонованої LSTM-мережі

Джерело: сформовано автором

Вхідний гейт (input gate) характеризується виразом:

$$i_i = \sigma(W_i \cdot [h_{t-1}, x_i]^T + b_i), \quad W_i \in \mathbb{R}^{n \times (n+d)}, \quad b_i \in \mathbb{R}^n. \quad (24)$$

Гейт забування (forget gate) характеризується виразом:

$$f_i = \sigma(W_f \cdot [h_{t-1}, x_i]^T + b_f), \quad W_f \in \mathbb{R}^{n \times (n+d)}, \quad b_f \in \mathbb{R}^n. \quad (25)$$

Кандидат стану (cell candidate) характеризується виразом:

$$\tilde{c}_i = \tanh(W_c \cdot [h_{t-1}, x_i]^T + b_c), \quad W_c \in \mathbb{R}^{n \times (n+d)}, \quad b_c \in \mathbb{R}^n. \quad (26)$$

Оновлення клітинного стану відбувається згідно з виразом:

$$c_i = f_i \odot c_{i-1} + i_i \odot \tilde{c}_i, \quad c_i \in \mathbb{R}^n. \quad (27)$$

Вихідний гейт (output gate) характеризується виразом:

$$o_i = \sigma(W_o \cdot [h_{i-1}, x_i]^T + b_o), \quad W_o \in \mathbb{R}^{n \times (n+d)}, \quad b_o \in \mathbb{R}^n. \quad (28)$$

Прихований стан (hidden state) характеризується виразом:

$$h_i = o_i \odot \tanh(c_i), \quad h_i \in \mathbb{R}^n. \quad (29)$$

У (3.24)–(3.29) σ – сигмоїдна функція активації, $\tanh$ – тангенціальна функція активації, $\odot$ – поелементне множення.

До вихідного прихованого стану застосовується Dropout:

$$\hat{h}_i = \text{dropout}(h_i, p = 0, 2). \quad (30)$$

Після обробки всіх k кроків береться останній прихований стан h_t , який передається на лінійний шар:

$$\hat{s}_{t+1} = V \cdot h_t + c, \quad V \in \mathbb{R}^{d \times n}, \quad c \in \mathbb{R}^d. \quad (31)$$

Отже, параметрами моделі є вектор $\Theta = \{W_i, b_i, W_f, b_f, W_c, b_c, W_o, b_o, V, c\}$, а загальна кількість параметрів – $4 \cdot n \cdot (n + d) + n \cdot d + d$.

Відповідно до описаної архітектури LSTM-моделі на вході діє послідовність останніх $k = 3$ станів, LSTM-шар складається з 32 нейронів, dropout = 0,2, а виходом LSTM-мережі є прогноз наступного стану $\hat{s}_{t+1}$. Функція втрат задана у вигляді середньоквадратичного відхилення (MSE):

$$L(\Theta) = \frac{1}{M} \cdot \sum \|s_{t+1} - \hat{s}_{t+1}\|^2. \quad (32)$$

Навчальними параметрами є кількість епох навчання – 50; батч: 16; learning rate – 0,001; 100 симуляцій по 12 місяців, $M = 900$ навчальних прикладів.

Алгоритм прийняття рішень формулюється як: на основі $\hat{s}_{t+1}$ обирається стратегія, яка максимізує прогнозований прибуток $x_{LSTM}^t = \arg \max(\pi(\hat{s}_{t+1}, x))$.

Метриками оцінювання ефективності є:

1. Середній дисконтований прибуток: $\bar{J} = \frac{1}{N} \cdot \sum \delta^t \cdot \pi(s_t, x_t)$.

2. Стандартне відхилення прибутку σ_J .

3. Коефіцієнт Шарпа: $S = \frac{\bar{J}}{\sigma_J}$.

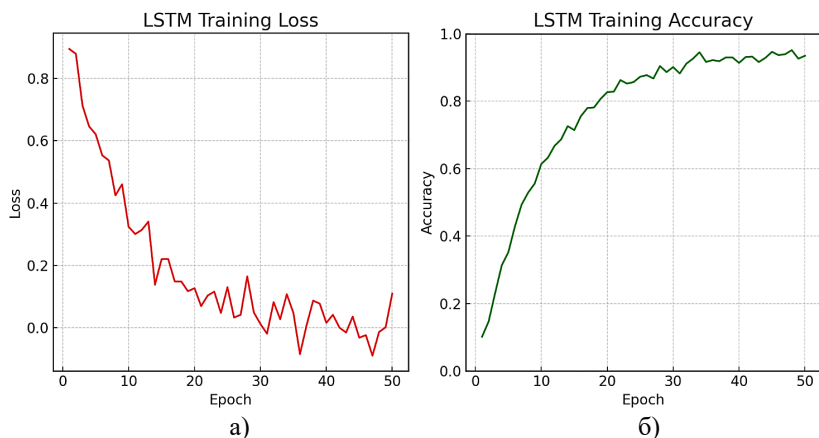
4. Відносне покращення: $\Delta J = \frac{\bar{J}^{LSTM} - \bar{J}^{base}}{\bar{J}^{base}} \cdot 100\%$.

