Технічні науки

УДК 004.042

Нєлєпін Дмітрій Сергійович

студент

НТУУ «Київський політехнічний інститут імені Ігоря Сікорського»

Науковий керівник:

Амонс Олександр Анатолійович

кандидат технічних наук, доцент,

доцент кафедри інформаційних систем та технологій

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

ORCID: 0000-0002-2621-9188

ІНТЕГРАЦІЯ ПІДСИСТЕМИ ПРОГНОЗУВАННЯ ПОПИТУ В РОЗПОДІЛЕНІ ІНФОРМАЦІЙНІ СИСТЕМИ ТОРГОВИХ МЕРЕЖ

Анотація. *У роботі розглянуто інтеграцію підсистеми прогнозування попиту в розподілені інформаційні системи торговельних мереж. Запропоновано комбіновану архітектуру, що поєднує локальну обробку даних, федеративне узгодження параметрів і мікросервісну взаємодію. Визначено ключові компоненти та технології для забезпечення масштабованості, автономності локальних вузлів і зменшення передавання сирих даних.*

Ключові слова: *прогнозування попиту, торговельні мережі, розподілена інформаційна система, федеративне навчання, мікросервісна архітектура, event-driven архітектура, локальний вузол.*

Вступ. У сучасних умовах цифровізації торговельних мереж прогнозування попиту слід розглядати не лише як аналітичну задачу, а і як складову технічної інфраструктури інформаційної системи підприємства. Практична цінність прогнозу визначається не тільки його точністю, а й здатністю системи своєчасно збирати, передавати, обробляти та узгоджувати дані, що надходять із великої кількості просторово розподілених торгових точок. У таких умовах особливого значення набувають питання організації обчислювальних контурів, інтеграції з джерелами операційних даних, вибору механізмів синхронізації та забезпечення стійкої роботи підсистеми в умовах обмежених мережевих і обчислювальних ресурсів [1; 2].

Централізована обробка даних у подібних системах спрощує контроль і керування моделлю, однак у реальному середовищі вона часто супроводжується перевантаженням каналів передавання, затримками оновлення стану та зниженням адаптивності системи до локальних змін. З цієї причини при побудові підсистем прогнозування попиту дедалі більшої уваги потребують розподілені підходи, у яких локальні вузли виконують попередню обробку даних і частину обчислень безпосередньо на місці їх формування, а до центрального рівня передаються лише узгоджені результати або параметричні оновлення моделі [3; 4].

Математична модель. Методологічною основою є підхід до латентно-факторного подання даних, розглянутий у статті [5; 6] для задач опрацювання великих масивів даних у промислових і рекомендаційних системах. У даному дослідженні зазначений підхід не переноситься безпосередньо, а адаптується до задачі прогнозування попиту в торговельних мережах. На відміну від вихідної двовимірної постановки, у цій роботі він поширюється на багатовимірний простір «магазин – товар – час» [7,8], що дає змогу врахувати просторовий, товарний і часовий аспекти формування попиту. У загальному випадку така постановка подається у

вигляді розрідженого тензора, де кожне значення відповідає обсягу попиту для певного магазину, товару та моменту часу. Для відновлення і прогнозування цих значень використовується низькорангове латентне представлення, у якому попит описується через взаємодію прихованих факторів магазину, товару та часу.

Архітектура розподіленої інформаційної системи. В основі запропонованої архітектури лежить поєднання підходів розподіленої матричної факторизації [3] та федеративного узгодження параметрів [5; 6], які в попередніх дослідженнях застосовувалися переважно для пришвидшення оброблення великих масивів даних у рекомендаційних системах. У межах даної роботи ці ідеї перенесено в інший прикладний контекст – задачу прогнозування попиту в торговельних мережах. Технічна адаптація полягає в тому, що обчислювальний процес організовується на двох рівнях: між окремими магазинами або регіональними вузлами та всередині кожного вузла за підмножинами товарів. У результаті формується комбінована схема, яка поєднує локальність обробки даних із можливістю паралельного виконання обчислень.

На відміну від класичної централізованої моделі, центральний компонент системи не акумулює повний масив первинних спостережень і не виконує повне навчання на сирих даних. Його функціональне призначення полягає в координації раундів обміну, прийманні локальних оновлень, агрегуванні параметрів та підтриманні узгодженого глобального стану моделі. Така організація змінює характер міжвузлової взаємодії: каналами зв'язку передаються не транзакційні набори даних, а компактні результати локального коригування моделі. Це дає змогу зменшити інтенсивність обміну, знизити навантаження на мережеву інфраструктуру та підвищити придатність системи до масштабування зі зростанням кількості торгових точок [4].

