УДК 004.8 + 004.056.5 DOI: 10.30748/soi.2025.183.01

М.І. Главчев, Д.М. Главчев, В.І. Панченко

Національний технічний університет "Харківський політехнічний інститут", Харків

БАЛАНСУВАННЯ НАВАНТАЖЕННЯ СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАПОБІГАННЯ DDOS-АТАКАМ

У статті розглянуто актуальну проблему захисту систем штучного інтелекту від розподілених атак на відмову в обслуговуванні (DDoS-атак) шляхом комплексного застосування сучасних архітектурних підходів та ефективних методів оптимізації продуктивності. З огляду на зростаючу складність таких систем, їхня вразливість до зовнішніх загроз вимагає проактивних рішень. Для підвищення стійкості та масштабованості пропонується використання мікросервісної архітектури, яка дозволяє ізолювати компоненти системи, а також впровадження сайдкар-патерну, контейнеризації та оркестрації у середовищі Kubernetes. Такий підхід забезпечує гнучкість у керуванні ресурсами та швидке відновлення після збоїв. Особливу увагу приділено балансуванню навантаження як ключовому засобу протидії перевантаженню, що дозволяє рівномірно розподіляти вхідний трафік і забезпечувати безперервну роботу інформаційних систем навіть за умов інтенсивних атак. Додаткові механізми захисту реалізуються за допомогою кешування для зниження навантаження на бази даних, централізованої авторизації через Keycloak для безпечного керування доступом, а також попередньої валідації запитів та ефективного розподілу обчислювальних ресурсів. Для об'єктивної оцінки ефективності запропонованих підходів було проведено експериментальне моделювання навантаження з використанням інструменту Gatling. Це дозволило не тільки виявити потенційні вузькі місця в архітектурі, але й підтвердити практичну результативність запропонованих рішень у реалістичних умовах. Отримані результати підтверджують актуальність і доцільність застосування комплексного підходу до побудови захищених, стійких і високопродуктивних систем штучного інтелекту.

Ключові слова: *балансування навантаження; захист інформації; розподіл ресурсів; сайдкар; штучний інтелект; Gatling; Keycloak; Kubernetes.*

Вступ

Постановка проблеми. Системи штучного інтелекту (ШІ) стають невід'ємною частиною сучасних інформаційних систем: від чат-ботів і рекомендаційних сервісів до платформ прогнозної аналітики та автоматизації бізнес-процесів. Зростання попиту на подібні рішення зумовлює високу залежність від їхньої продуктивності та доступності. Будь-які збої у роботі таких систем можуть призвести до значних фінансових втрат, втрати довіри користувачів та зупинки критично важливих сервісів.

Однією з найбільш поширених загроз для вебсервісів, у тому числі й для систем ШІ, залишаються DDoS-атаки (Distributed Denial of Service). Їхня мета – створення надмірного навантаження на інформаційну систему з подальшою втратою доступності для легальних користувачів. У випадку систем ШІ ризики зростають, оскільки обробка запитів до моделей зазвичай вимагає значних обчислювальних ресурсів, що робить їх особливо вразливими до перевантажень.

Аналіз останніх досліджень і публікацій. Сучасні архітектурні підходи дають можливість підвищити стійкість систем до подібних атак. Зокрема, мікросервісна архітектура та використання сайдкар-патерну забезпечують гнучкість, а розгортання сервісів у контейнерах з подальшим керуванням через Kubernetes надає можливість динамічного масштабування [1]. Ключову роль у протидії перевантаженням відіграє балансування навантаження, яке дозволяє рівномірно розподіляти запити між сервісами, зберігати продуктивність та запобігати відмовам. У роботі [2] запропоновано модель прогнозування споживання ресурсів (центральний процесор (CPU), пам'ять та інші) контейнеризованих мікросервісів, розгорнутих у Kubernetes, на основі емпіричних вимірювань і формальних ресурсних моделей. Додаткові засоби захисту [3] включають авторизацію через Keycloak [4], використання кешу для зменшення кількості повторних обчислень, а також правильний розподіл ресурсів між контейнерами. Ефективність впроваджених рішень може перевірятися за допомогою Gatling та інших open-source інструментів для тестування навантаження та моніторингу стану системи.

