

Артем Васильович АНТОНЕНКО¹,

кандидат технічних наук, доцент, кафедра стандартизації та сертифікації
сільськогосподарської продукції, ORCID: <https://orcid.org/0000-0001-9397-1209>

Наталія Олександрівна ЛАЩЕВСЬКА²,

кандидат технічних наук, доцент, кафедра комп'ютерної інженерії,
ORCID: <https://orcid.org/0000-0003-2148-115X>

Олексій Григорович ТОНКИХ³,

кандидат економічних наук, доцент, кафедра менеджменту та адміністрування,
ORCID: <https://orcid.org/0000-0001-7823-4761>

Микола Анатолійович ЛЯШУК⁴,

викладач, кафедра інформаційної та комунікаційних технологій,
ORCID: <https://orcid.org/0009-0007-2910-4160>

Андрій Павлович ПРИХОДЬКО⁵,

магістр, ORCID: <https://orcid.org/0009-0001-2365-9903>

¹ Національний університет біоресурсів і природокористування України

² Державний університет інформаційно-комунікаційних технологій

³ Національний транспортний університет

⁴ Заклад вищої освіти «Міжнародний науково-технічний університет імені академіка Юрія Бугая»

⁵ Міжрегіональна академія управління персоналом

Прийняття: 20/05/2026

Рецензія: 29/05/2026

Публікація: 09/07/2026

DOI: <https://doi.org/10.53920/ITS-2026-1-5>

ШТУЧНІ НЕЙРОННІ МЕРЕЖІ ЯК ІНСТРУМЕНТ ІНТЕНСИФІКАЦІЇ ПРОДУКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК

004.415.53:004.891.3



This is an Open Access
article distributed
under the terms
of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)
CC-BY 4.0

© Антоненко А.В.,
Лашевська Н.О.,
Тонких О.Г.,
Ляшук М.А.,
Приходько А.П.,
2026

ISSN 2786-7226 (print)

ISSN 2786-7234 (online)

У статті досліджено та проаналізовано вплив нейронних мереж і систем глибинного навчання на підвищення продуктивності програмного забезпечення через механізми адаптивної та проактивної оптимізації. Актуальність теми зумовлена стрімким ускладненням архітектур (мікросервіси, контейнеризація, хмарні платформи), динамічністю навантажень і генерацією величезних масивів телеметричних даних, обробка яких класичними методами (статичний аналіз, ручний тюнінг, традиційний профайлінг) є неефективною. У роботі детально розглянуто архітектурний перехід сучасного ПЗ від реактивного копіювання збоями до інтелектуального управління обчислювальними процесами. Основними напрямками застосування нейромережевих моделей у контексті підвищення швидкодії визначено: прогнозування пікових навантажень на основі історичних логів (із точністю, що забезпечує автоматичне масштабування інфраструктури), інтелектуальний роз-

поділ ресурсів у хмарах, автоматичне налаштування параметрів (кешування, маршрутизації, запитів до БД) у режимі реального часу, а також нейрокомпіляцію та інтеграцію моделей у системи виконання для вибору оптимальних трансформацій коду та підвищення енергоефективності. У процесі дослідження проаналізовано досвід світових лідерів (Google DeepMind, Microsoft, IBM) та поточні практичні інструменти інтелектуального моніторингу (Azure Monitor AI Insights, Amazon DevOps Guru, Google Vizier, Meta Ax). Наведено результати опитування IT-фахівців, які ілюструють структуру впровадження ШІ-оптимізаторів: автоналаштування сервісів (41%), прогнозування продуктивності (36%), балансування навантаження мікросервісів (28%), прискорення алгоритмів (19%). Обґрунтовано стримувальні фактори впровадження технології, серед яких висока обчислювальна вартість навчання моделей, ризик помилкових рекомендацій і дефіцит профільних компетенцій у команд. У висновках сформульовано довгостроковий прогноз розвитку галузі до 2030 року, згідно з яким понад 50% корпоративних платформ міститимуть вбудовані ШІ-агенти адаптивної оптимізації, що підвищить продуктивність сервісів на 20–40% та знизить інфраструктурні витрати. Наголошено на необхідності розробки нових технічних стандартів, підвищення технічної грамотності інженерів (DevOps, Cloud) та створення автономних програмних систем, здатних до самонавчання без участі людини.

Ключові слова: нейронні мережі, штучний інтелект, оптимізація, продуктивність, машинне навчання, програмне забезпечення, хмарні обчислення, відмовистість, прогнозування, навантаження.

