



DOI 10.28925/2663-4023.2026.34.1390

УДК 004.75:004.89

Читулян Вадим Олегович

Аспірант, старший викладач

Державний університет інформаційно-комунікаційних технологій, Київ, Україна

ORCID:0009-0001-8846-9094

mag5187403@stud.duikt.edu.ua

Поперешняк Світлана Володимирівна

к.ф.-м.н, доцент, доцент кафедри інформатики та програмної інженерії,

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

Державний університет інформаційно-комунікаційних технологій, Київ, Україна

ORCID ID 0000-0002-0531-9809

spopereshnyak@gmail.com

**ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ
ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ДЛЯ КОРПОРАТИВНИХ СТРУКТУР: СУЧАСНИЙ СТАН ТА
ПЕРСПЕКТИВИ**

Анотація. У роботі досліджено сучасний стан та перспективи розвитку інформаційних технологій автоматизованого проєктування обчислювальних систем для корпоративних структур. Актуальність дослідження зумовлена зростанням складності корпоративних інформаційних систем, поширенням розподілених і хмарних архітектур та необхідністю одночасного врахування вимог до продуктивності, масштабованості, кібербезпеки, відмовостійкості й живучості. Проведено аналітичний огляд і порівняння сучасних підходів за трьома взаємопов'язаними напрямками: застосування машинного навчання для підтримки вибору архітектурних рішень і прогнозування їх характеристик; інтеграція тестування навантаження в ітеративний цикл проєктування; забезпечення кібербезпеки та відмовостійкості розподілених корпоративних систем. Розглянуто можливість використання ML-класифікаторів, засобів пояснюваного штучного інтелекту SHAP, AutoML та сурогатного моделювання для попереднього вибору й оцінювання архітектурних конфігурацій. Проаналізовано підходи до тестування навантаження та обґрунтовано доцільність використання його результатів як зворотного зв'язку для ітеративної верифікації архітектурних рішень. Систематизовано підходи до забезпечення кібербезпеки, відмовостійкості та живучості й показано можливість їх урахування як критеріїв багатокритеріального вибору конфігурації. Встановлено, що наявні методи та інструментальні засоби переважно орієнтовані на вирішення окремих задач і недостатньо інтегровані в наскрізний процес автоматизованого проєктування. За результатами проведеного аналізу запропоновано концептуальну формалізацію такого процесу, що послідовно поєднує формалізацію бізнес- і технічних вимог, ML-вибір архітектурного класу, прогнозування характеристик конфігурації, її ітеративну верифікацію за результатами тестування навантаження та багатокритеріальний вибір захищеного й відмовостійкого рішення. Визначено перспективи подальших досліджень, пов'язані з розробленням та експериментальною верифікацією відповідних методів і інформаційних технологій автоматизованого проєктування корпоративних обчислювальних систем.

Ключові слова: автоматизоване проєктування; корпоративні обчислювальні системи; машинне навчання; архітектурні рішення; SHAP; AutoML; сурогатне моделювання; тестування навантаження; кібербезпека; відмовостійкість; живучість.

ВСТУП

Постановка проблеми. Корпоративні обчислювальні системи є критичною складовою інформаційної інфраструктури сучасних організацій, забезпечуючи виконання бізнес-процесів, обробку та зберігання даних, взаємодію між структурними підрозділами й доступ користувачів до інформаційних



сервісів. Зростання масштабів і складності таких систем, поширення хмарних і розподілених архітектур, динамічність навантаження та підвищення вимог до продуктивності, безпеки й відмовостійкості істотно ускладнюють процес їх проєктування. За цих умов якість архітектурних рішень значною мірою залежить від досвіду фахівців, тоді як можливості формалізованого вибору, автоматизованого оцінювання та ранньої верифікації проєктних рішень залишаються обмеженими [1, 2].

Традиційний процес проєктування корпоративної IT-інфраструктури охоплює формування вимог, вибір архітектури та конфігурації, реалізацію, тестування і розгортання системи. Незважаючи на значний рівень автоматизації окремих операцій, прийняття ключових архітектурних рішень, оцінювання їх відповідності бізнес-вимогам та перевірка очікуваних характеристик системи часто залишаються розділеними між різними етапами й інструментальними засобами. Унаслідок цього невідповідність обраної архітектури вимогам до продуктивності, масштабованості, відмовостійкості або супроводжуваності може бути виявлена вже на пізніх етапах реалізації та експлуатації, що особливо суттєво для складних розподілених і багатоорендних (multi-tenant) корпоративних систем [3].

Водночас розвиток сучасних інформаційних технологій створює передумови для переходу від автоматизації окремих операцій до комплексної підтримки процесу проєктування корпоративних обчислювальних систем. Методи машинного навчання можуть використовуватися для аналізу вхідних параметрів, підтримки вибору архітектурних рішень та прогнозування характеристик систем [4, 5]. Засоби тестування навантаження та моніторингу дають можливість включати оцінювання продуктивності до ітеративного циклу проєктування [6]. Методи забезпечення кібербезпеки, відмовостійкості та живучості розподілених систем створюють основу для врахування відповідних вимог уже під час формування архітектури, а не лише на етапі її подальшої експлуатації [7].

Однак зазначені технології розвиваються переважно як окремі напрями та вирішують різні часткові задачі проєктування. Недостатньо дослідженим залишається їх узгоджене використання в межах цілісного процесу автоматизованого проєктування, який забезпечував би послідовний перехід від формалізованих бізнес-вимог до вибору архітектурного рішення, прогнозування його характеристик, верифікації під навантаженням та формування відмовостійкої й захищеної конфігурації. Тому актуальним є систематизація сучасних інформаційних технологій автоматизованого проєктування корпоративних обчислювальних систем, визначення рівня їх інтеграції на різних етапах проєктування та виявлення напрямів подальшого розвитку комплексних засобів підтримки архітектурних рішень.

Аналіз останніх досліджень і публікацій. Сучасні дослідження автоматизації проєктування корпоративних обчислювальних систем охоплюють декілька взаємопов'язаних напрямів. Методи машинного навчання застосовуються для прогнозування характеристик обчислювальних систем і підтримки прийняття технічних рішень, а засоби пояснюваного штучного інтелекту, зокрема SHAP, дають змогу інтерпретувати результати роботи ML-моделей [4], [8]–[10]. Розвиток засобів тестування навантаження, моніторингу та інтелектуального керування ресурсами забезпечує можливості автоматизованого оцінювання продуктивності й адаптації обчислювальних конфігурацій [6], [13]–[15]. Паралельно розвиваються методи забезпечення кібербезпеки, відмовостійкості та живучості розподілених систем, включаючи формалізовані критерії безпеки, архітектуру нульової довіри та постквантові механізми захисту [7], [17].

Водночас зазначені підходи переважно спрямовані на розв'язання окремих задач вибору, оцінювання, експлуатації або захисту обчислювальних систем. Недостатньо дослідженим залишається їх узгоджене використання в межах наскрізного процесу автоматизованого проєктування, який пов'язував би бізнес- і технічні вимоги з вибором архітектурного рішення, прогнозуванням та верифікацією його характеристик, а також формуванням захищеної й відмовостійкої конфігурації. Це визначає необхідність систематизації наявних підходів, виявлення їх функціональних взаємозв'язків та визначення перспектив інтеграції в єдину інформаційну технологію автоматизованого проєктування корпоративних обчислювальних систем.