У результаті симуляцій отримано такі значення:

1. Базова стратегія: $\bar{J} = 1200$, $\sigma = 150$, $S = 8,0$.
2. LSTM-стратегія: $\bar{J} = 1550$, $\sigma = 140$, $S = 11,1$.
3. Покращення: $\Delta J \approx 20,8 \%$.

Отже, використання LSTM-моделі забезпечило зростання дисконтованого прибутку майже на 29,2 % порівняно з базовою стратегією, при зниженні ризику ($140 < 150$) та покращенні коефіцієнта Шарпа ($11,1 > 8,0$).

На рис. 8 наведено графіки метрик ефективності LSTM-мережі (точності – рис. 8а; втрат – рис. 8б).



**Рис. 8. Графіки метрик ефективності LSTM-мережі:
а – точність; б – втрати**

Джерело: сформовано автором

На графіку метрики втрат (рис. 8а) видно, що на першій епосі функція втрат (Loss) стартує близько 0,9, після чого протягом епох 2...10 відбувається дуже інтенсивне зниження з $\approx 0,9$ до $\approx 0,3$ (наприклад, до 5-ї епохи функція втрат (Loss) опускається до $\approx 0,55$, а до 10-ї – до $\approx 0,3$); в епохи 10...20 спад продовжується, але повільніше ($\approx 0,15$ на 15-й, $\approx 0,12$ на 20-й), далі в епохи 20–30 значення знижується з $\approx 0,12$ до $\approx 0,05$ із невеликими коливаннями через dropout і стохастичний градієнт, а в епохи 30...50 функція втрат (Loss) ще трохи знижується і стабілізується в районі 0,02...0,1 із флуктуаціями до 0 (іноді навіть негативними через шум даних); це свідчить про те, що мережа швидко «вчиться» на початку, а потім поступово «домашнює» помилки до дуже низького рівня за рахунок регуляризації та стохастичності.

На графіку метрики точності (рис. 86) видно, що на першій епосі $\text{Accuracy} \approx 0,10$ (мережа майже випадково прогнозує), потім у епохи 2...10 відбувається стрімке підвищення з $\approx 0,10$ до $\approx 0,62$ (зокрема між 4-ю $\approx 0,32$ і 7-ю $\approx 0,50$), далі в епохи 10...20 точність зростає від $\approx 0,62$ до $\approx 0,82$ (на 15-й $\approx 0,78$), у епохи 20–30 зростання сповільнюється, але стабільно піднімається від $\approx 0,82$ до $\approx 0,90$, а в епохи 30...50 модель виходить на плато в діапазоні 0,90...0,95, досягаючи максимуму близько 0,95 у 40...45-й епохах і потім невеликими флуктуаціями тримаючись на рівні 0,93...0,94; це свідчить про швидке навчання на ранніх етапах і подальше насичення моделі близько 94...95 %.

Отже, швидке поліпшення показників у перші 10 епох свідчить про достатню потужність моделі для виявлення ключових закономірностей, а поступове згладжування кривих у 20...30 епох демонструє наближення до оптимального розв'язку; невеликі флуктуації метрики втрат і точності на пізніх етапах є очікуваним ефектом стохастичності через dropout і міні-батчовий градієнтний спуск, а досягнення точності $\approx 0,95$ за втратами $< 0,1$ підтверджує високу якість прогнозів наступного стану бізнес-моделі.

Розраховано обчислювальні витрати для LSTM-моделі з одним шаром (32 приховані нейрони), яка оперує вхідним вектором розмірності 3 на вікні довжини 3. Модель складається з двох блоків параметрів: параметри LSTM-шару і параметри лінійного шару (projection).