З технічного погляду важливою властивістю запропонованої архітектури є підтримка часткової автономності локального вузла. Якщо зв'язок із центральним координатором тимчасово порушується, вузол не втрачає працездатності: він може і далі використовувати локальні дані, виконувати обчислення та будувати прогнози на базі останньої доступної версії параметрів. Після відновлення з'єднання результати, накопичені за період автономної роботи, можуть бути передані до центрального рівня для подальшого узгодження. Завдяки цьому система зберігає функціональність не лише в штатному режимі, а і в умовах нестабільної мережевої взаємодії, затримок синхронізації або короткочасної недоступності центрального вузла [4].

Практична доцільність такої архітектурної організації визначається тим, що вона дозволяє одночасно врахувати кілька критичних вимог реальної торговельної мережі: швидке формування прогнозів, помірні витрати на передавання даних, можливість локального продовження роботи та централізоване узгодження результатів. Саме тому комбінована архітектура може розглядатися не лише як концептуальна схема побудови підсистеми прогнозування, а і як технічно придатна основа для подальшої експериментальної перевірки, зокрема за показниками точності, часової ефективності, комунікаційного навантаження та стійкості до мережевих обмежень.

Аналіз існуючих підходів до побудови розподілених систем прогнозування. У сучасних розподілених інформаційних системах широко застосовуються мікросервісні та event-driven підходи, які сприяють зменшенню зв'язності між компонентами, підвищенню масштабованості та спрощенню розгортання сервісів. Під час переходу від монолітної архітектури до мікросервісної особливого значення набувають CQRS, event-driven architecture та event sourcing, оскільки вони забезпечують організацію обміну даними, журналювання змін і відновлення стану компонентів через

послідовність подій. У контексті прогнозної підсистеми це важливо для побудови керованого контуру обміну повідомленнями між локальними вузлами та центральним координатором [8]. Поєднання мікросервісної архітектури або подієво-орієнтованої інфраструктуру з федеративним навчанням і локальною підготовкою даних представлено в роботі, де підсистема прогнозування розглядається як інтегрований контур із локальними джерелами даних, федеративним шаром узгодження та окремим прогнозним модулем [9].

Інтеграція розподіленої підсистеми прогнозування попиту в інформаційну систему торговельної мережі. У прикладній постановці запропонований метод доцільно реалізовувати як окрему розподілену аналітичну підсистему в складі корпоративної інформаційної системи торговельної мережі. Така підсистема має будуватися за сервісно-орієнтованим або мікросервісним принципом, що забезпечує незалежне масштабування компонентів, гнучкість розгортання та чіткий розподіл функцій між локальними вузлами й центральним координатором. Методологічно цей підхід спирається на адаптацію матричної факторизації та федеративного навчання, які раніше застосовувалися у рекомендаційних системах, але в даній роботі перенесені на задачу прогнозування попиту в торговельних мережах. Додатково доцільність мікросервісної декомпозиції та event-driven взаємодії підтверджується сучасними роботами з інтеграції складних інформаційних систем [10].

На локальному рівні кожен магазин або регіональний вузол доцільно розглядати як edge-вузол, діаграму якого зображено на рис. 1. Він має включати адаптери джерел даних, модуль попередньої обробки, локальне сховище ознак, модуль локального оновлення параметрів моделі, сервіс інференсу та журнал локальних змін. Дані можуть надходити з POS-систем, ERP- і WMS-компонентів, модулів обліку залишків, промоактивностей і календарних сервісів. Для інтеграції з цими джерелами можуть

використовуватися REST/HTTPS API, JDBC/ODBC або періодичне завантаження пакетів у CSV/JSON-форматах. На цьому рівні здійснюються очищення, уніфікація та агрегування даних, після чого до центрального рівня передаються не первинні транзакції, а лише компактні результати локального навчання, що відповідає принципам федеративного підходу [11].