Метою дослідження є систематизація сучасного стану інформаційних технологій автоматизованого проєктування обчислювальних систем для корпоративних структур, виявлення не вирішених питань та визначення перспектив їх інтеграції в цілісний процес, що охоплює вибір архітектурних рішень, прогнозування їх характеристик, ітеративну верифікацію продуктивності та врахування вимог до кібербезпеки, відмовостійкості й живучості системи.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. проаналізувати можливості методів машинного навчання, пояснюваного штучного інтелекту та прогнозування характеристик для підтримки вибору архітектурних рішень;
2. дослідити підходи до інтеграції тестування навантаження та оцінювання продуктивності в ітеративний процес архітектурного проєктування;

3. проаналізувати підходи до врахування кібербезпеки, відмовостійкості та живучості під час формування архітектури корпоративних обчислювальних систем;
4. узагальнити виявлені прогалини та визначити перспективи інтеграції розглянутих технологій у цілісний процес автоматизованого проектування.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Застосування машинного навчання для класифікації архітектурних рішень.

Застосування методів машинного навчання до задач вибору та класифікації IT-архітектур є відносно новим напрямом. Значна частина досліджень у цій галузі зосереджена на прогнозуванні продуктивності та інших характеристик обчислювальних систем, тоді як використання машинного навчання безпосередньо для підтримки вибору архітектурних рішень досліджено меншою мірою [8].

Одним із поширених підходів до інтерпретації результатів ML-моделей є SHAP (Shapley Additive Explanations), що ґрунтується на використанні значень Шеплі [9]. Його ефективність для пояснення результатів роботи ML-моделей продемонстровано, зокрема, у контексті управління датацентрами, де SHAP-аналіз дає змогу визначати вплив окремих параметрів системи на сформований прогноз та підвищувати інтерпретованість автоматизованих рекомендацій [4]. Водночас у відповідному дослідженні не розглядається задача класифікації архітектурних рішень за бізнес-параметрами організації, оскільки аналіз здійснюється переважно на операційному рівні вже розгорнутих систем.

За результатами аналізу розглянутих підходів узагальнену послідовність використання ML-засобів для підтримки вибору архітектурних рішень можна подати у вигляді схеми, наведеної на рис. 1.

У роботі [8] запропоновано систему MLASP (Machine Learning Assisted System Performance), яка використовує ML-моделі для прогнозування ключових показників продуктивності обчислювальних систем. За наведеними авторами результатами, точність прогнозування для досліджених середовищ сягає 95 %. Водночас питання вибору архітектурного класу системи безпосередньо на етапі проектування в цій роботі не розглядається.

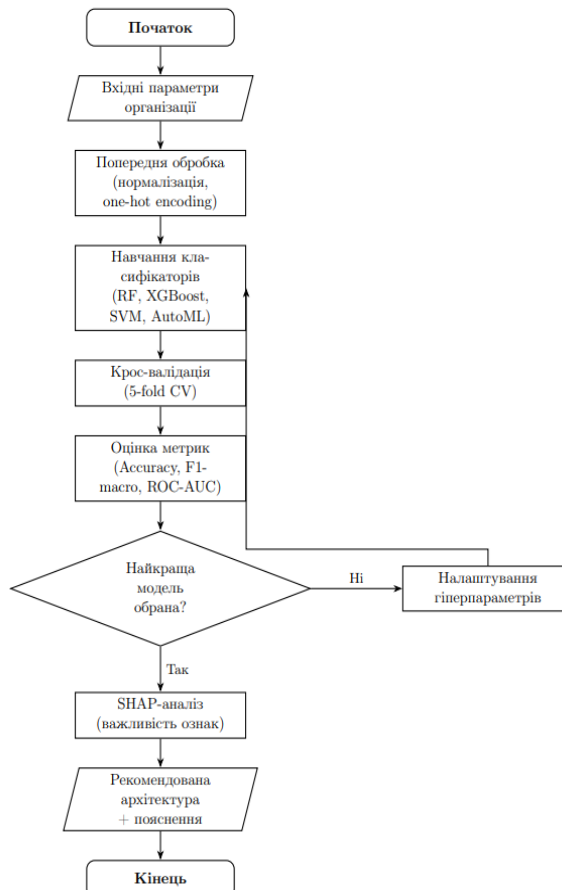


Рис. 1. Схема ML-конвеєру класифікації архітектурних рішень



Поряд із класифікацією розвивається напрям прогнозування продуктивності обраної конфігурації методами машинного навчання. Порівняльні дослідження алгоритмів, зокрема градієнтного бустингу та рекурентних нейронних мереж, показують можливість їх використання для прогнозування показників спроможності та навантаження на різних часових горизонтах [10]. Перспективним у контексті автоматизованого проєктування є поєднання класифікації архітектурних рішень із сурогатним прогнозуванням продуктивності. Сурогатна модель у цьому випадку може використовуватися для попереднього оцінювання характеристик альтернативних конфігурацій без необхідності повного розгортання та проведення ресурсомістких натурних випробувань кожного варіанта.

Додаткові можливості створюють засоби автоматизованого формування та налаштування ML-конвеєрів (AutoML), які дають змогу автоматизувати вибір моделей, їх гіперпараметрів та окремих етапів підготовки даних. Зокрема, TROT реалізує автоматизований пошук і оптимізацію ML-конвеєрів на основі генетичного програмування, поєднуючи операції попередньої обробки ознак і моделі машинного навчання [19]. Такий підхід потенційно може бути використаний для автоматизації вибору моделей під час класифікації архітектурних рішень за сукупністю корпоративних параметрів.

Окремого розгляду потребує формалізація експертних знань для формування навчальних вибірок. Дослідження застосування машинного навчання для автоматизованого оцінювання управління бізнес-процесами в IT-підприємствах підтверджують перспективність перетворення експертних оцінок на структуровані дані для слабко формалізованих предметних областей [11]. Проте методологія формування та розмітки наборів даних саме для задачі вибору архітектури корпоративної обчислювальної системи за сукупністю бізнес- і технічних параметрів у розглянутих джерелах детально не представлена.

Проведений аналіз показує, що в розглянутих дослідженнях недостатньо представлено підходи, які забезпечують наскрізний зв'язок між бізнес-параметрами корпоративної системи, ML-класифікацією архітектурних рішень, поясненням отриманих рекомендацій та попереднім прогнозуванням характеристик обраної конфігурації. Це визначає перспективність інтеграції засобів машинного навчання, пояснюваного штучного інтелекту та сурогатного моделювання в інформаційні технології автоматизованого проєктування корпоративних обчислювальних систем.

Ітеративне проєктування з вбудованим тестуванням навантаження.

Тестування навантаження традиційно розглядалося переважно як окрема фаза, що виконується після реалізації основних компонентів системи. Водночас розвиток DevOps-підходів, засобів автоматизації тестування та моніторингу сприяє перенесенню перевірки характеристик продуктивності на більш ранні етапи життєвого циклу програмних і обчислювальних систем [12]. Це створює передумови для використання результатів тестування не лише для контролю вже реалізованої системи, а й для обґрунтування та коригування архітектурних рішень у процесі проєктування.

Проведений аналіз дає змогу виділити перспективний напрям розвитку технологій автоматизованого проєктування, за якого тестування навантаження вбудовується безпосередньо в ітеративний цикл формування та верифікації архітектури. Узагальнену послідовність такого циклу наведено на рис. 2. У цьому випадку результати навантажувального тестування — зокрема показники латентності, пропускну здатності, використання ресурсів і точка насичення — можуть розглядатися не лише як характеристики продуктивності, а як формалізовані критерії прийнятності поточної архітектурної конфігурації.

У роботі [6] досліджено застосування ML-моделей спільно із засобами спостережуваності для автоматизованого провізйонування хмарної інфраструктури. Інтеграція засобів k6 та Grafana забезпечує зворотний зв'язок між показниками продуктивності та конфігурацією обчислювальних ресурсів. Однак основну увагу приділено управлінню вже розгорнутою інфраструктурою, тоді як використання результатів тестування навантаження як критерію вибору або коригування архітектурного рішення на ранніх етапах проєктування безпосередньо не розглядається.