Artem ANTONENKO, Natalia LASHCHEVSKA, Oleksii TONKYKH, Mykola LIASHUK, Andrii PRYKHODKO

ARTIFICIAL NEURAL NETWORKS AS A TOOL FOR INTENSIFYING SOFTWARE PERFORMANCE

The article investigates and analyzes the impact of neural networks and deep learning systems on increasing software performance through adaptive and proactive optimization mechanisms. The relevance of the topic is due to the rapid complication of architectures (microservices, containerization, cloud platforms), the dynamics of loads and the generation of huge arrays of telemetric data, the processing of which by classical methods (static analysis, manual tuning, traditional profiling) is inefficient. The paper examines in detail the architectural transition of modern software from reactive coping with failures to intelligent management of computational processes. The main areas of application of neural network models in the context of increasing performance are: forecasting peak loads based on historical logs (with an accuracy that ensures automatic scaling of the infrastructure), intelligent resource allocation

in the clouds, automatic parameter tuning (caching, routing, database queries) in real time, as well as neurocompilation and integration of models into execution systems to select optimal code transformations and increase energy efficiency. The research analyzed the experience of world leaders (Google DeepMind, Microsoft, IBM) and current practical intelligent monitoring tools (Azure Monitor AI Insights, Amazon DevOps Guru, Google Vizier, Meta Ax). The results of a survey of IT professionals are presented, which illustrate the structure of the implementation of AI optimizers: auto-tuning of services (41%), performance forecasting (36%), load balancing of microservices (28%), algorithm acceleration (19%). The restraining factors for the implementation of the technology are substantiated, including the high computational cost of training models, the risk of erroneous recommendations, and the lack of specialized competencies in teams. The conclusions formulate a long-term forecast for the development of the industry until 2030, according to which more than 50% of corporate platforms will contain built-in AI agents for adaptive optimization, which will increase the performance of services by 20–40% and reduce infrastructure costs. The need to develop new technical standards, increase the technical literacy of engineers (DevOps, Cloud), and create autonomous software systems capable of self-learning without human intervention is emphasized.

Keywords: neural networks, artificial intelligence, optimization, performance, machine learning, software, cloud computing, fault tolerance, forecasting, workload.

Постановка проблеми. Нейронні мережі сьогодні відіграють дедалі важливішу роль у підвищенні продуктивності програмного забезпечення в різних сферах цифрових технологій. Стрімкий розвиток методів глибинного навчання, автоматичного аналізу даних та інтелектуального керування обчислювальними процесами відкриває нові можливості для оптимізації програмних систем. Завдяки здатності виявляти складні залежності, прогнозувати поведінку системи та адаптивно змінювати параметри роботи, нейромережеві моделі суттєво розширюють інструментарій розробників у вирішенні проблем продуктивності. Це призводить до того, що традиційні підходи — ручне налаштування, класичний профайлінг або статична оптимізація — все більше поступаються місцем автоматизованим інтелектуальним механізмам [1–3].

Застосування нейронних мереж у процесах оптимізації продуктивності почало докорінно змінювати підхід до створення програмних систем, формуючи нову парадигму адаптивних та самонавчальних рішень. Раніше розробники могли із значною часткою точності визначати оптимальні параметри роботи алгоритмів, проте зі збільшенням обсягів даних, ускладненням архітектур і зростанням вимог до швидкодії стало немож-

ливим вручну охопити всі фактори, що впливають на ефективність ПЗ. Нейронні мережі, здатні аналізувати великі масиви метрик, журналів роботи та параметрів навантаження, створюють більш реалістичні та точні моделі продуктивності, які відповідають сучасним масштабам й складності програмних систем [4-7].

Важливо враховувати й швидкість зміни робочих умов, зокрема в хмарних середовищах та мікросервісних архітектурах, де алгоритми машинного навчання не лише аналізують поведінку системи, але й оперативно реагують на перевантаження, прогнозують можливі збої і навіть самостійно оптимізують конфігурацію обчислювальних ресурсів. У таких умовах ефективність програмного забезпечення залежить від здатності моделі навчатися на реальних сценаріях використання та адаптуватися до нових вимог. Завдяки цьому нейронні мережі стають не просто інструментом аналізу, а активним компонентом управління продуктивністю.

Розвиток технологій штучного інтелекту, зокрема нейронних мереж і систем глибокого навчання, справив революційний вплив на підходи до оптимізації програмного забезпечення. Сучасні програмні системи працюють у складних обчислювальних середовищах, де навантаження є динамічним та непередбачуваним, а кількість параметрів, що впливають на ефективність, зростає щороку. Традиційні методи оптимізації — статичний аналіз, ручний тюнінг або профайлінг — вже не можуть забезпечити високу продуктивність у масштабованих, розподілених та багатокомпонентних системах [8, 9].

Однією з основних проблем стає те, що сучасне програмне забезпечення генерує величезні масиви телеметричних даних, які неможливо ефективно обробляти вручну. Нейронні мережі, здатні виявляти складні закономірності та взаємозв'язки між метриками продуктивності, відкривають нові можливості для автоматизованого аналізу ефективності й оптимізації параметрів системи. Відсутність таких інструментів ускладнює планування ресурсів, прогнозування навантаження, виявлення "вузьких місць" та запобігання збоєм [10-13].

Крім того, сучасні програмні архітектури — мікросервіси, хмари, контейнеризація — характеризуються високою варіативністю конфігурацій. В таких системах навіть незначні зміни навантаження можуть спричинити різке падіння продуктивності, якщо оптимізація не адаптується в режимі реального часу. Відсутність адаптивних механізмів робить програмне забезпечення вразливим до збоїв, перевантажень та неефективного використання ресурсів [14-17].