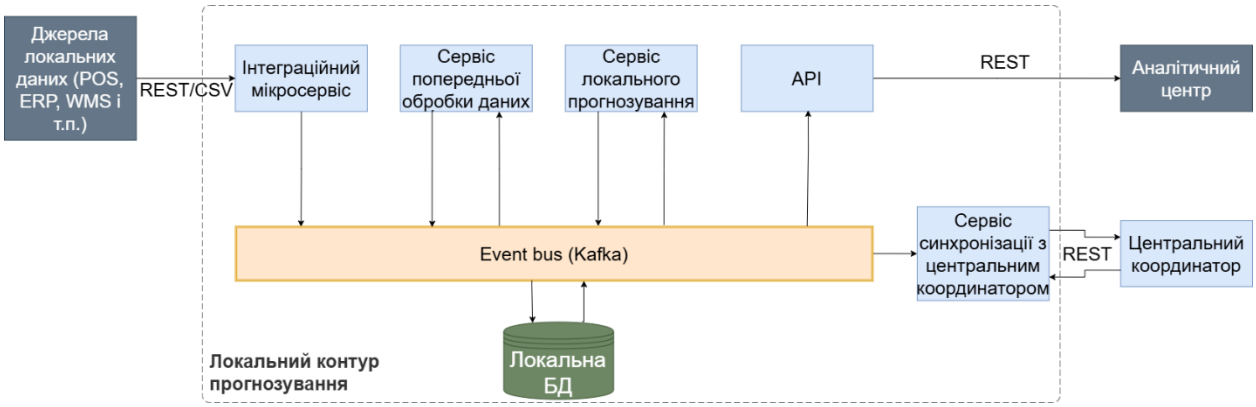

Рис. 1. Діаграма архітектури локального вузла

Взаємодію між компонентами доцільно організовувати через поєднання синхронних та асинхронних каналів. Синхронні виклики потрібні для отримання актуальної версії моделі, керування конфігурацією вузла, службових health-check запитів та повернення короткострокових прогнозів до прикладних модулів. Для цього найбільш придатними є REST/HTTPS або gRPC. Використання gRPC разом із Protocol Buffers є доцільним для внутрішнього високопродуктивного обміну, оскільки забезпечує компактне бінарне подання повідомлень, сувору типізацію контрактів і підтримку ефективної міжсервісної комунікації [12; 13].

Асинхронний контур взаємодії доцільно реалізовувати за подієво-орієнтованим принципом через шину повідомлень. У межах підсистеми прогнозування це дозволяє передавати події про продажі, залишки, промоактивності, готовність прогнозу, оновлення моделі та зміну її версії без жорсткого зв'язування сервісів між собою. Як транспорт для такого контуру доцільно використовувати Apache Kafka, яка поєднує функції event

bus і журналу потокових повідомлень. Для периферійної телеметрії або взаємодії з ресурсно-обмеженими edge-пристроями може додатково застосовуватися MQTT, що є придатним для середовищ із нестабільним або повільним зв'язком [14].

Центральний рівень системи має включати сервер агрегації параметрів, планувальник раундів синхронізації, сервіс керування версіями моделі, API доступу до агрегованих прогнозів, централізований журнал обміну та підсистему моніторингу. Центральний координатор не виконує навчання на сирих даних, а узгоджує локальні оновлення, синхронізує параметри моделі та формує глобальний стан. Тому кожен пакет оновлення повинен містити ідентифікатор вузла, версію моделі, часову мітку, локальні метрики та зміну параметрів. Така організація дозволяє зменшити мережеве навантаження, підвищити масштабованість і підтримати коректну роботу навіть у разі затримок або асинхронного надходження оновлень.

З погляду прикладного використання прогнозів доцільно передбачити два режими їх споживання. Перший є синхронним, коли зовнішній модуль поповнення запасів або замовлення звертається до сервісу прогнозування через API та отримує відповідь безпосередньо. Другий є асинхронним, коли сформовані прогнози або сигнали про перевищення порогів поширюються через подієвий контур до модулів replenishment, pricing, promotion planning або BI-аналітики. Такий підхід забезпечує одночасну підтримку як транзакційних сценаріїв, так і потокового поширення результатів.

Для розгортання підсистеми доцільно використовувати контейнеризацію та оркестрацію, зокрема Kubernetes, що дає змогу ізольовано розгортати окремі сервіси, масштабувати stateless-компоненти та підтримувати stateful-служби через persistent volumes. Окрему роль відіграє спостережуваність системи: крім журналювання помилок, необхідно контролювати затримки синхронізації, розміри черг, час локального інференсу, частку відхилених оновлень, а також якісні метрики

прогнозування, зокрема RMSE, MAE та freshness прогнозу. Для цього доцільно використовувати OpenTelemetry як основу наскрізного збирання traces, metrics і logs.

Отже, технічно інтегрована підсистема прогнозування попиту має будуватися як багаторівнева event-driven мікросервісна інфраструктура, у якій REST/HTTPS забезпечує зовнішню сумісність, gRPC і Protocol Buffers – внутрішній високопродуктивний обмін, Kafka – асинхронну взаємодію між сервісами, MQTT – периферійну телеметрію, Kubernetes – керування контейнеризованими компонентами, а OpenTelemetry – наскрізну спостережуваність. Саме така комбінація найбільш повно відповідає вимогам запропонованої комбінованої архітектури, де локальні вузли зберігають автономність, а центральний рівень виконує координацію та агрегацію без передавання сирих даних.