У дослідженнях, присвячених автоматизації проєктування інфраструктури, розглядаються підходи до автоматизованого формування архітектурних рішень, за яких програмні засоби можуть інтерпретувати проєктні обмеження та формувати або моделювати топологію інфраструктури [5]. Такі підходи сприяють скороченню часу розгортання та зменшенню кількості конфігураційних помилок, однак механізми використання результатів тестування навантаження для ітеративної верифікації сформованої архітектури залишаються недостатньо формалізованими.

Подальший розвиток цього напрямку пов'язаний із застосуванням методів штучного інтелекту для автоматизованого тестування та керування ресурсами. У роботі [13] запропоновано AI-систему на основі агентів навчання з підкріпленням для автоматизованого тестування навантаження та масштабування ресурсів у хмарних середовищах. Такий підхід дає змогу виявляти вузькі місця та адаптивно змінювати

обсяг ресурсів, проте питання використання отриманих результатів для коригування архітектурних рішень саме на етапі проектування не є центральним предметом дослідження.

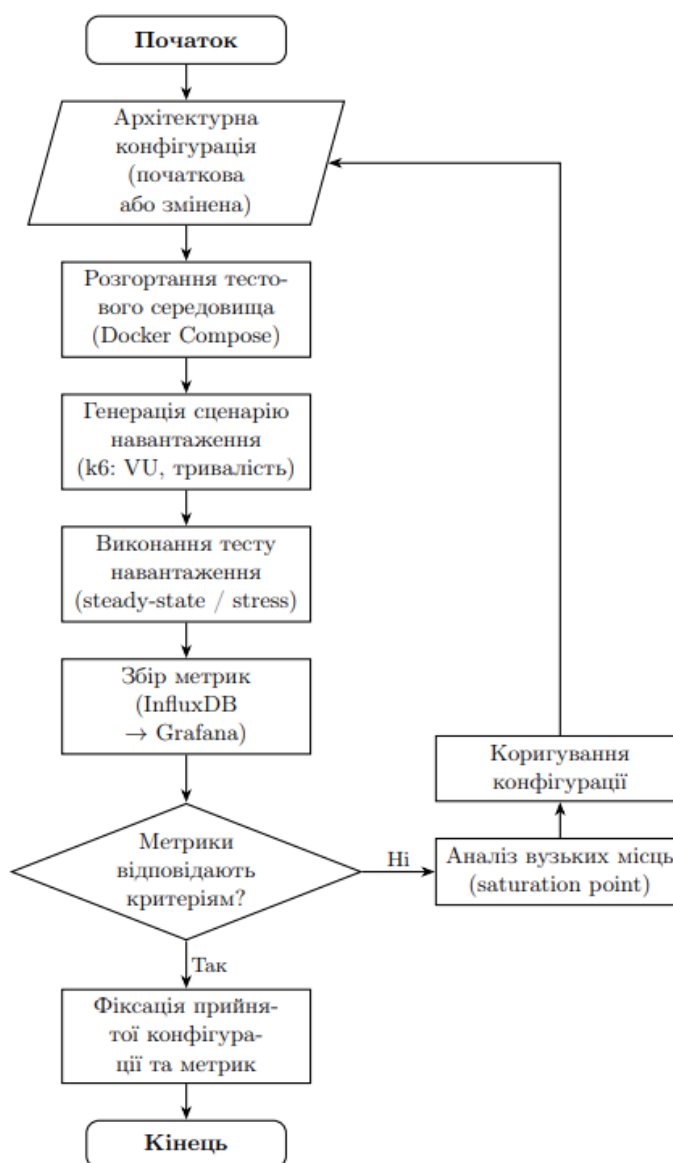


Рис. 2. Схема ітеративного циклу проектування з вбудованим тестуванням навантаження

У роботі [14] запропоновано гібридну платформну архітектуру для AI-augmented performance engineering, у якій темпоральні нейронні мережі використовуються для прогнозування потреб у ресурсах, а агенти навчання з підкріпленням — для оптимізації параметрів системи. Запропонований підхід орієнтований переважно на рівень функціонування розгорнутих систем, тому можливості його використання безпосередньо в ітеративному циклі архітектурного проектування потребують подальшого дослідження.

ML-driven predictive scaling для Kubernetes-оркестрації розглянуто у [15]. За наведеними авторами результатами, гібридне поєднання gradient boosting, LSTM та reinforcement learning забезпечує прогнозування потреби в ресурсах на горизонті до 30 хвилин. Отримані результати підтверджують можливість використання ML-підходів для прогнозного управління ресурсами, однак безпосередній зв'язок між результатами навантажувального тестування та прийняттям рішень на рівні архітектурного проектування в цій роботі не формалізовано.

Із тестуванням навантаження безпосередньо пов'язане забезпечення функціональної стійкості корпоративної обчислювальної системи – її здатності зберігати необхідний рівень виконання функцій за



змінних умов функціонування та навантаження. У роботі [16] запропоновано моделі та методи оцінювання функціональної стійкості інформаційних систем за показниками доступності, відмовостійкості та ресурсовитратності. Водночас використання таких показників як критеріїв прийнятності архітектурної конфігурації безпосередньо в ітеративному процесі проєктування залишається недостатньо дослідженим.

Таким чином, у розглянутих дослідженнях достатньо розвинені окремі засоби автоматизованого тестування, моніторингу, прогнозування навантаження та адаптивного керування ресурсами. Водночас недостатньо дослідженим залишається їх використання в межах **замкненого циклу архітектурного проєктування**, у якому результати навантажувального тестування безпосередньо визначають прийнятність поточної конфігурації та необхідність її подальшого коригування. Формалізація такого зворотного зв'язку створює передумови для автоматизованого пошуку архітектурних рішень, що відповідають заданим вимогам до продуктивності та функціональної стійкості.

Кібербезпека та відмовостійкість розподілених корпоративних систем.

Проєктування сучасних корпоративних обчислювальних систем здійснюється в умовах зростання складності розподілених архітектур, розширення поверхні атак та підвищення вимог до безперервності функціонування критичних сервісів. За таких умов кібербезпека, відмовостійкість і живучість системи мають розглядатися не лише як характеристики її подальшої експлуатації, а й як складові архітектурного рішення, що повинні враховуватися вже на етапі проєктування. Це потребує переходу від подальшого налаштування окремих механізмів захисту та резервування до формалізованого врахування відповідних вимог під час вибору структури й параметрів корпоративної обчислювальної системи.

У роботі [7] запропоновано метод організації функціонування розподілених систем на основі автоматичного застосування формалізованих критеріїв безпеки. Такий підхід створює можливість системного врахування вимог до захищеності компонентів розподіленої системи, однак орієнтований передусім на організацію функціонування та конфігурування системи, тоді як задача вибору архітектури безпосередньо за сукупністю бізнес-, технічних і безпекових вимог потребує подальшої формалізації.

Важливою характеристикою розподіленої корпоративної системи є її здатність зберігати виконання критичних функцій у разі відмови або недоступності частини компонентів. У цьому контексті поряд із відмовостійкістю доцільно враховувати живучість та адаптивність системи, зокрема можливість її граційної (плавної) деградації, за якої погіршення умов функціонування або часткова втрата ресурсів не спричиняють миттєвої втрати всієї функціональності. Включення відповідних характеристик до множини проєктних критеріїв створює можливість оцінювати альтернативні архітектурні рішення не лише за продуктивністю та вартістю, а й за здатністю забезпечувати критичні функції за несприятливих умов.