У роботах [1; 5; 7–11] описано сучасні підходи до захисту систем штучного інтелекту від DDoS-атак, що свідчить про активне використання технологій контейнеризації, сервісних сіток (service mesh), балансування навантаження та інструментів навантажувального тестування. Це дуже важлива та актуальна тема, адже багато спеціалістів та науковців намагаються знайти найкраще архітектурне рішення, що буде побудоване на мікросервісній архітектурі, та забезпечить стабільну роботу, та можливість витримувати високі навантаження в кількості запитів.

У документі [1] описано практичні рішення щодо забезпечення підключення та безпеки AI/ML-навантажень у Kubernetes. Автори підкреслюють важливість поєднання мережевої безпеки та масштабованості, що дозволяє одночасно підвищити продуктивність і знизити ризик перевантаження систем.

У роботі Chuang Y. [5] досліджено специфіку DDoS-атак у контейнеризованих середовищах. Наголошено на вразливості мікросервісної архітектури до перевантаження та запропоновано стратегії ізоляції контейнерів та адаптивного розподілу ресурсів для зменшення наслідків атак.

У роботі [6] наводиться огляд сучасних стратегій виявлення, запобігання та пом'якшення атак DDoS-атак у хмарних середовищах. Автори пропонують модель для ідентифікації та нейтралізації таких загроз, підкреслюючи важливість впровадження глибокого машинного навчання Deep Machine Learning (ML) та інших інноваційних технологій для підвищення стійкості хмарної інфраструктури.

Дослідження [7] акцентує на застосуванні технологій виконання програм у привілейованому контексті extended Berkeley Packet Filter (eBPF) і високопродуктивного методу обробки даних eXpress Data Path (XDP) для виявлення атак у хмарних системах. Автори доводять, що низькорівневий моніторинг мережевого трафіку забезпечує своєчасне реагування на загрози та є ефективним способом захисту середовищ ШІ.

Матеріал [8; 9] присвячений захисту від прикладних атак (рівень L7). Автори демонструють, як балансувальник HAProxy може використовуватися не лише для розподілу трафіку, а й для фільтрації шкідливих запитів, що особливо актуально для вебсерверів.

Статті [10; 11] пропонують огляд рішень для запобігання DDoS-атакам у Kubernetes, акцентуючи на багаторівневій архітектурі захисту, яка включає фільтрацію трафіку, автоматичне масштабування та інтелектуальний аналіз запитів.

Національний інститут стандартів і технологій США (NIST) надає детальний аналіз різних архітектур даних у service mesh для хмарних застосунків з оцінкою потенційних загроз та їхнього впливу на безпеку, а також рекомендації щодо вибору відповідних моделей проксі для застосунків з різними рівнями ризику [12].

У блозі [13] розглянуто роль service mesh у Kubernetes-середовищах. Показано, що ця технологія дозволяє централізовано контролювати маршрутизацію трафіку, посилюючи як безпеку, так і стабільність систем.

У матеріалі [14] детально проаналізовано завдання балансування трафіку в Kubernetes-кластерах. Зроблено акцент на підвищенні стійкості до перевантажень і забезпеченні безперервного доступу до сервісів під час DDoS-атак.

Інструменти для тестування продуктивності систем розглянуті у кількох джерелах. У матеріалі [15] висвітлено особливості навантажувального тестування мікросервісної архітектури. Матеріал [16] пропонує розподілене тестування з використанням Gatling у Kubernetes, що дозволяє моделювати DDoS-сценарії.

Довідковий ресурс Wikipedia [17] описує основні можливості Gatling, підкреслюючи його застосування у дослідженнях стійкості систем до високих навантажень.

Таким чином, огляд літератури демонструє, що комплексний підхід до запобігання DDoS-атакам на системи ШІ передбачає використання рішень на різних рівнях: від низькорівневого аналізу мережевого трафіку (eBPF, XDP) до архітектурних рішень (Kubernetes, service mesh) та практичного тестування навантажень (Gatling, Testkube). Це підтверджує актуальність поєднання балансування навантаження, контейнеризації та сучасних інструментів аналізу для формування стійких і масштабованих систем.

Мета статті – дослідження сучасних підходів до підвищення стійкості систем штучного інтелекту до DDoS-атак з особливим акцентом на балансування навантаження з використанням інструменту Gatling.