У цій ситуації зростає потреба у впровадженні нейромережевих моделей, здатних прогнозувати поведінку програмних компонентів, аналі-

зувати ефективність алгоритмів та автоматично оптимізувати параметри системи. Такі рішення стають ключовими для підвищення стабільності та продуктивності ПЗ, а також для забезпечення безперервної роботи критично важливих сервісів. [18, 19]

Аналіз останніх досліджень і публікацій. Аналіз сучасних досліджень показує, що нейронні мережі стають важливим інструментом у підвищенні продуктивності програмних систем. Праці провідних дослідницьких центрів — Google DeepMind, Microsoft Research, IBM AI Lab — демонструють високу ефективність моделей глибинного навчання для прогнозування навантаження, оптимізації ресурсів та інтелектуального керування обчислювальними процесами.

Так, в роботах DeepMind показано, що використання нейронних мереж для аналізу поведінки алгоритмів дозволяє зменшити час виконання програм на 20–60%, завдяки автоматичному вибору оптимальних гілок обчислень. Дослідники Google продемонстрували, що рекурентні та трансформерні моделі можуть з високою точністю прогнозувати пікові навантаження у хмарних середовищах, забезпечуючи більш стабільний розподіл ресурсів.

Численні публікації в IEEE Transactions та ACM Computing Surveys підтверджують, що нейромережеві підходи до оптимізації продуктивності значно випереджають класичні методи, оскільки здатні адаптуватися до змін середовища та працювати з великими обсягами телеметричних даних. Також відзначається зростаюча роль reinforcement learning, який дозволяє системам самостійно підбирати найкращі параметри під різні робочі сценарії.

Однак дослідження вказують і на певні виклики: висока обчислювальна вартість тренування моделей, складність у зборі якісних даних для навчання, обмеження інтерпретованості рішень нейромереж. Незважаючи на це, сучасна наукова спільнота активно рухається в напрямку розробки ефективних, надійних і масштабованих нейромережевих оптимізаторів.

Мета статті. Метою статті є дослідження та аналіз впливу нейронних мереж на підвищення продуктивності програмних систем за допомогою механізмів адаптивної оптимізації.

Виклад основного матеріалу дослідження. Сучасні підходи до використання нейронних мереж в оптимізації продуктивності програмного забезпечення охоплюють широкий спектр технологій та методів, серед яких ключову роль відіграють прогнозування навантаження, автоматичний тюнінг параметрів, інтелектуальний розподіл ресурсів та виявлення аномалій у роботі систем.

Одним із найбільш перспективних напрямів є моделі прогнозування продуктивності, що дозволяють оцінювати майбутні стани системи на основі історичних даних. Ці моделі виявляють закономірності, які неможливо помітити традиційними методами аналізу, та забезпечують можливість попередження перевантажень і запобігання збоєм. У хмарних середовищах такі системи дають змогу оптимізувати розподіл обчислювальних ресурсів, автоматизувати масштабування та зменшувати витрати на інфраструктуру.

Автоматичний тюнінг параметрів за допомогою нейронних мереж також набуває значної популярності. Моделі машинного навчання аналізують ефективність різних конфігурацій, вибирають оптимальні параметри та адаптують їх у режимі реального часу. Це стосується, зокрема, налаштування кешування, оптимізації запитів до баз даних, вибору структур даних або регулювання параметрів мікросервісів.

Окремого значення набуває застосування нейромереж у виявленні “вузьких місць” у програмних системах. Нейронні моделі здатні аналізувати телеметрію та виявляти аномалії у часі виконання, затримках, використанні пам’яті чи пропускній здатності. Це дозволяє розробникам швидко визначати проблемні елементи та прискорювати процес налагодження.

Важливим напрямом досліджень є також інтеграція нейронних мереж у компілятори та системи виконання. Нейромережеві компілятори здатні автоматично обирати оптимальні трансформації коду, зменшувати обчислювальні витрати та підвищувати енергоефективність програмних рішень (рис. 1). Нейронні мережі формують нову парадигму оптимізації продуктивності — від реактивних до проактивних та самонавчальних систем, що здатні забезпечувати високу ефективність навіть у складних та змінних умовах.

Швидке зростання обсягів даних і ускладнення програмних архітектур створюють нові виклики для розробників, оскільки традиційні методи оптимізації продуктивності вже не забезпечують потрібного рівня ефективності. У таких умовах нейронні мережі починають відігравати ключову роль, формуючи нову парадигму інтелектуального керування обчислювальними процесами. Програмні системи дедалі частіше автоматично коригують власні параметри на основі фактичних метрик роботи, створюючи своєрідні «адаптивні середовища», у яких алгоритми навчаються та вдосконалюються без участі людини. Такий підхід кардинально змінює уявлення про масштабованість і стабільність ПЗ та впливає на те, як компанії будують інфраструктуру і управляють ресурсами.