Окремий напрям розвитку засобів захисту пов'язаний з архітектурою нульової довіри (Zero Trust) та постквантовою криптографією. Принципи Zero Trust передбачають мінімізацію неявної довіри між компонентами та посилення контролю взаємодії в розподіленому середовищі. У роботі [17], зокрема, досліджено методи динамічної оптимізації постквантових криптографічних протоколів залежно від умов мережі та обчислювальних можливостей вузлів. Реалізація таких механізмів створює додаткове навантаження на обчислювальні та комунікаційні ресурси, тому їх урахування під час архітектурного проєктування потребує узгодження вимог до захищеності, продуктивності та доступних ресурсів.

За наявності декількох взаємопов'язаних критеріїв вибір захищеної та відмовостійкої конфігурації може розглядатися як **багатокритеріальна задача**. До множини критеріїв можуть входити рівень захищеності, доступність, живучість, продуктивність, ресурсовитратність і вартість реалізації. Така постановка створює основу для переходу від переважно експертного вибору засобів захисту та резервування до формалізованого порівняння альтернативних архітектурних конфігурацій. Можливість використання оптимізаційних підходів до розподілу ресурсів безпеки в розподілених системах розглядається, зокрема, у [7].

Водночас аналіз розглянутих досліджень показує, що методи забезпечення кібербезпеки, відмовостійкості та живучості переважно розробляються для окремих етапів створення або експлуатації розподілених систем. Недостатньо дослідженою залишається їх інтеграція безпосередньо в процес автоматизованого архітектурного проєктування, у якому вимоги до безпеки, відмовостійкості та живучості використовувалися б як формалізовані критерії вибору структури й параметрів системи поряд із бізнес-вимогами та характеристиками продуктивності. Перспективним напрямом є структурно-параметричне проєктування захищених і відмовостійких корпоративних обчислювальних систем на основі багатокритеріального оцінювання та оптимізації альтернативних архітектурних конфігурацій.

**Зведений аналіз прогалин.**

Результати аналізу трьох взаємопов'язаних напрямів інформаційних технологій автоматизованого проектування корпоративних обчислювальних систем узагальнено в табл. 1. Порівняння охоплює досягнуті результати та невирішені питання, що визначають перспективи подальшої інтеграції розглянутих підходів.

Зведений аналіз показує, що в кожному з розглянутих напрямів сформовано достатньо розвинений набір методів та інструментальних засобів, однак їх застосування переважно орієнтоване на вирішення окремих задач або етапів життєвого циклу корпоративної обчислювальної системи. Спільною невирішеною проблемою залишається інтеграція цих можливостей у цілісний процес автоматизованого проектування, що забезпечував би перехід від формалізованих бізнес-вимог до обґрунтованого вибору архітектури, прогнозування її характеристик, ітеративної верифікації продуктивності та формування захищеної, відмовостійкої конфігурації.

Саме ця інтеграційна прогалина визначає основний напрям подальшого розвитку інформаційних технологій автоматизованого проектування корпоративних обчислювальних систем.

Таблиця 1

Зведений аналіз досліджень і виявлених прогалин

№	Напрямок	Досягнуті результати	Виявлена прогалина
1	ML-класифікація архітектурних рішень	ML-класифікація та прогнозування характеристик; пояснення результатів засобами SHAP; автоматизований підбір і налаштування ML-моделей	Недостатньо досліджено наскрізний зв'язок між бізнес-параметрами, ML-класифікацією архітектурних рішень, поясненням рекомендацій та попереднім прогнозуванням характеристик конфігурації
2	Ітеративне проектування з тестуванням навантаження	Автоматизоване тестування; засоби спостережуваності; AI-assisted provisioning; прогнозування навантаження та адаптивне масштабування ресурсів	Недостатньо формалізовано використання результатів тестування навантаження як критеріїв прийнятності архітектурної конфігурації та зворотного зв'язку в ітеративному циклі проектування
3	Кібербезпека, відмовостійкість і живучість	Формалізовані критерії безпеки; Zero Trust; постквантова криптографія; адаптивні механізми захисту; підходи до забезпечення відмовостійкості та живучості	Недостатньо досліджено інтеграцію вимог до безпеки, відмовостійкості та живучості як формалізованих критеріїв структурно-параметричного вибору архітектури

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ**Порівняльний аналіз підходів до ML-класифікації архітектурних рішень**

Вибір архітектурного рішення корпоративної обчислювальної системи може бути представлений як задача відображення множини формалізованих бізнес- і технічних параметрів на множину допустимих архітектурних класів. Вектор вхідних параметрів подамо у вигляді

$$\mathbf{x} = (x_1, x_2, \dots, x_m), \quad (1)$$

де x_i – параметри, що характеризують вимоги до корпоративної обчислювальної системи, зокрема кількість користувачів, очікуване навантаження, вимоги до доступності та масштабованості, особливості розподіленості системи, бюджетні обмеження та вимоги до захисту інформації.

За такого представлення задача автоматизованого вибору архітектурного класу може бути формалізована як задача багатокласової класифікації:

$$\hat{a} = f_{ML}(\mathbf{x}; \boldsymbol{\theta}), \quad \hat{a} \in \mathcal{A}, \quad (2)$$

де $\mathcal{A} = \{a_1, a_2, \dots, a_k\}$ – множина допустимих класів архітектурних рішень; f_{ML} – модель машинного навчання; $\boldsymbol{\theta}$ – параметри моделі; $\hat{a}$ – рекомендований архітектурний клас.

Специфікою розглядуваної задачі є змішана природа вхідних параметрів, які можуть бути числовими, категоріальними та бінарними, багатокласовий характер цільової змінної, а також необхідність формування навчальної вибірки на основі формалізованих архітектурних рішень. Окремою вимогою є пояснюваність отриманої рекомендації, оскільки в корпоративному середовищі результат



автоматизованої класифікації має не лише визначати рекомендований клас архітектури, а й давати можливість встановити, які бізнес- і технічні параметри найбільше вплинули на сформоване рішення.

Порівняльні характеристики основних підходів машинного навчання, які потенційно можуть бути використані для розв'язання такої задачі, наведено в табл. 2.

Таблиця 2

Порівняльні характеристики ML-підходів для класифікації архітектурних рішень

Метод	Робота з табличними даними	Нелінійні залежності	Пояснення засобами SHAP	Робота з дисбалансом класів	Особливості застосування
Random Forest	Так	Так	TreeExplainer	Підтримується засобами балансування ваг класів	Ансамблева модель; придатна для різнорідних структурованих ознак
XGBoost	Так	Так	TreeExplainer	Підтримується налаштуванням ваг класів	Гradientний бустинг; підтримує регуляризацію та гнучке налаштування параметрів
SVM	Так	Через функцію ядра	KernelExplainer	Підтримується налаштуванням ваг класів	Потребує масштабування ознак; результат залежить від вибору ядра та його параметрів
Logistic Regression	Так	Обмежено без додаткового перетворення ознак	LinearExplainer	Підтримується вагами класів	Інтерпретована базова модель; ефективна насамперед для відносно простих залежностей
AutoML	Залежить від сформованого конвеєра	Залежить від обраної моделі	Залежить від обраної моделі	Залежить від сформованого конвеєра	Автоматизує вибір моделі, гіперпараметрів та окремих етапів підготовки даних
Neural Network	Так	Так	Залежить від архітектури та засобу пояснення	Потребує налаштування процесу навчання	Доцільність зростає за достатнього обсягу навчальних даних і складних залежностей

Порівняння показує, що вибір конкретного ML-методу залежить від структури сформованого набору даних, кількості навчальних прикладів, співвідношення класів, складності залежностей між ознаками та вимог до пояснюваності результатів. Для задач із переважно структурованими табличними даними перспективними кандидатами є ансамблеві деревоподібні моделі, зокрема Random Forest та XGBoost, для яких доступні спеціалізовані засоби SHAP-інтерпретації [9, 18]. Водночас визначення найбільш ефективного класифікатора потребує експериментального порівняння моделей на сформованому наборі даних архітектурних рішень, тому без такого порівняння перевага окремого алгоритму не може вважатися встановленою.