Питання побудови стійких до DDoS-атак ШІ-систем має практичну значущість, а балансування навантаження в поєднанні із сучасними методами контейнеризації, масштабованості та контролю доступу виступає основним механізмом забезпечення безперервної роботи інформаційних систем на базі штучного інтелекту.

Виклад основного матеріалу

1. Вразливість систем ШІ до DDoS-атак

Системи ШІ, на відміну від класичних веб-застосунків, характеризуються високою ресурсоємністю. Кожен запит до сервісу ШІ може включати складні обчислення, обробку великих обсягів даних або звернення до попередньо натренованих моделей. Це означає, що навіть відносно невелика кількість додаткових запитів може істотно вплинути на продуктивність системи. Саме ця особливість робить ШІ-сервіси вразливими до DDoS-атак [5]. Основні фактори уразливості систем ШІ такі [7]:

– висока обчислювальна складність запитів. Якщо традиційний веб-сервер може обслуговувати тисячі простих HTTP-запитів на секунду, то модель ШІ здатна обробити значно менше складних запитів за той самий час;

– залежність від апаратних ресурсів. Моделі машинного навчання часто працюють на GPU або спеціалізованих процесорах, доступність яких об-

межена. DDoS-атака може швидко вичерпати ці ресурси;

– мікросервісна архітектура як нова поверхня атаки. Хоча мікросервіси забезпечують масштабованість, атака на окремий сервіс (наприклад, API авторизації чи кешування) може паралізувати всю систему;

– висока вартість простою. Якщо ШІ використовується в критичних інформаційних системах (наприклад, у фінансових операціях, транспорті чи медичній діагностиці), навіть короткочасне перевантаження може мати серйозні наслідки.

До типів DDoS-атак, які небезпечні для систем ШІ [8], згідно системи OSI визначено такі:

– атаки на мережевому рівні (L3/L4): спрямовані на перевантаження каналів зв'язку, що робить веб-сервер або API недоступним;

– атаки на рівні застосунків (L7): численні складні запити до моделей ШІ, які імітують легітимну поведінку користувачів;

– гібридні атаки: поєднання мережевого перевантаження та атак на бізнес-логіку, що ускладнює їх виявлення;

Для визначення роботи системи використовується ряд показників:

– коефіцієнт відмов (rejection rate) R – визначається як

$$R = N_{\text{total}} / N_{\text{rej}} \cdot 100\ \%, \qquad (1)$$

де $N_{\text{total}}, N_{\text{rej}}$ – відхилені та загальні запити;

– час середньої затримки (latency) $\overline{T}$, який швидко зростає при атаці – розраховується за формулою

$$\overline{T} = \sum_{i=1}^{N} t_i, \qquad (2)$$

де N – кількість запитів; t_i – час відповіді на i-й запит;

– загальний час доступності системи A – описується виразом

$$A = MTBF / \left(MTBF + MTTR\right), \qquad (3)$$

де $MTBF$; $MTTR$ – середній час між відмовами та час на відновлення відповідно.

У результаті, навіть добре захищена інформаційна система без належних механізмів балансування навантаження, розподілу ресурсів та авторизації стає мішенню для зловмисників. Саме тому питання архітектурного захисту та впровадження гнучких механізмів масштабованості є критично важливим для систем штучного інтелекту.

2. Архітектурні підходи до підвищення стійкості

Ефективне запобігання DDoS-атакам на системи ШІ неможливе без правильного архітектурного підходу до побудови інформаційної системи. Для визначення пропускної здатності архітектури (throughput) використовується співвідношення $N_{\text{ok}} / \Delta t$, яке визначає, скільки успішних запитів N_{ok} обробляється за одиницю часу Δt при різних архітектурах. Використання сучасних моделей розробки, таких як мікросервісна архітектура, сайдкар-патерн та контейнеризація з оркестрацією в Kubernetes, дозволяє підвищити стійкість і забезпечити масштабованість сервісів.

Перехід від монолітних застосунків до мікросервісів дозволяє розділити систему на невеликі незалежні компоненти, кожен з яких виконує певну функцію [11]. Для систем ШІ це можуть бути окремі сервіси авторизації, управління даними, виклику моделей, кешування тощо. Такий підхід знижує ризик, що DDoS-атака на один сервіс паралізує всю систему. Крім того, мікросервіси можна масштабувати незалежно, що є важливим у випадках нерівномірного навантаження.