Висновки. У роботі обґрунтовано доцільність інтеграції підсистеми прогнозування попиту як окремого розподіленого аналітичного контуру в інформаційну систему торговельної мережі. Запропонований підхід поєднує локальне опрацювання даних на рівні магазинів або регіональних вузлів із централізованим узгодженням параметрів моделі, що дає змогу зменшити потребу в передаванні сирих транзакційних даних і підвищити масштабованість системи. Визначено, що ефективна технічна реалізація такої підсистеми має спиратися на мікросервісну та event-driven архітектуру, у якій синхронна взаємодія через REST/HTTPS або gRPC доповнюється асинхронним обміном подіями через шину повідомлень. Особливе значення має підтримка автономності локальних вузлів, які можуть продовжувати роботу навіть за тимчасової недоступності центрального координатора. Таким чином, запропонована архітектура створює технічну основу для побудови стійкої, масштабованої та придатної до практичного впровадження системи прогнозування попиту в умовах розподіленої торговельної інфраструктури.

Література

1. Improving sales forecasting accuracy: a tensor factorization approach with demand awareness / X. Bi et al. *INFORMS journal on computing*. 2022. DOI: https://doi.org/10.1287/ijoc.2021.1147

2. Improved collaborative filtering for cross-store demand forecasting / M. Liang et al. *Computers & industrial engineering*. 2024. P. 110067. DOI: https://doi.org/10.1016/j.cie.2024.110067

3. Large-scale matrix factorization with distributed stochastic gradient descent / R. Gemulla et al. *The 17th ACM SIGKDD international conference*, San Diego, California, USA, 21–24 August 2011. New York, New York, USA, 2011. DOI: https://doi.org/10.1145/2020408.2020426

4. LightFR: lightweight federated recommendation with privacy-preserving matrix factorization / H. Zhang et al. *ACM transactions on information systems*. 2022. DOI: https://doi.org/10.1145/3578361

5. Hordiichuk-Bublivska O. V., Fabri L. P. Matrix factorization of big data in the industrial systems. *Ukrainian journal of information technology*. 2022. Vol. 4, no. 2. P. 68–73. DOI: https://doi.org/10.23939/ujit2022.02.068

6. Distributed singular value decomposition method for fast data processing in recommendation systems / K. Przystupa et al. *Energies*. 2021. Vol. 14, no. 8. P. 2284. DOI: https://doi.org/10.3390/en14082284

7. Koren Y., Bell R., Volinsky C. Matrix factorization techniques for recommender systems. *Computer*. 2009. Vol. 42, no. 8. P. 30–37. DOI: https://doi.org/10.1109/mc.2009.263

8. Amlan Ghosh Event-Driven architectures for microservices: a framework for scalable and resilient rearchitecting of monolithic systems. *International journal on science and technology*. 2025. Vol. 16, no. 1. DOI: https://doi.org/10.71097/ijsat.v16.i1.2498

9. Performance of an end-to-end inventory demand forecasting pipeline using a federated data ecosystem / H. D. Moura et al. *Itise 2024*. Basel Switzerland, 2024. DOI: https://doi.org/10.3390/engproc2024068033

10. Demand forecasting of e-commerce enterprises based on horizontal federated learning from the perspective of sustainable development / J. Li et al. *Sustainability*. 2021. Vol. 13, no. 23. P. 13050. DOI: https://doi.org/10.3390/su132313050

11. Fantacci R., Picano B. Federated learning framework for mobile edge computing networks. *CAAI transactions on intelligence technology*. 2020. Vol. 5, no. 1. P. 15–21. DOI: https://doi.org/10.1049/trit.2019.0049

12. Optimizing microservice communication with grpc and protocol buffers in distributed low-latency api-driven applications / E. Obuse et al. *International journal of multidisciplinary futuristic development*. 2020. Vol. 1, no. 1. P. 45–55. DOI: https://doi.org/10.54660/ijmfd.2020.1.1.45-55

13. Eficiency of REST and grpc realizing communication tasks in microservice-based ecosystems / M. Bolanowski et al. *Frontiers in artificial intelligence and applications*. 2022. DOI: https://doi.org/10.3233/faia220242

14. Optimal distributed MQTT broker and services placement for sdn-edge based smart city architecture / D. Z. Fawwaz et al. *Sensors*. 2022. Vol. 22, no. 9. P. 3431. DOI: https://doi.org/10.3390/s22093431