Важливою складовою автоматизованого вибору архітектури є пояснення отриманої рекомендації. Для цього можуть використовуватися значення SHAP, що характеризують внесок окремих вхідних ознак у результат роботи ML-моделі [9]. Для окремого прогнозу адитивне представлення пояснення може бути записано у вигляді

$$f(\mathbf{x}) = \phi_0 + \sum_{i=1}^m \phi_i, \quad (3)$$

де ϕ_0 – базове значення прогнозу моделі; ϕ_i – внесок i -ї ознаки у сформований результат.

У контексті автоматизованого проєктування таке представлення дає змогу не лише отримати рекомендований архітектурний клас $\hat{a}$, а й визначити вплив окремих бізнес- і технічних параметрів на сформовану рекомендацію. Локальні SHAP-пояснення можуть використовуватися для аналізу конкретного архітектурного рішення, тоді як агрегування значень SHAP для множини спостережень дає змогу оцінювати загальний вплив окремих параметрів на результати класифікації. Це підвищує



інтерпретованість автоматизованого вибору та створює можливість експертної перевірки його відповідності вихідним вимогам.

Визначення архітектурного класу є лише першим етапом формування проєктного рішення. Для подальшого вибору конкретної конфігурації необхідно оцінити її очікувані характеристики ще до повного розгортання системи. Перспективним для цього є використання сурогатного моделювання, за якого ресурсомістке оцінювання реальної конфігурації частково замінюється прогнозом моделі, побудованої на основі попередньо отриманих даних. У загальному вигляді прогнозовану характеристику конфігурації можна представити як

$$\hat{y} = g(\mathbf{x}; \mathbf{c}), \quad (4)$$

де g – сурогатна модель; $\mathbf{x}$ – вектор вихідних бізнес- і технічних параметрів; $\mathbf{c}$ – вектор параметрів обраної архітектурної конфігурації; $\hat{y}$ – прогнозоване значення показника, наприклад латентності, пропускну здатності або використання обчислювальних ресурсів.

Для задач, у яких поряд із прогнозованим значенням необхідно враховувати невизначеність прогнозу, перспективним є застосування гаусівської процесової регресії. У цьому випадку прогноз для заданої конфігурації може бути представлений розподілом

$$y(\mathbf{x}; \mathbf{c}) \sim \mathcal{N}(\mu(\mathbf{x}; \mathbf{c}), \sigma^2(\mathbf{x}; \mathbf{c})), \quad (5)$$

де $\mu(\mathbf{x}; \mathbf{c})$ – прогнозоване середнє значення характеристики, а $\sigma^2(\mathbf{x}; \mathbf{c})$ – дисперсія, що характеризує невизначеність прогнозу. Врахування невизначеності є важливим для автоматизованого проєктування, оскільки дає змогу виділяти конфігурації, для яких прогнозовані характеристики потребують додаткової перевірки шляхом моделювання або навантажувального тестування.

Таким чином, перспективна організація першого етапу автоматизованого проєктування передбачає послідовне використання ML-класифікації для попереднього вибору архітектурного класу, SHAP-аналізу для пояснення сформованої рекомендації та сурогатного моделювання для прогнозування характеристик конкретної конфігурації. При цьому результати прогнозування не замінюють експериментальної перевірки характеристик системи, а використовуються для попереднього відбору перспективних конфігурацій. Подальша верифікація їх відповідності заданим вимогам потребує включення тестування навантаження безпосередньо до ітеративного циклу проєктування.

Порівняльний аналіз підходів до тестування навантаження в циклі проєктування

Результати аналізу показують поступовий перехід від використання тестування навантаження переважно для перевірки вже реалізованої системи до його інтеграції з процесами безперервної доставки, моніторингу та адаптивного керування ресурсами. У контексті автоматизованого проєктування принципним є подальший розвиток цього підходу — використання результатів навантажувального тестування як зворотного зв'язку для оцінювання та коригування архітектурної конфігурації ще до її промислового розгортання.

Нехай конкретна архітектурна конфігурація, сформована після попереднього вибору архітектурного класу, визначається вектором

$$\mathbf{c}^{(k)} = (c_1^{(k)}, c_2^{(k)}, \dots, c_n^{(k)}), \quad (6)$$

де $c_j^{(k)}$ – значення j -го параметра конфігурації на k -й ітерації проєктування. До таких параметрів можуть належати кількість і характеристики обчислювальних вузлів, обсяг виділеної пам'яті, кількість екземплярів сервісів, параметри балансування навантаження, конфігурація сховищ даних та інші характеристики, що можуть змінюватися в процесі пошуку прийнятного архітектурного рішення.

Для кожної конфігурації $\mathbf{c}^{(k)}$ виконується тестування за визначеним сценарієм навантаження, у результаті якого формується вектор вимірних характеристик

$$\mathbf{m}^{(k)} = (L_{p95}^{(k)}, L_{p99}^{(k)}, T^{(k)}, U^{(k)}, S^{(k)}), \quad (7)$$

де $L_{p95}^{(k)}$ і $L_{p99}^{(k)}$ – відповідно 95-й та 99-й процентилі латентності; $T^{(k)}$ – пропускну здатність; $U^{(k)}$ – рівень використання критичного обчислювального ресурсу; $S^{(k)}$ – характеристика точки насичення системи.

Порівняльні характеристики основних підходів до тестування навантаження та управління продуктивністю наведено в табл. 3.



Таблиця 3

Порівняльні характеристики підходів до тестування навантаження в контексті архітектурного проєктування

Підхід	Типові засоби	Етап застосування	Основні характеристики	Зв'язок архітектурним рішенням ³
Класичне post-deployment testing	JMeter та аналогічні засоби	Після розгортання системи	Латентність, пропускна здатність, кількість помилок	Результати використовуються переважно для виявлення проблем уже реалізованої системи
CI/CD-інтегроване тестування	Gatling, k6	У процесі безперервної інтеграції та доставки	Латентність, throughput, error rate, ресурсні метрики	Забезпечує автоматизований контроль характеристик, але не обов'язково змінює архітектурне рішення
AI-assisted testing	Засоби навантажувального тестування та ML/RL-механізми [13]	Переважно експлуатація та адаптивне керування	Метрики продуктивності, прогнозовані характеристики, виявлення вузьких місць	Результати можуть використовуватися для адаптивного керування ресурсами
Predictive scaling	Kubernetes, ML-моделі [15]	Експлуатація	Прогноз навантаження та потреби в ресурсах	Орієнтоване переважно на масштабування вже розгорнутої системи
Перспективне design-time testing	Ізольоване/контейнеризоване тестове середовище, засоби генерації навантаження та моніторингу	Ітеративне архітектурне проєктування	p95/p99, пропускна здатність, використання ресурсів, точка насичення	Результати тестування безпосередньо використовуються як критерії прийнятності та коригування конфігурації

Таким чином, принципова відмінність перспективного design-time testing полягає не стільки у використанні конкретного інструменту тестування, скільки у функціональній ролі результатів тесту в процесі проєктування. Якщо у традиційних підходах метрики продуктивності використовуються переважно для виявлення проблем, контролю якості або адаптивного масштабування вже сформованої системи, то в ітеративному архітектурному проєктуванні вони можуть виступати формальними критеріями прийнятності поточної конфігурації.