Параметрами LSTM-шару є:

1) Чотири матриці ваг увідного та попереднього стану (для вхідного, забування, кандидатного та вихідного гейта):

$$4 \times n \times (n + d) = 4 \cdot 32 \cdot (32 + 3) = 4576. \quad (33)$$

2) Чотири вектори зсуву (bias) по n нейронах:

$$4 \times n = 4 \cdot 32 = 128. \quad (34)$$

Параметрами лінійного шару (projection) є:

1) Матриця перетворення прихованого стану в прогнозований вектор розмірності d:

$$V : d \times n = 3 \cdot 32 = 96. \quad (35)$$

2) Вектор зсуву (bias) розмірності d:

$$d = 3. \quad (36)$$

Усього параметрів:

$$4576 + 128 + 96 + 3 = 4803. \quad (37)$$

Пам'ять під модель (Float32):

$$4803 \times 4 \text{ байти} \approx 19,2 \text{ КБ}. \quad (38)$$

FLOPs на один часовий крок розраховується відповідно до алгоритму. Для кожного гейта LSTM (їх чотири) відбувається:

1) Матрикс-векторне множення: $(n + d)$ множень i $(n + d - 1)$ додавань на кожен з n нейронів.

2) Поелементні операції (сигмоїда/tanh та множення): приблизно $3 \cdot n$ FLOPs.

У підсумку для одного гейта:

$$\underbrace{2 \cdot (n + d) \cdot n}_{\text{множення+додавання (матрикс-вектор)}} + 3 \cdot n \approx 2 \cdot (32 + 3) \cdot 32 + 3 \cdot 32 = 2336 \text{ FLOPs.} \quad (39)$$

Оскільки гейтів 4, отримаємо:

$$4 \times 2336 = 9344 \text{ FLOPs на крок.} \quad (40)$$

Вартість одного forward-проходу через всю послідовність відповідно до вікна довжини $k = 3$ кроки обчислюється як:

$$\text{FLOPs}_{\text{forward}} = 9344 \cdot k \approx 9344 \cdot 3 = 28032 \text{ FLOPs.} \quad (41)$$

Зворотне поширення помилки (backward) $\approx 2 \times$ витрат forward обчислюється як:

$$\text{FLOPs}_{\text{backward}} = 2 \cdot 28032 = 56064 \text{ FLOPs.} \quad (42)$$

Підсумок на один зразок за одну епоху обчислюється як:

$$\begin{aligned} \text{FLOPs}_{\text{one sample /one epoch}} &= \text{FLOPs}_{\text{forward}} + \text{FLOPs}_{\text{backward}} = \\ &= 28032 + 56064 = 84096 \text{ FLOPs.} \end{aligned} \quad (43)$$

Загальні FLOPs на тренування для $M = 900$ зразків і $E = 50$ епох визначаються як:

$$\begin{aligned} \text{Total}_{\text{FLOPs}} &= M \cdot E \cdot \text{FLOPs}_{\text{one sample /one epoch}} = 900 \cdot 50 \cdot 84096 \approx \approx \\ &3,78 \cdot 10^9 \text{ FLOPs.} \end{aligned} \quad (44)$$

Якщо застосований GPU/CPU підтримує, скажімо, $5 \text{ TFLOPs} = 5 \cdot 10^{12} \text{ FLOPs/c}$, то

$$\text{Time} = \frac{3,78 \cdot 10^9}{5 \cdot 10^{12}} \approx 0,00076 \text{ (c)} \approx 7,6 \cdot 10^{-4} \text{ (c)}, \quad (45)$$

що є чистими обчисленнями (без врахування пам'яті, I/O, overhead), тобто модель є надзвичайно легкою.

Отже, розроблена LSTM-модель із одним шаром (32 нейрони) містить близько 4,8 тис. параметрів (~19 КБ) та потребує порядку 3,8 GFLOPs для повного навчання на 900 зразках протягом 50 епох, що в чистих арифметичних операціях на сучасному GPU займає мілісекунди; такий рівень обчислювальних витрат робить модель надлегкою й ефективною для інтеграції в інтелектуальну систему стратегічного управління бізнесом із можливістю онлайн-навчання й масштабування.

У табл. 1 наведено практичні рекомендації щодо застосування інтелектуальної системи стратегічного управління бізнесом за допомогою LSTM-мережі.