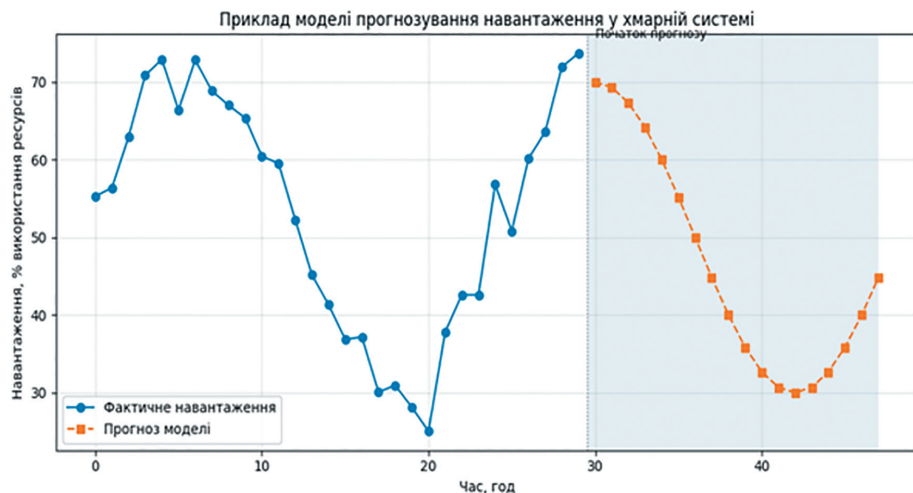


Рис. 1. Приклад моделі прогнозування навантаження у хмарній системі

Джерело: авторська розробка

Важливою особливістю впровадження нейромереж є те, що програмні системи поступово переходять від реактивного до проактивного керування продуктивністю. Моделі машинного навчання здатні передбачати зміни навантаження заздалегідь, що дозволяє уникати критичних затримок або падіння сервісів. Компанії, які інтегрували такі моделі у свої платформи, зазначають значне скорочення часу відгуку та підвищення стабільності роботи серверів, особливо під час пікових навантажень. Це підтверджують результати низки досліджень, у яких нейромережеві оптимізатори демонструють суттєву перевагу над традиційним профайлінгом та ручними налаштуваннями параметрів.

Опитування серед ІТ-фахівців виявило, що приблизно 30% команд розробників уже впроваджують нейромережеві інструменти для автоматичного аналізу логів, оптимізації запитів до баз даних, розподілу ресурсів у хмарних середовищах та тестування продуктивності. Серед причин, чому решта ще не застосовує ці технології, найчастіше називають недостатній досвід роботи з машинним навчанням, складність налаштування моделей та побоювання щодо некоректних рекомендацій алгоритму. Крім того, частина фахівців вважає, що нейронні мережі можуть генерувати помилкові висновки у складних сценаріях, де потрібен детальний інженерний аналіз.

Системи, які вже застосовують нейромережі в оптимізації, найчастіше використовують їх для таких задач: автоматичне масштабування сервісів (41%), прогнозування продуктивності (36%), оптимізація навантаження між мікросервісами (28%), прискорення роботи алгоритмів завдяки адаптивним конфігураціям (19%). Команди, які імплементували такі рішення, відзначають значне прискорення рутинних процесів розробки та експлуатації програмного забезпечення. Проте водночас фіксуються випадки, коли моделі потребують додаткового навчання, оскільки в певних умовах вони можуть давати не зовсім точні рекомендації або неправильно інтерпретувати системні метрики.

У контексті розвитку сучасних технологій питання впровадження нейронних мереж у системи оптимізації продуктивності стає дедалі важливішим. Вплив таких алгоритмів на ефективність роботи ПЗ вже підтверджений численними практичними експериментами, що відкриває шлях до створення повністю автономних програмних систем. Проте, разом із перевагами, виникає потреба у формуванні нових технічних стандартів, підходів до навчання моделей та механізмів контролю їхньої роботи. Розуміння цих аспектів є критично важливим для забезпечення стабільності, надійності та прогнозованості програмних платформ у майбутньому.

Використання нейронних мереж суттєво впливає на підходи до оптимізації роботи програмних систем — програмне забезпечення стає більш адаптивним, гнучким і здатним самостійно підлаштовуватися під потреби конкретних користувачів або робочих сценаріїв. Завдяки автоматичній обробці великих обсягів телеметричних даних і методам машинного навчання системи можуть швидко визначати ключові тенденції у навантаженні, виявляти повторювані патерни та прогнозувати зміни продуктивності. Це дозволяє формувати своєрідний «динамічний профіль» роботи програмного забезпечення, який постійно оновлюється залежно від умов експлуатації.

Такий підхід докорінно трансформує традиційні методи оптимізації, оскільки скорочує час між появою проблеми та її усуненням. Нейромережеві моделі аналізують дані у режимі реального часу, пропонуючи оптимальні конфігурації ресурсів або зміну параметрів роботи алгоритмів без необхідності ручного втручання. У результаті програмні системи отримують можливість працювати стабільніше, швидше реагувати на зміну навантаження та забезпечувати вищий рівень продуктивності для широкої аудиторії користувачів (рис. 2).