Для цього для кожної вимірюваної характеристики задається допустиме граничне значення відповідно до вихідних вимог до системи. Наприклад, критерій прийнятності конфігурації можна визначити як

$$Q(\mathbf{c}^{(k)}) = \begin{cases} 1, & L_{p95}^{(k)} \leq L_{\max}, T^{(k)} \geq T_{\min}, U^{(k)} \leq U_{\max}, \\ 0, & \text{в іншому випадку,} \end{cases} \quad (8)$$

де $L_{\max}$ – максимально допустиме значення 95-го процентиля латентності; $T_{\min}$ – мінімально допустима пропускна здатність; $U_{\max}$ – максимально допустимий рівень використання критичного ресурсу. Значення $Q = 1$ означає виконання встановлених критеріїв продуктивності, тоді як $Q = 0$ вказує на необхідність коригування конфігурації.

За такого підходу процес перевірки архітектурного рішення набуває ітеративного характеру:

$$\mathbf{c}^{(k)} \xrightarrow{\text{тестування}} \mathbf{m}^{(k)} \xrightarrow{\text{оцінювання}} Q^{(k)} \xrightarrow{\text{коригування}} \mathbf{c}^{(k+1)}, \quad (9)$$

Якщо поточна конфігурація не відповідає встановленим вимогам, результати тестування використовуються для визначення напрямку її модифікації, після чого формується конфігурація наступної ітерації $\mathbf{c}^{(k+1)}$ та процедура повторюється. Завершення ітеративного циклу можливе після досягнення



заданих критеріїв прийнятності або встановленого обмеження на кількість ітерацій чи доступні ресурси проектування.

Важливим доповненням до оцінювання окремої конфігурації є визначення точки насичення, за якої подальше збільшення вхідного навантаження призводить до різкого зростання латентності, збільшення частоти помилок або вичерпання критичних ресурсів без пропорційного збільшення пропускної здатності. Виявлення цієї області на етапі проектування дає змогу оцінити запас продуктивності архітектури та перевірити її відповідність прогнозованому діапазону робочих навантажень.

Запропонована формалізація також забезпечує зв'язок із сурогатним прогнозуванням. Сурогатна модель може використовуватися для попереднього відбору конфігурацій, тоді як навантажувальне тестування — для верифікації їх фактичних характеристик. Результати таких випробувань, своєю чергою, можуть поповнювати набір даних для подальшого уточнення сурогатної моделі, утворюючи замкнений контур «прогнозування — тестування — уточнення».

Таким чином, перспективним напрямом розвитку автоматизованого проектування є перехід від використання тестування навантаження як самостійної контрольної процедури до ітеративної верифікації архітектурних конфігурацій, за якої виміряні характеристики продуктивності безпосередньо впливають на подальше проєктне рішення. Після визначення конфігурації, що відповідає встановленим вимогам до продуктивності, наступним етапом є оцінювання її відповідності вимогам до кібербезпеки, відмовостійкості та живучості.

Порівняльний аналіз підходів до відмовостійкості та захисту розподілених систем

Забезпечення кібербезпеки та відмовостійкості розподілених корпоративних систем охоплює механізми захисту, резервування, адаптації та збереження критичних функцій за умов відмов і деструктивних впливів. У контексті автоматизованого проектування принциповим є врахування цих характеристик безпосередньо під час вибору структури та параметрів архітектури, а не лише після її формування.

Порівняння показує, що окремі підходи забезпечують різні складові захищеності та стійкості системи, однак їх спільне врахування потребує багатокритеріального оцінювання архітектурних конфігурацій. Для конфігурації $\mathbf{c}$ вектор критеріїв можна подати як

$$\mathbf{J}(\mathbf{c}) = [J_{\text{sec}}, J_{\text{rel}}, J_{\text{surv}}, J_{\text{res}}, J_{\text{cost}}], \quad (10)$$

де J_{sec} — критерій кібербезпеки; J_{rel} — відмовостійкості; J_{surv} — живучості; J_{res} — ресурсовитратності; J_{cost} — вартості реалізації.

Таблиця 4

Порівняльні характеристики підходів до відмовостійкості та захисту розподілених систем

Підхід	Основний механізм	Основне призначення	Зв'язок із проектуванням
Периметровий захист	Фільтрація, сегментація, контроль доступу	Запобігання несанкціонованому доступу	Впливає на структуру мережі та точки контролю
Резервування та реплікація	Надлишковість вузлів і сервісів	Забезпечення доступності	Безпосередньо впливає на структуру системи
Zero Trust	Мінімізація довіри, мікросегментація	Контроль взаємодії компонентів	Формує архітектурні вимоги до доступу
Постквантовий захист	Постквантові криптографічні протоколи [17]	Криптографічна стійкість	Потребує врахування ресурсних витрат
Адаптивне виявлення загроз	ML-аналіз аномалій і атак	Виявлення та реагування	Переважає експлуатаційний рівень
Оптимізація ресурсів безпеки	Формалізовані критерії та оптимізація [7]	Раціональний розподіл ресурсів захисту	Створює основу для автоматизованого вибору конфігурації

Вибір архітектурної конфігурації після нормування критеріїв може бути представлений як багатокритеріальна задача

$$\mathbf{c}^* = \arg \max_{\mathbf{c} \in \mathcal{C}_A} (w_{\text{sec}} \tilde{J}_{\text{sec}} + w_{\text{rel}} \tilde{J}_{\text{rel}} + w_{\text{surv}} \tilde{J}_{\text{surv}} - w_{\text{res}} \tilde{J}_{\text{res}} - w_{\text{cost}} \tilde{J}_{\text{cost}}), \quad \sum_i w_i = 1, \quad (11)$$

де $\mathcal{C}_A$ – множина допустимих конфігурацій, що пройшли попередню верифікацію; $\tilde{J}$ – нормовані значення критеріїв; w_i – їх вагові коефіцієнти.

Такий підхід дає змогу розглядати кібербезпеку, відмовостійкість і живучість як характеристики самого проектного рішення та враховувати ресурсні обмеження під час вибору архітектури. Це створює основу для інтеграції відповідного етапу з ML-вибором архітектури та її верифікацією навантаженням у єдиному процесі автоматизованого проектування.

Взаємозв'язок трьох напрямів у контексті автоматизованого проектування

Проведений аналіз показує, що розглянуті напрями можуть бути інтегровані в послідовний процес автоматизованого проектування корпоративної обчислювальної системи (рис. 3). Його узагальнену структуру можна подати як

$$\mathbf{x} \xrightarrow{\text{ML прогнозування}} \hat{\mathbf{a}} \xrightarrow{\text{верифікація}} \mathbf{c} \xrightarrow{\text{оптимізація}} \mathbf{c}^*, \quad (12)$$

де $\mathbf{x}$ – вектор бізнес- і технічних вимог; $\hat{\mathbf{a}}$ – рекомендований архітектурний клас; $\mathbf{c}$ – сформована конфігурація; $\mathcal{C}_A$ – множина конфігурацій, що відповідають установленим вимогам до продуктивності; $\mathbf{c}^*$ – конфігурація, обрана з урахуванням вимог до кібербезпеки, відмовостійкості, живучості та ресурсних обмежень.

На першому етапі ML-класифікація та сурогатне моделювання забезпечують попередній вибір архітектурного рішення і прогнозування його характеристик. На другому етапі конфігурація ітеративно верифікується за результатами тестування навантаження. На третьому етапі серед допустимих конфігурацій здійснюється багатокритеріальний вибір з урахуванням вимог до кібербезпеки, відмовостійкості, живучості, ресурсовитратності та вартості.