Використання сайдкарів дозволяє винести допоміжні функції (логування, моніторинг, проксі-запити, авторизацію) у додаткові сервіси, які розгортаються поруч з основними [8]. Це спрощує керування складними системами та підвищує їхню гнучкість. Наприклад, за допомогою сайдкара можна реалізувати локальний кеш для зменшення кількості звернень до основної моделі ШІ, тим самим знижуючи вплив DDoS-атаки.

Застосування контейнерів (Docker та ін.) дозволяє уніфікувати середовище виконання мікросервісів і швидко розгортати копії сервісів у разі підвищеного навантаження. Це забезпечує стійкість до атак шляхом горизонтального масштабування.

Kubernetes є ключовим інструментом для керування контейнерами у великих інформаційних системах [8; 9]. Його можливості включають: автоматичне масштабування сервісів залежно від навантаження; балансування навантаження між контейнерами; розподіл ресурсів (CPU, RAM, GPU) між подами (Pods) з урахуванням їхніх пріоритетів; самовідновлення сервісів у разі збою.

Таким чином, поєднання мікросервісної архітектури, сайдкар-патерну, контейнеризації та Kubernetes створює гнучку інфраструктуру, здатну витримувати навіть масштабні DDoS-атаки.

3. Балансування навантаження як засіб протидії DDoS

Одним із ключових механізмів забезпечення стійкості систем ШІ до DDoS-атак є балансування навантаження. Його основне завдання полягає в рівномірному розподілі вхідних запитів між сервісами, веб-серверами чи контейнерами, що дозволяє уникнути перевантаження окремих компонентів системи та підтримувати стабільну продуктивність [11]. Балансування виступає "посередником" між користувачем і системою та визначається співвідношенням

$$J = \frac{\left(\sum_{k=1}^{m} x_k\right)^2}{m\sum_{k=1}^{m} x_k^2}, 0 < J \le 1, \quad (4)$$

яке оцінює рівномірність розподілу навантаження x_k (або пропускну здатність) на k-му вузлі між m вузлами, наскільки воно рівномірно розкладає трафік (чим ближче до 1 – тим краще). У випадку DDoS-атаки, коли кількість запитів різко зростає, балансування не дозволяє одному серверу чи контейнеру стати "точкою відмови" та забезпечує рівномірний розподіл ресурсів, підвищену масштабованість системи та можливість динамічного додавання або відключення контейнерів [14].

Традиційні веб-сервери, такі як NGINX та HAProxy, широко застосовуються як балансування навантаження і можуть фільтрувати та маршрутизувати HTTP(S)-запити, відкидати підозрілі або надто часті запити, інтегруватися з системами кешування для зниження навантаження на сервіси ШІ.

У хмарних та контейнеризованих середовищах балансування навантаження виконується на рівні Kubernetes, що використовує такі механізми: Ingress-контролери (NGINX Ingress, Traefik, Istio) – керують трафіком і дозволяють застосовувати правила маршрутизації, Horizontal Pod Autoscaler (HPA) – автоматично масштабує кількість контейнерів залежно від навантаження, Pod Disruption Budgets та QoS-класи – гарантують, що критично важливі сервіси залишатимуться доступними навіть під час високого навантаження.

Балансування навантаження не лише підвищує доступність, але й допомагає нейтралізувати наслідки DDoS-атак, оскільки ускладнює перевантаження окремих вузлів, дає змогу швидко масштабувати інфраструктуру, дозволяє інтегрувати додаткові рівні захисту (авторизація, кешування, обмеження швидкості запитів). У комплексі з механізмами контролю доступу та моніторингом балансування є основним елементом архітектури захисту ШІ-систем від атак типу DDoS.

4. Оптимізація продуктивності та захист

Окрім балансування навантаження, важливим аспектом запобігання DDoS-атакам є підвищення продуктивності системи та застосування механізмів захисту, що знижують вплив шкідливих запитів ще до того, як вони досягають критично важливих ресурсів.

Кешування є ефективним засобом зменшення навантаження на обчислювальні моделі та бази даних за рахунок використання локального кешу (сайдкар), який може зберігати результати попередніх обчислень, що дозволяє відповідати на повторні запити без звернення до моделі ШІ, та розподіленого кешу (Redis, Memcached), який застосовується для масштабних систем, де необхідний доступ до однакових даних з кількох сервісів. Все це зменшує кількість операцій та підвищує стійкість до DDoS.