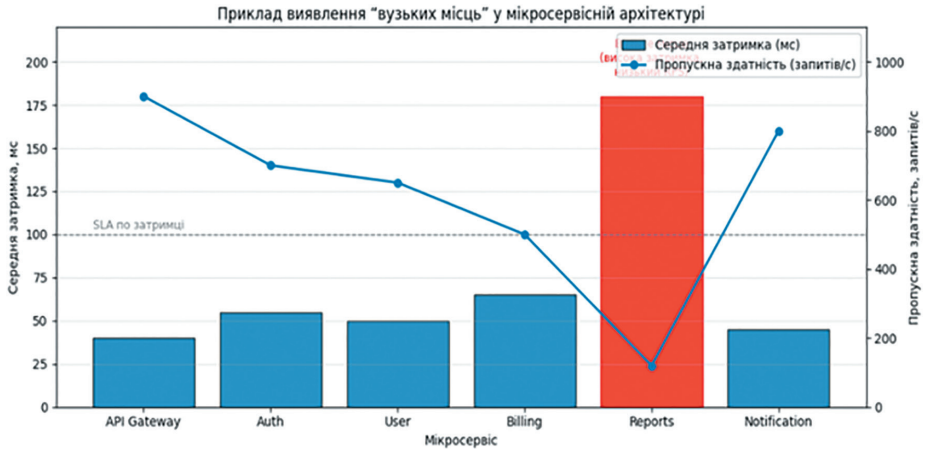


Рис. 2. Приклад виявлення «вузьких місць» у мікросервісній архітектурі

Джерело: авторська розробка

Технології застосування нейронних мереж для оптимізації продуктивності програмного забезпечення стрімко розвиваються у відповідь на зростаючу складність сучасних цифрових систем і потребу в обробці великих обсягів даних у реальному часі. Сучасні моделі машинного та глибинного навчання здатні аналізувати тисячі параметрів виконання програм, виявляючи закономірності в роботі компонентів, які впливають на швидкість. Завдяки цьому стає можливим відрізнити ефективні алгоритмічні стратегії від тих, що створюють надмірне навантаження або спричиняють затримки. Такі моделі враховують аномалії у використанні оперативної пам'яті, неефективні шаблони доступу до баз даних, нестабільність часу відгуку, а також приховані симптоми деградації продуктивності, які не завжди може виявити людина.

Інструменти моніторингу та інтелектуальної аналітики, наприклад, Azure Monitor AI Insights або Amazon DevOps Guru, дозволяють розробникам відстежувати поведінку програмних систем на основі даних, зібраних під час виконання. Такі сервіси аналізують мільйони метрик, автоматично розпізнаючи аномалії, перебої в роботі, потенційні «вузькі місця» та надаючи рекомендації щодо оптимізації. Подібні рішення зменшують навантаження на команди інженерів, даючи змогу спрямовувати свої зусилля на стратегічні завдання замість рутинних перевірок. Вони та-

кож формують безпечні середовища для моделювання навантажень і прогнозів, допомагаючи компаніям ухвалювати технологічні рішення більш обґрунтовано.

Автоматизація процесів оптимізації стала одним із ключових напрямів розвитку систем управління продуктивністю. Інструменти на основі ШІ можуть аналізувати великі масиви логів, знаходити повторювані патерни навантаження й автоматично визначати ділянки, що потребують оптимізації. Системи типу Google Vizier або Meta Ax скорочують час, необхідний для експериментів з параметрами, оскільки здатні самостійно підбирати оптимальні значення для конфігурації сервісів, баз даних, кешів або мікро-сервісних взаємодій. Це дозволяє значно підвищити точність і швидкість оптимізації, водночас зберігаючи можливість для остаточного людського контролю.

Алгоритми інтелектуальної модерації роботи програмних систем набувають популярності у великих компаніях, де в реальному часі потрібно відстежувати збої, "просідання" продуктивності та неправильний розподіл ресурсів. Нейронні мережі виявляють нерівномірне навантаження, помилки маршрутизації запитів, надмірну конкуренцію потоків і інші проблеми, які в традиційних системах залишалися б непоміченими. Разом із тим їх необхідно постійно вдосконалювати, оскільки поява нових архітектурних рішень та складних сценаріїв навантаження потребує адаптивних моделей, здатних до самонавчання.

Прогнози щодо ролі нейронних мереж у забезпеченні продуктивності програмного забезпечення до 2030 року демонструють їх стрімке зростання та інтеграцію у всі етапи життєвого циклу ПЗ. За даними міжнародних досліджень, до кінця десятиліття понад 50% корпоративних систем матимуть вбудовані моделі прогнозування й адаптивної оптимізації. Очікується значне збільшення ефективності обчислень, зменшення витрат на інфраструктуру та прискорення часу розробки за рахунок автоматизації рутинних задач. Продуктивність сервісів прогнозують підвищити на 20–40% завдяки впровадженню архітектур, які самостійно оцінюють ефективність власної роботи.

У сфері хмарних обчислень це означатиме поступовий перехід до динамічних систем керування ресурсами, які здатні автоматично масштабувати інфраструктуру, розподіляти навантаження між вузлами та адаптувати конфігурації залежно від реальних потреб користувачів. Очікується поширення архітектур, де ШІ-агенти самі визначають найкращий спосіб виконання задач, зменшуючи кількість ручних налаштувань і потребу в глибокому втручанні інженерів.

Не менш важливою є потреба у підвищенні рівня технічної грамотності спеціалістів, які працюватимуть із такими системами. Компетенції у сфері машинного навчання, аналізу даних, DevOps і cloud-інженерії стануть основою для успішної інтеграції розумних оптимізаційних систем у сучасне програмне забезпечення. Розвиток освітніх програм, курсів і систем сертифікації забезпечить підготовку кадрів, здатних ефективно працювати з адаптивними архітектурами.