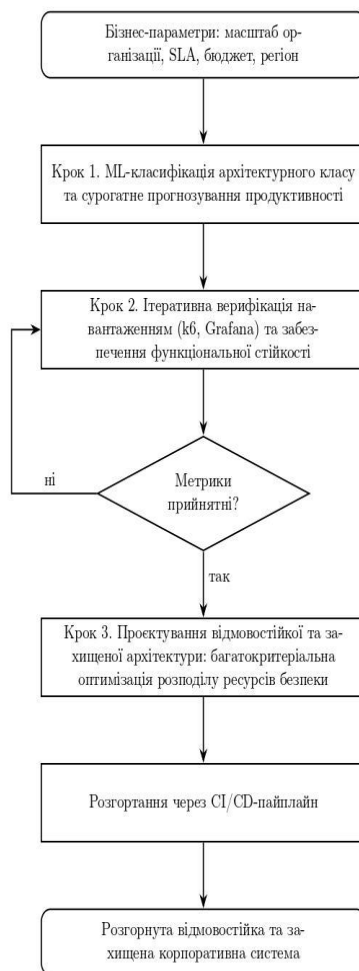


Рис. 3. Інтегрована схема процесу автоматизованого проектування



Таким чином, результати аналізу дають змогу розглядати ML-вибір і прогнозування, ітеративну верифікацію продуктивності та багатокритеріальне формування захищеної й відмовостійкої конфігурації як взаємопов'язані етапи єдиного процесу автоматизованого проєктування. У проаналізованих джерелах зазначені задачі розглядаються переважно окремо [4, 7, 10, 13], що визначає перспективність їх подальшої інтеграції в межах єдиної інформаційної технології.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У результаті проведеного дослідження систематизовано сучасні підходи до автоматизованого проєктування обчислювальних систем для корпоративних структур за трьома взаємопов'язаними напрямками: застосування машинного навчання для вибору архітектурних рішень і прогнозування їх характеристик; інтеграція тестування навантаження в ітеративний цикл проєктування; забезпечення кібербезпеки, відмовостійкості та живучості розподілених систем.

Встановлено, що наявні методи та інструментальні засоби забезпечують розв'язання окремих задач зазначених напрямів, однак їх наскрізна інтеграція в єдиний процес автоматизованого проєктування залишається недостатньо дослідженою. За результатами аналізу обґрунтовано перспективність послідовного поєднання ML-класифікації та пояснення архітектурних рішень, сурогатного прогнозування їх характеристик, ітеративної верифікації конфігурацій за результатами тестування навантаження та багатокритеріального вибору захищеної й відмовостійкої архітектури.

Запропонована концептуальна формалізація дає змогу розглядати ці складові як взаємопов'язані етапи єдиного процесу: від формалізованих бізнес- і технічних вимог до вибору архітектурного класу, оцінювання та верифікації конфігурації й подальшого врахування вимог до кібербезпеки, відмовостійкості, живучості та ресурсних обмежень.

Перспективи подальших досліджень пов'язані з розробленням і експериментальною верифікацією відповідних методів та інформаційних технологій, формуванням навчальних наборів даних архітектурних рішень, визначенням формальних критеріїв прийнятності конфігурацій і дослідженням ефективності інтегрованого процесу автоматизованого проєктування на реальних корпоративних обчислювальних системах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice* (4th ed.). Addison-Wesley Professional.
2. Lankhorst, M. (2017). *Enterprise Architecture at Work: Modelling, Communication and Analysis* (4th ed.). Springer. <https://doi.org/10.1007/978-3-662-53933-0>
3. Bezemer, C.-P., & Zaidman, A. (2010). Multi-tenant SaaS applications: Maintenance dream or nightmare? In *Proceedings of the 4th International Joint ERCIM/IWPSE Symposium on Software Evolution (IWPSE-EVOL 2010)* (pp. 88–92). Association for Computing Machinery. <https://doi.org/10.1145/1862372.1862393>
4. Gebreyesus, Y., Dalton, D., De Chiara, D., Chinnici, M., & Chinnici, A. (2024). AI for automating data center operations: Model explainability in the data centre context using Shapley Additive Explanations (SHAP). *Electronics*, 13(9), Article 1628. <https://doi.org/10.3390/electronics13091628>
5. Thota, M. R. (2024). Generative artificial intelligence as a catalyst for next-generation infrastructure design: Transforming the way enterprises architect, deploy, and scale digital platforms. *European Journal of Advances in Engineering and Technology*. <https://doi.org/10.5281/zenodo.18183400>
6. Mehla, A., Dahiya, N., & Singh, K. (2025). AI-driven cloud infrastructure provisioning: Leveraging ML, observability, and IaC for autonomous cloud operations. In *Proceedings of ETNCC 2025*. <https://doi.org/10.1109/ETNCC66224.2025.11299663>
7. Кушнір, Д., Регіда, П., Клейн, О., & Віжевський, П. (2026). Метод організації функціонування розподілених систем на основі автоматичного застосування критеріїв безпеки. *Вимірювальна та обчислювальна техніка в технологічних процесах*, (2), 374–386. <https://doi.org/10.31891/2219-9365-2026-86-44>
8. Vitui, A., & Chen, T.-H. P. (2021). MLASP: Machine learning assisted capacity planning: An industrial experience report. *Empirical Software Engineering*, 26(5). <https://doi.org/10.1007/s10664-021-09994-0>
9. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)* (pp. 4765–4774).



10. Гудь, О. О., & Кунанець, Н. Е. (2026). Порівняльний аналіз ефективності роботи алгоритмів XGBoost та LSTM у задачі прогнозування спроможності scrum-команд на різних горизонтах. *Науковий вісник НЛТУ України*, 36(1), 162–168. <https://doi.org/10.36930/40360118>
11. Олексів, Т. І. (2025). Застосування машинного навчання для автоматизованого оцінювання управління бізнес-процесами в IT-підприємствах. *Актуальні питання економічних наук*, (9). <https://doi.org/10.5281/zenodo.15164477>
12. Foo, K. C., Jiang, Z. M., Adams, B., Hassan, A. E., Zou, Y., & Flora, P. (2010). Mining performance regression testing repositories for automated performance analysis. In *Proceedings of the 10th International Conference on Quality Software (QSIC 2010)* (pp. 32–41). IEEE. <https://doi.org/10.1109/QSIC.2010.35>
13. Christadoss, J., & Mani, K. (2024). AI-based automated load testing and resource scaling in cloud environments using self-learning agents. *Journal of Artificial Intelligence and General Science*, 6(1), 604–618. <https://doi.org/10.60087/jaigs.v6i1.401>
14. Gaddam, R. R. (2025). AI-augmented dynamic performance engineering: A hybrid platform architecture. *Journal of Information Systems Engineering and Management*, 10(63s), 14–23. <https://doi.org/10.52783/jisem.v10i63s.13804>
15. Grant, O. H. (2025). Resource-aware cloud orchestration using ML-driven predictive scaling for high-demand IT applications. *Innovative Research Thoughts*, 11(4), 133–140. <https://doi.org/10.36676/irt.v11.i4.1716>
16. Додонов, О. Г., Кузнєцова, М. Г., & Горбачик, О. С. (2025). Моделювання і оцінювання функціональної стійкості інформаційних систем. *Реєстрація, зберігання і обробка даних*, 27(1), 76–88. <https://doi.org/10.35681/1560-9189.2025.27.1.335752>
17. Кучма, Ю., Поліновський, В., & Плахтій, М. (2026). Методи динамічної оптимізації постквантових цифрових підписів у протоколах автентифікації надщільних мереж 6G. *Кібербезпека: освіта, наука, техніка*, 1(33), 156–164. <https://doi.org/10.28925/2663-4023.2026.33.1125>
18. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939785>
19. Olson, R. S., & Moore, J. H. (2016). TPOT: A tree-based pipeline optimization tool for automating machine learning. In *Proceedings of the Workshop on Automatic Machine Learning (Proceedings of Machine Learning Research*, 64, pp. 66–74). PMLR