Застосування Keycloak як open-source рішення для управління авторизацією забезпечує контроль доступу до сервісів та забезпечує підтримку OAuth2, OpenID Connect, SAML [18], можливість обмеження швидкості запитів для окремих користувачів, а також інтеграцію з мікросервісами та Kubernetes для централізованого управління політиками доступу. Таким чином, більшість нелегітимних запитів може бути заблоковано ще до потрапляння в ядро інформаційної системи. Підвищення безпеки також забезпечується за рахунок застосування підходу Path-aware Security [19], який дозволяє відслідковувати та контролювати маршрути запитів між мікросервісами для підвищення захищеності. У роботах [20; 21] запропоновано систематичний огляд методів захисту мікросервісів, класифікуючи існуючі практики та проведено емпіричне дослідження практик безпеки.

Перед передачею запиту до основного сервісу ШІ виконується перевірка на синтаксис запиту (для відкидання некоректних або зловмисних запитів), частоту звернень та відповідність запитів правилам доступу. Цей підхід дозволяє відсіяти частину шкідливого трафіку на ранньому етапі.

У середовищах на основі Kubernetes можна задавати обмеження використання CPU, оперативної пам'яті (RAM) і графічного процесора (GPU) для кожного контейнера. Це унеможливлює ситуацію, коли одна служба, атакована DDoS, "з'їдає" всі ресурси й блокує інші сервіси системи.

У цілому, оптимізація продуктивності та багаторівневий захист дозволяють не лише підвищити стійкість системи до DDoS, але й гарантувати стабільність роботи ШІ-сервісів у критичних умовах.

5. Експериментальне тестування із застосуванням Gatling

Теоретичні методи протидії DDoS-атакам необхідно підкріплювати практичними дослідженнями. Для цього застосовуються інструменти моделювання навантаження [15; 16], які дозволяють перевірити стійкість системи до інтенсивного потоку запитів. Одним із найефективніших open-source рішень є Gatling (рис. 1).

Gatling – це платформа для стрес-тестування та навантажувального тестування веб-застосунків і мікросервісів, яка дозволяє створювати сценарії, що імітують велику кількість одночасних користувачів, аналізувати затримки, помилки та падіння продуктивності, оцінювати масштабованість і роботу балансувальників навантаження [17].

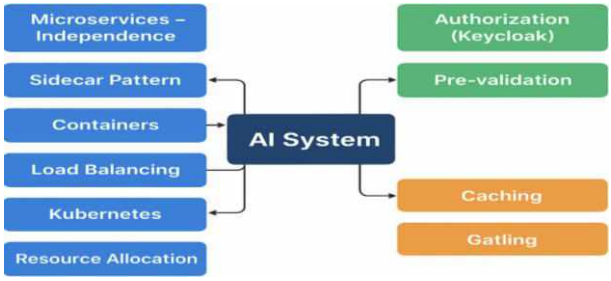

Рис. 1. Схема тестування балансування навантаження системи ШІ запобігання DDoS-атакам
Джерело: розроблено авторами.

Для проведення експерименту необхідно виконати такі дії:

1. Створення сценарію атаки. За допомогою Gatling формується набір HTTP-запитів, які імітують як легітимну поведінку користувачів, так і DDoS-подібне навантаження.

2. Запуск у середовищі Kubernetes. Система розгортається в контейнерах, де кожен мікросервіс має власні ресурси.

3. Застосування балансування та кешування. У конфігурації передбачаються механізми Ingress-контролера, HPA, Keycloak та Redis.

4. Збір метрик. Вимірюється продуктивність, час відгуку, стійкість до перевантажень, споживання ресурсів.

Експериментальні дослідження були проведені з метою оцінки ефективності різних архітектурних підходів до підвищення стійкості системи ШІ під час DDoS-атак. Проведення експерименту щодо порівняння продуктивності запропонованої системи балансування навантаження дозволило отримати результати, наведені в табл. 1, відповідно до обраних сценаріїв тестування.