Успішне використання нейронних мереж у сфері оптимізації продуктивності ПЗ вимагатиме комплексного підходу, що включає технологічні інновації, стандартизацію процесів, розвиток аналітики та співпрацю між інженерами різних профілів. Важливою є міжнародна комунікація та обмін досвідом, оскільки розвиток таких систем відбувається глобально і потребує узгоджених рішень для забезпечення стабільності, безпеки та прозорості алгоритмів.

Висновки. Нейронні мережі суттєво трансформують підходи до забезпечення продуктивності програмного забезпечення, відкриваючи нові можливості для автоматизації, адаптивного керування та глибокого аналізу системних процесів. Вони здатні значно підвищити ефективність роботи цифрових платформ, однак поряд із цим виникає потреба у чітких стандартах використання, відповідальності за прийняті моделі рішення та розвитку компетенцій фахівців. Найперспективнішим шляхом є багатосторонній підхід, що поєднує технічні рішення, інженерну культуру та системну співпрацю, що дозволить забезпечити стабільність і надійність програмних систем у майбутньому.

ЛІТЕРАТУРА

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. <https://www.deeplearningbook.org>.
2. Dean, J., & Barroso, L. A. (2013). The tail at scale. *Communications of the ACM*, 56(2), 74–80. <https://arxiv.org/abs/1802.04799>.
3. Антоненко А.В., Лемешко А.В., Петрик А.В. (2023) Нейроморфні системи як інструмент реалізації штучного інтелекту. Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки. Том 34 (73) № 3 2023, 175–183. <https://doi.org/10.32782/2663-5941/2023.3.1/28>.
4. Chen, T., Moreau, T., Zheng, L., et al. (2018). TVM: An automated end-to-end optimizing compiler for deep learning. *OSDI*, 1–16. <https://arxiv.org/abs/1802.04799>.
5. Amiri, F., Zhou, R., & Li, Z. (2021). Predicting cloud workload using LSTM neural networks. *IEEE Access*, 9, 125507–125517. <https://doi.org/10.1109/ACCESS.2021.3112342>.
6. Антоненко А. В., Лемешко А. В., Балвак А. А., & Новіченко Є. О. (2023). Актуальні засади створення алгоритмів обробки інформації для логістичних центрів. Тав-

рійський науковий вісник. Серія: Технічні науки, (1), 25-32. <https://doi.org/10.32851/tnv-tech.2023.1.3>.

7. Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. *arXiv preprint*, arXiv:1611.07411. <https://arxiv.org/abs/1611.07411>.

8. Chen, Y., Li, S., Dai, J., et al. (2020). NeuSight: Neural performance diagnosis in large-scale systems. *ACM Symposium on Cloud Computing*, 117–129. <https://doi.org/10.1145/3419111.3421293>.

9. Антоненко А., Пахомов М., Калита Т., Галета В. Використання штучного інтелекту в автоматизованих системах. Вісник Хмельницького національного університету, №4, 2023 (323). 11-20. <https://www.doi.org/10.31891/2307-5732-2023-323-4-11-20>.

10. Abadi, M., Barham, P., Chen, J., et al. (2016). TensorFlow: A system for large-scale machine learning. *OSDI*, 265–283. <https://www.tensorflow.org>.

11. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *NeurIPS*, 5998–6008. <https://arxiv.org/abs/1706.03762>.

12. Антоненко А. В., Бенедіко І. В., Вічкарук А. І., Лисенко К. В., & Сижко О. Ю. (2023). Класифікації моделей застосування машинного навчання у кібербезпеці. Таврійський науковий вісник. Серія: Технічні науки, (4), 11-22. <https://doi.org/10.32782/tnv-tech.2023.4.2>.

13. Hwang, J., Kim, M., & Choi, J. (2020). Dynamic resource scaling using deep reinforcement learning in cloud computing. *IEEE Transactions on Cloud Computing*. <https://doi.org/10.1109/TCC.2020.2971991>.

14. Li, Y., Wang, S., & Li, J. (2019). Neural-based bottleneck detection for microservices. *IEEE International Conference on Web Services*, 65–72. <https://doi.org/10.1109/ICWS.2019.00033>.

15. Антоненко А.В., Шрам М.М., Замковий О.Ю., Орлюк Х.В., Дутко П.М. Особливості побудови масиву даних на основі нейромережі у сфері інтернету речей для готельно-ресторанного бізнесу. Вісник Хмельницького національного університету, Том 1, №5, 2023 (325), 252-257. <http://journals.khnu.km.ua/vestnik/wp-content/uploads/2023/11/325-n5-1-252-257.pdf>.

16. Peng, B., Zhang, X., & Chen, H. (2021). ML-based performance prediction for distributed systems. *ACM Transactions on Architecture and Code Optimization*, 18(4). <https://doi.org/10.1145/3490422>.

17. Xu, H., Li, Z., & Wu, J. (2022). Optimizing microservice deployment with neural architecture search. *Journal of Systems and Software*, 183, 111091. <https://doi.org/10.1016/j.jss.2021.111091>.

18. Антоненко, А., Бурачинський А., Сольський Д., Твердохліб А., Мішкур Y., & Зіняр D. (2024). Аспекти застосування нейронних мереж для криптографії. *Measuring And Computing Devices In Technological Processes*, (4), 394–400. <https://doi.org/10.31891/2219-9365-2024-80-47>.

19. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1), 41–50. <https://doi.org/10.1109/MC.2003.1160055>.
20. Zheng, S., Kolb, S., & Liu, H. (2021). AI techniques for microservice bottleneck detection. *IEEE Transactions on Services Computing*. <https://doi.org/10.1109/TSC.2021.3109876>.
21. Антоненко, А., Мішкур, Ю., Твердохліб, А., Востріков, С., & Балвак, А. (2024). Нейромережі в мистецтві як іструмент графічного дизайну. *Herald of Khmelnytskyi National University. Technical Sciences*, 345(6(2), 95–101. <https://doi.org/10.31891/2307-5732-2024-345-6-14>.
22. Google Cloud. (2023). Machine learning for performance optimization. Google Technical Report. <https://cloud.google.com/architecture/ml-for-performance-optimization>.
23. Amazon Web Services. (2022). ML-based auto scaling for cloud applications. AWS Technical Documentation. <https://aws.amazon.com/autoscaling>.
24. Антоненко, А. В., Солобаєв, С. Г., Востріков, С. О., Ткаченко, О. В., Ходосов, А. О., & Остапенко, О. С. (2025). Використання нейронних мереж у прогнозуванні безпеки мережі. *Таврійський науковий вісник. Серія: Технічні науки*, (2), 3–10. <https://doi.org/10.32782/tnv-tech.2025.2.1>.
25. Антоненко, А., Сольський Д., Солобаєв С., Чечик С., & Черевик О. (2025). Використання бібліотеки scikit-learn у методах класифікації машинного навчання. *Measuring And Computing Devices In Technological Processes*, (1), 468–472. <https://doi.org/10.31891/2219-9365-2025-81-58>.
26. Твердохліб А.О., Антоненко А.В., Коротін Д.С. Ефективність функціонування комп'ютерних систем при використанні технології блокчейн і баз даних. *Таврійський науковий вісник. Серія: Технічні науки*, 2022, (6), 25–36 <https://doi.org/10.32851/tnv-tech.2022.6.4>.
27. Цвик О.С., Антоненко А.В. Аналіз і особливості програмного забезпечення для контролю трафіку. *Вісник Хмельницького національного університету. Серія: Технічні науки*, 2023, (1), 64–68. <https://doi.org/10.31891/2307-5732-2023-321-3-64-68>.
28. Новіченко Є.О., Антоненко А.В. Актуальні засади створення алгоритмів обробки інформації для логістичних центрів. *Таврійський науковий вісник. Серія: Технічні науки*, 2023 (1), 25–32. <https://doi.org/10.32851/tnv-tech.2023.1.3>.
29. Зайцев Є.О., Антоненко А.В. Smart засоби визначення аварійних станів у розподільних електричних мережах міст. *Таврійський науковий вісник. Серія: Технічні науки*, 2022, (5), 3–12. <https://doi.org/10.32851/tnv-tech.2022.5.1>.
30. Microsoft Research. (2020). Neural resource allocation for large-scale cloud systems. Technical Report. <https://www.microsoft.com/en-us/research>.
31. DeepMind. (2022). Neural algorithmic reasoning for system optimization. Research Publication. <https://deepmind.com/research>.

REFERENCES

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. <https://www.deeplearningbook.org>.
2. Dean, J., & Barroso, L. A. (2013). The tail at scale. *Communications of the ACM*, 56(2), 74–80. <https://arxiv.org/abs/1802.04799>.
3. Antonenko, A. V., Lemeshko, A. V., & Petryk, A. V. (2023). Neuromorphic systems as a tool for artificial intelligence implementation. *Vcheni Zapysky TNU Imeni V.I. Vernadskoho. Seriya: Tekhnichni Nauky*, 34(73), No. 3, 175–183. <https://doi.org/10.32782/2663-5941/2023.3.1/28>.
4. Chen, T., Moreau, T., Zheng, L., et al. (2018). TVM: An automated end-to-end optimizing compiler for deep learning. *OSDI*, 1–16. <https://arxiv.org/abs/1802.04799>.
5. Amiri, F., Zhou, R., & Li, Z. (2021). Predicting cloud workload using LSTM neural networks. *IEEE Access*, 9, 125507–125517. <https://doi.org/10.1109/ACCESS.2021.3112342>.
6. Antonenko, A. V., Lemeshko, A. V., Balvak, A. A., & Novichenko, Ye. O. (2023). Actual principles of creating information processing algorithms for logistics centers. *Taurian Scientific Herald. Series: Technical Sciences*, (1), 25–32. <https://doi.org/10.32851/tnv-tech.2023.1.3>.
7. Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. *arXiv preprint*, arXiv:1611.07411. <https://arxiv.org/abs/1611.07411>.
8. Chen, Y., Li, S., Dai, J., et al. (2020). NeuSight: Neural performance diagnosis in large-scale systems. *ACM Symposium on Cloud Computing*, 117–129. <https://doi.org/10.1145/3419111.3421293>.
9. Antonenko, A., Pakhomov, M., Kalyta, T., & Haleta, V. (2023). Use of artificial intelligence in automated systems. *Herald of Khmelnytskyi National University*, (4), 11–20. <https://www.doi.org/10.31891/2307-5732-2023-323-4-11-20>.
10. Abadi, M., Barham, P., Chen, J., et al. (2016). TensorFlow: A system for large-scale machine learning. *OSDI*, 265–283. <https://www.tensorflow.org>.
11. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *NeurIPS*, 5998–6008. <https://arxiv.org/abs/1706.03762>.
12. Antonenko, A. V., Benediko, I. V., Vichkaruk, A. I., Lysenko, K. V., & Syzhko, O. Yu. (2023). Classifications of models for applying machine learning in cybersecurity. *Taurian Scientific Herald. Series: Technical Sciences*, (4), 11–22. <https://doi.org/10.32782/tnv-tech.2023.4.2>.
13. Hwang, J., Kim, M., & Choi, J. (2020). Dynamic resource scaling using deep reinforcement learning in cloud computing. *IEEE Transactions on Cloud Computing*. <https://doi.org/10.1109/TCC.2020.2971991>.
14. Li, Y., Wang, S., & Li, J. (2019). Neural-based bottleneck detection for microservices. *IEEE International Conference on Web Services*, 65–72. <https://doi.org/10.1109/ICWS.2019.00033>.