**Vadym Chytulian**

PhD Student,

State University of Information and Communication Technologies, Kyiv, Ukraine

ORCID:0009-0001-8846-9094

mag5187403@stud.duikt.edu.ua

Svitlana V. Popereshnyak

Candidate of Physics and Mathematics, Associate of Professor,

Associate Professor of the Department of Informatics and Software Engineering,

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

State University of Information and Communication Technologies, Kyiv, Ukraine

ORCID ID 0000-0002-0531-9809

spopereshnyak@gmail.com

INFORMATION TECHNOLOGIES FOR COMPUTER-AIDED DESIGN OF ENTERPRISE COMPUTING SYSTEMS: CURRENT STATE AND PROSPECTS

Abstract. The paper investigates the current state and prospects for the development of information technologies for computer-aided design of computing systems for enterprise structures. The relevance of the study is determined by the increasing complexity of enterprise information systems, the widespread adoption of distributed and cloud architectures, and the need to simultaneously address requirements for performance, scalability, cybersecurity, fault tolerance, and survivability. An analytical review and comparative analysis of current approaches are conducted in three interrelated areas: the application of machine learning to support architectural decision-making and predict system characteristics; the integration of load testing into the iterative design cycle; and the provision of cybersecurity and fault tolerance in distributed enterprise systems. The potential of ML classifiers, SHAP-based explainable artificial intelligence, AutoML, and surrogate modeling for the preliminary selection and evaluation of architectural configurations is considered. Load-testing approaches are analyzed, and the feasibility of using their results as feedback for iterative verification of architectural decisions is substantiated. Approaches to cybersecurity, fault tolerance, and survivability are systematized, and their incorporation as criteria for multi-criteria configuration selection is considered. The analysis demonstrates that existing methods and tools are primarily focused on solving individual tasks and remain insufficiently integrated into an end-to-end computer-aided design process. Based on the analysis, a conceptual formalization of such a process is proposed, sequentially combining the formalization of business and technical requirements, ML-based selection of an architectural class, prediction of configuration characteristics, iterative verification through load testing, and multi-criteria selection of a secure and fault-tolerant solution. Prospects for further research are identified, including the development and experimental validation of corresponding methods and information technologies for computer-aided design of enterprise computing systems.

Keywords: computer-aided design; enterprise computing systems; machine learning; architectural decisions; SHAP; AutoML; surrogate modeling; load testing; cybersecurity; fault tolerance; survivability

REFERENCES

1. Bass, L., Clements, P., & Kazman, R. (2021). *Software architecture in practice* (4th ed.). Addison-Wesley Professional.
2. Lankhorst, M. (2017). *Enterprise architecture at work: Modelling, communication and analysis* (4th ed.). Springer. <https://doi.org/10.1007/978-3-662-53933-0>
3. Bezemer, C.-P., & Zaidman, A. (2010). Multi-tenant SaaS applications: Maintenance dream or nightmare? In *Proceedings of the 4th International Joint ERCIM/IWPSE Symposium on Software Evolution (IWPSE-EVOL 2010)* (pp. 88–92). Association for Computing Machinery. <https://doi.org/10.1145/1862372.1862393>
4. Gebreyesus, Y., Dalton, D., De Chiara, D., Chinnici, M., & Chinnici, A. (2024). AI for automating data center operations: Model explainability in the data centre context using Shapley Additive Explanations (SHAP). *Electronics*, 13(9), Article 1628. <https://doi.org/10.3390/electronics13091628>



5. Thota, M. R. (2024). Generative artificial intelligence as a catalyst for next-generation infrastructure design: Transforming the way enterprises architect, deploy, and scale digital platforms. *European Journal of Advances in Engineering and Technology*. <https://doi.org/10.5281/zenodo.18183400>
6. Mehla, A., Dahiya, N., & Singh, K. (2025). AI-driven cloud infrastructure provisioning: Leveraging ML, observability, and IaC for autonomous cloud operations. In *Proceedings of ETNCC 2025*. <https://doi.org/10.1109/ETNCC66224.2025.11299663>
7. Kushnir, D., Rehida, P., Klein, O., & Vizhevskyi, P. (2026). Method of organizing the functioning of distributed systems based on the automatic application of security criteria [Metod orhanizatsii funktsionuvannya rozpodilennykh system na osnovi avtomatychnoho zastosuvannya kryteriiv bezpeky]. *Measuring and Computing Devices in Technological Processes*, (2), 374–386. <https://doi.org/10.31891/2219-9365-2026-86-44>
8. Vitui, A., & Chen, T.-H. P. (2021). MLASP: Machine learning assisted capacity planning: An industrial experience report. *Empirical Software Engineering*, 26(5). <https://doi.org/10.1007/s10664-021-09994-0>
9. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)* (pp. 4765–4774).
10. Hud, O. O., & Kunanets, N. E. (2026). Comparative analysis of the effectiveness of XGBoost and LSTM algorithms in predicting the capacity of Scrum teams at different horizons [Porivnialnyi analiz efektyvnosti roboty alhorytmiv XGBoost ta LSTM u zadachi prohnozuvannya spromozhnosti scrum-komand na riznykh horizontakh]. *Scientific Bulletin of UNFU*, 36(1), 162–168. <https://doi.org/10.36930/40360118>
11. Oleksiv, T. I. (2025). Application of machine learning for automated assessment of business process management in IT enterprises [Zastosuvannya mashynnoho navchannia dlia avtomatyzovanoho otsiniuvannya upravlinnia biznes-protsesamy v IT-pidpriemstvakh]. *Current Issues in Economic Sciences*, (9). <https://doi.org/10.5281/zenodo.15164477>
12. Foo, K. C., Jiang, Z. M., Adams, B., Hassan, A. E., Zou, Y., & Flora, P. (2010). Mining performance regression testing repositories for automated performance analysis. In *Proceedings of the 10th International Conference on Quality Software (QSIC 2010)* (pp. 32–41). IEEE. <https://doi.org/10.1109/QSIC.2010.35>
13. Christadoss, J., & Mani, K. (2024). AI-based automated load testing and resource scaling in cloud environments using self-learning agents. *Journal of Artificial Intelligence and General Science*, 6(1), 604–618. <https://doi.org/10.60087/jaigs.v6i1.401>
14. Gaddam, R. R. (2025). AI-augmented dynamic performance engineering: A hybrid platform architecture. *Journal of Information Systems Engineering and Management*, 10(63s), 14–23. <https://doi.org/10.52783/jisem.v10i63s.13804>
15. Grant, O. H. (2025). Resource-aware cloud orchestration using ML-driven predictive scaling for high-demand IT applications. *Innovative Research Thoughts*, 11(4), 133–140. <https://doi.org/10.36676/irt.v11.i4.1716>
16. Dodonov, O. H., Kuznietsova, M. H., & Horbachyk, O. S. (2025). Modeling and evaluation of functional stability of information systems [Modeliuvannya i otsiniuvannya funktsionalnoi stiiikosti informatsiinykh system]. *Data Recording, Storage & Processing*, 27(1), 76–88. <https://doi.org/10.35681/1560-9189.2025.27.1.335752>
17. Kuchma, Y., Polinovskiy, V., & Plakhtii, M. (2026). Methods of dynamic optimization of post-quantum digital signatures in authentication protocols for ultra-dense 6G networks [Metody dynamichnoi optymizatsii postkvantovykh tsyfrovyykh pidpysiv u protokolakh avtentyfikatsii nadshchilnykh merezh 6G]. *Cybersecurity: Education, Science, Technique*, 1(33), 156–164. <https://doi.org/10.28925/2663-4023.2026.33.1125>
18. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939785>
19. Olson, R. S., & Moore, J. H. (2016). TPOT: A tree-based pipeline optimization tool for automating machine learning. In *Proceedings of the Workshop on Automatic Machine Learning (Proceedings of Machine Learning Research, 64, pp. 66–74)*. PMLR.

Отримано редакцією журналу / Received: 06.03.26

Прорецензовано / Revised: 25.03.26

Схвалено до друку / Accepted: 24.09.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.