Таблиця 1

Практичні рекомендації щодо застосування інтелектуальної системи стратегічного управління бізнесом

№	Найменування	Опис
1	Чітке формулювання стратегічних цілей та KPI	Перед впровадженням LSTM-моделі необхідно узгодити ключові цілі бізнесу та конкретні показники ефективності. Це дозволить відслідковувати прогнози моделі у контексті реальних бізнес-результатів і коригувати стратегії відповідно до досягнутих або очікуваних KPI.
2	Підготовка та якість даних	Забезпечте накопичення повного історичного масиву даних про KPI, зовнішні та внутрішні фактори. Проведіть очищення, нормалізацію та заповнення пропусків, а також видалення аномалій. Якісні дані дозволяють LSTM точніше виявляти часові закономірності та знижують ризик «шумових» прогнозів.
3	Налаштування гіперпараметрів та валідація моделі	Проведіть серію експериментів з різними розмірами прихованих шарів, довжиною часових вікон та швидкістю навчання. Використовуйте крос-валідацію та методи підбору гіперпараметрів (Grid Search, Bayesian Optimization) для знаходження оптимальної конфігурації.
4	Інтеграція у бізнес-процеси	Впровадьте модель у вигляді сервісу з REST API або мікросервісу, який автоматично отримує нові дані, запуск прогнозів та передає рекомендації в існуючі інформаційні системи (ERP, BI). Це забезпечить безперервну та прозору взаємодію між AI-моделлю та користувачами.
5	Організація MLOps та моніторинг	та роллбеками. Реалізуйте моніторинг ключових метрик (FLOPs, втрати, точність прогнозів) та бізнес-KPI, щоб вчасно виявляти деградацію моделі й оперативно реагувати.
6	Адаптивне перенавчання та оновлення моделі	Визначте критерії (наприклад, зростання помилки або зміна розподілу вхідних даних), за якими запускається перенавчання. Регулярно оновлюйте модель на збільшеному датасеті, щоб зберігати актуальність прогнозів у мінливому середовищі.
7	Інтерпретованість прогнозів та explainability	Впровадьте методи пояснення (SHAP, LIME), щоб користувачі могли розуміти фактори, що вплинули на прогноз. Це підвищить довіру до системи та допоможе виявляти можливі помилки або зміщення.

8	Управління безпекою та правами доступу	Забезпечте шифрування даних у процесі передачі та зберігання, розграничення прав доступу до сервісу прогнозування, а також аудит дій користувачів і змін у налаштуваннях моделі для дотримання внутрішніх політик безпеки.
9	Масштабування та оптимізація обчислень	прунинг та квантизацію моделі, щоб знизити затримки та витрати ресурсів.
10	Пілотний запуск та поступове розгортання	Проведіть пілот серед обмеженого кола бізнес-користувачів або для одного напрямку, зібравши зворотний зв'язок і проаналізувавши точність прогнозів у реальних умовах. На основі результатів поступово розширюйте зону застосування системи та адаптуйте рекомендації під специфіку різних підрозділів.

Джерело: сформовано автором

Висновки

1. Проведений аналіз підтвердив, що сучасні інтелектуальні системи (IBM watsonx, SAP Analytics Cloud, Microsoft Dynamics 365 Copilot, Oracle NetSuite AI) забезпечують автоматизацію прогнозування, сценаріїв «що-якщо» та рекомендацій, проте мають обмеження (висока вартість, складність впровадження, ризик vendor lock-in). Фактично доведена потреба у створенні доступного та адаптивного рішення для МСБ і великих корпорацій без значних технологічних бар'єрів.

2. Розроблено метод побудови інтелектуальної системи, що ґрунтується на поетапному підході та включає чітке визначення стратегічних цілей і KPI, об'єднання даних з ERP, CRM, BI та зовнішніх джерел у єдиний інформаційний шар, модульну архітектуру з компонентами для foundation-моделей, прогнозування, симуляції сценаріїв та AIGovernance, а також адаптивний інтерфейс із інтерактивними дашбордами, чат- та голосовими асистентами, що надає змогу оперативно генерувати сценарії «що-якщо», автоматизувати прийняття управлінських рішень і швидко реагувати на зміни зовнішнього середовища.

3. Запропонована модель стратегічної ефективності $U(S(t), E(t))$ у поєднанні з LSTM/GRU-прогнозуванням та механізмом адаптивного контролю (коефіцієнт навчання γ) дозволяє генерувати оптимальні стратегії на основі мультикритеріальної оцінки і гібридного аналізу внутрішніх і зовнішніх факторів.

4. За результатами 100 симуляцій було встановлено, що інтелектуальна стратегія забезпечила зростання дисконтованого прибутку на ~20,8 %

при одночасному зниженні ризику ($\sigma_{AI} = 120$ менш ніж $\sigma_{base} = 150$) та підвищенні коефіцієнта Шарпа ($S_{AI} = 12,1$ проти $S_{base} = 8,0$).