15. Antonenko, A. V., Shram, M. M., Zamkovyi, O. Yu., Orliuk, Kh. V., & Dutko, P. M. (2023). Features of building a data array based on a neural network in the field of the internet of things for the hotel and restaurant business. *Herald of Khmelnytskyi National University*, 1(5), 252–257.
16. Peng, B., Zhang, X., & Chen, H. (2021). ML-based performance prediction for distributed systems. *ACM Transactions on Architecture and Code Optimization*, 18(4). <https://doi.org/10.1145/3490422>.
17. Xu, H., Li, Z., & Wu, J. (2022). Optimizing microservice deployment with neural architecture search. *Journal of Systems and Software*, 183, 111091. <https://doi.org/10.1016/j.jss.2021.111091>.
18. Antonenko, A., Burachynskiy, A., Solskyi, D., Tverdokhlib, A., Mishkur, Y., & Ziniar, D. (2024). Aspects of neural networks application for cryptography. *Measuring and Computing Devices in Technological Processes*, (4), 394–400. <https://doi.org/10.31891/2219-9365-2024-80-47>.
19. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computer*, 36(1), 41–50. <https://doi.org/10.1109/MC.2003.1160055>.
20. Zheng, S., Kolb, S., & Liu, H. (2021). AI techniques for microservice bottleneck detection. *IEEE Transactions on Services Computing*. <https://doi.org/10.1109/TSC.2021.3109876>.
21. Antonenko, A., Mishkur, Yu., Tverdokhlib, A., Vostrikov, S., & Balvak, A. (2024). Neural networks in art as a graphic design tool. *Herald of Khmelnytskyi National University. Technical Sciences*, 345(6(2)), 95–101. <https://doi.org/10.31891/2307-5732-2024-345-6-14>.
22. Google Cloud. (2023). Machine learning for performance optimization. Google Technical Report. <https://cloud.google.com/architecture/ml-for-performance-optimization>.
23. Amazon Web Services. (2022). ML-based auto scaling for cloud applications. AWS Technical Documentation. <https://aws.amazon.com/autoscaling>.
24. Antonenko, A. V., Solobaiev, S. H., Vostrikov, S. O., Tkachenko, O. V., Khodosov, A. O., & Ostapenko, O. S. (2025). Using neural networks in predicting network security. *Taurian Scientific Herald. Series: Technical Sciences*, (2), 3–10. <https://doi.org/10.32782/tnv-tech.2025.2.1>.
25. Antonenko, A., Solskyi, D., Solobaiev, S., Chechyk, S., & Cherevyk, O. (2025). Using the scikit-learn library in machine learning classification methods. *Measuring and Computing Devices in Technological Processes*, (1), 468–472. <https://doi.org/10.31891/2219-9365-2025-81-58>.
26. Tverdokhlib, A. O., Antonenko, A. V., & Korotin, D. S. (2022). Efficiency of computer systems functioning when using blockchain technology and databases. *Taurian Scientific Herald. Series: Technical Sciences*, (6), 25–36. <https://doi.org/10.32851/tnv-tech.2022.6.4>.

27. Tsvyk, O. S., & Antonenko, A. V. (2023). Analysis and features of software for traffic control. *Herald of Khmelnytskyi National University. Series: Technical Sciences*, (1), 64-68. <https://doi.org/10.31891/2307-5732-2023-321-3-64-68>.
28. Novichenko, Ye. O., & Antonenko, A. V. (2023). Actual principles of creating information processing algorithms for logistics centers. *Taurian Scientific Herald. Series: Technical Sciences*, (1), 25-32. <https://doi.org/10.32851/tnv-tech.2023.1.3>.
29. Zaitsev, Ye. O., & Antonenko, A. V. (2022). Smart tools for determining emergency conditions in urban distribution electrical networks. *Taurian Scientific Herald. Series: Technical Sciences*, (5), 3-12 <https://doi.org/10.32851/tnv-tech.2022.5.1>.
30. Microsoft Research. (2020). Neural resource allocation for large-scale cloud systems. Technical Report. <https://www.microsoft.com/en-us/research>.
31. DeepMind. (2022). Neural algorithmic reasoning for system optimization. Research Publication. <https://deepmind.com/research>.