5. Використання LSTM-мережі з 32 прихованими нейронами та вікном $k = 3$ забезпечило зростання дисконтованого прибутку на $\sim 29,2\%$ ($J_{LSTM} = 1550$ проти $J_{base} = 1200$), зниження ризику ($\sigma_{LSTM} = 140$) та підвищення точності прогнозів до $\approx 94\ldots 95\%$ при низьких витратах обчислювальних ресурсів ($\sim 3,8$ GFLOPs загалом).

6. Розроблено практичні рекомендації для впровадження, відповідно до яких для успішної інтеграції системи необхідно забезпечити якість даних та їхню нормалізацію, впровадити MLOps (CI/CD, моніторинг метрик і бізнес-KPI), організувати explainability (SHAP, LIME) для підвищення довіри, гарантувати безпеку даних і контроль доступу, а також почати з пілотного запуску й поетапного масштабування, що надає змогу мінімізувати ризики, оперативно адаптувати рішення під змінні умови та поступово нарощувати функціональність без втрати стабільності бізнес-процесів.

Список використаних джерел:

1. Latruwe T., Van der Wee M., Vanleenhove P., Devriese J., Verbrugge S., Colle D. A Long-Term Forecasting and Simulation Model for Strategic Planning of Hospital Bed Capacity. *Operations Research for Health Care*. 2023. Vol. 36. 100375. DOI: <https://doi.org/10.1016/j.orhc.2022.100375>.

2. Eichenseer P., Hans L., Winkler H. A Data-Driven Machine Learning Model for Forecasting Delivery Positions in Logistics for Workforce Planning. *Supply Chain Analytics*. 2025. Vol. 9. 100099. DOI: <https://doi.org/10.1016/j.sca.2024.100099>.

3. Barati S. A System Dynamics Approach for Leveraging Blockchain Technology to Enhance Demand Forecasting in Supply Chain Management. *Supply Chain Analytics*. 2025. Vol. 10. 100115. DOI: <https://doi.org/10.1016/j.sca.2025.100115>.

4. Aljuneidi T., Punia S., Jebali A., Nikolopoulos K. Forecasting and Planning for a Critical Infrastructure Sector during a Pandemic: Empirical Evidence from a Food Supply Chain. *European Journal of Operational Research*. 2024. Vol. 317. Iss. 3. P. 936–952. DOI: <https://doi.org/10.1016/j.ejor.2024.04.009>.

5. Cheng M. N., Wong J. W. K., Cheung C. F., Leung K. H. A Scenario-Based Roadmapping Method for Strategic Planning and Forecasting: A Case Study in a Testing, Inspection and Certification Company. *Technological Forecasting and Social Change*. 2016. Vol. 111. P. 44–62. DOI: <https://doi.org/10.1016/j.techfore.2016.06.005>.

6. Chen Y., Yen P.-Y., Zhang Y., Liu H. Conducting Business Disciplinary Research by Playing Educational Games: A Case of the Forecast Sharing Game in Supply Chain Management. *The International Journal of Management Education*. 2023. Vol. 21. Iss. 2. 100773. DOI: <https://doi.org/10.1016/j.ijme.2023.100773>.

7. Yan E., Tang C., Li Y. A New Workflow for Load Forecasting in Planning Areas: Considering Uncertainties in Building Scenarios. *Sustainable Cities and Society*. 2024. Vol. 108. 105472. DOI: <https://doi.org/10.1016/j.scs.2024.105472>.
8. Liu Q., Yuan F., Wu F., Yu S. Digital Transformation and Management Earnings Forecast. *International Review of Economics & Finance*. 2024. Vol. 96. 103570. DOI: <https://doi.org/10.1016/j.iref.2024.103570>.
9. Rettenmeier M., Petrik D., Möller M., Sauer A. A Modeling Framework and Benchmark for End-of-Life Automotive Traction Battery Pack Forecasting. *Journal of Cleaner Production*. 2025. Vol. 491. 144752. DOI: <https://doi.org/10.1016/j.jclepro.2025.144752>.
10. Hu X., Liu W., Huo H. An Intelligent Network Traffic Prediction Method Based on Butterworth Filter and CNN–LSTM. *Computer Networks*. 2024. Vol. 240. 110172. DOI: <https://doi.org/10.1016/j.comnet.2024.110172>.