Таблиця 1

Порівняння продуктивності системи під час DDoS-атаки

Сценарій тестування	Кількість одночасних запитів	Середній час відповіді, мс	Пропускна здатність (запитів/сек)	Відсоток відхилених запитів (%)	Використання CPU (%)	Використання RAM (%)
Без балансування	500	480	950	22	85	78
З балансуванням (1 вузол)	500	260	1800	8	72	70
З балансуванням (3 вузли, Kubernetes)	500	150	2800	2	68	65
З кешуванням та балансуванням	500	95	3400	1	60	58

Джерело: розроблено авторами.

Наведені в табл. 1 дані показують, як балансування навантаження та масштабування в Kubernetes знижують час відповіді, як росте пропускна здатність системи, падає відсоток відхилених запитів під час атаки та змінюється використання ресурсів (CPU, RAM). Було проведено аналіз отриманих результатів.

На рис. 2 наведено результати експериментального дослідження ефективності запропонованого підходу до балансування навантаження та оптимізації роботи системи під час DDoS-атаки.

Як видно з діаграм, відсутність балансування призводить до суттєвого збільшення середнього часу відповіді (480 мс) та високого рівня відхилених запитів (22 %). Впровадження балансування навіть на одному вузлі дозволяє знизити затримку майже удвічі (260 мс) та зменшити кількість відхилених запитів до 8 %.

Подальше масштабування до трьох вузлів у середовищі Kubernetes забезпечує ще кращі показники – середній час відповіді скорочується до 150 мс, а відсоток відхилених запитів знижується до 2 %. Найбільш ефективним виявився сценарій з додатковим використанням кешування, де час відповіді становив лише 95 мс, пропускна здатність досягла 3400 запитів/сек, а кількість відхилених запитів була мінімальною (1 %).

Рис. 2. Порівняння продуктивності системи у різних сценаріях
Джерело: розроблено авторами.

На рис. 3 наведено зміну пропускної здатності (кількості оброблених запитів за секунду (req/sec)) у різних сценаріях функціонування системи під час симульованої DDoS-атаки.

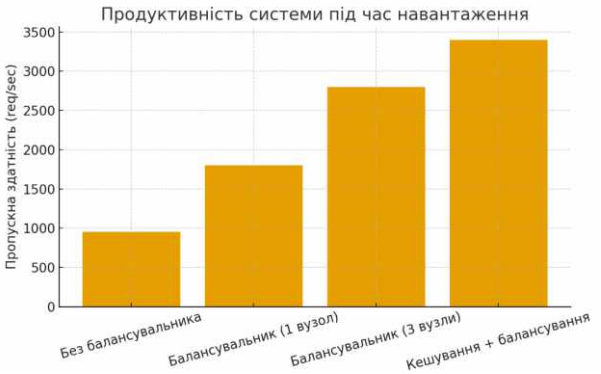

Рис. 3. Динаміка пропускної здатності системи
Джерело: розроблено авторами.

Без балансування система здатна обробляти лише близько 950 запитів/сек, що є обмеженням для високонавантажених інформаційних систем. Використання балансування на одному вузлі дозволяє майже вдвічі підвищити пропускну здатність (до 1800 запитів/сек). Подальше масштабування на три вузли в Kubernetes забезпечує ще кращі результати – 2800 запитів/сек. Найвищий показник досягнуто в сценарії з поєднанням балансування та кешування, де пропускна здатність сягнула 3400 запитів/сек.

На рис. 4 наведено ефективність різних архітектурних підходів у зменшенні частки відхилених запитів під час атаки.

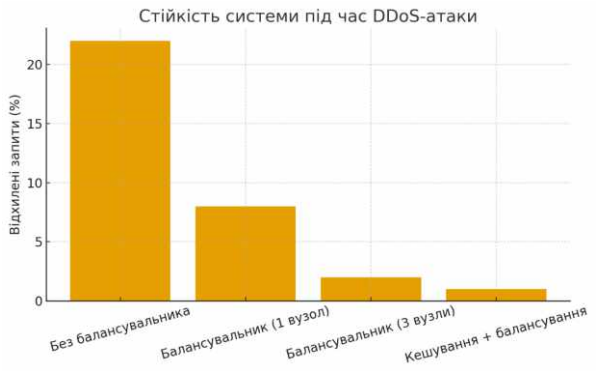

Рис. 4. Відсоток відхилених запитів
під час DDoS-атаки
Джерело: розроблено авторами.

У базовому сценарії без балансування майже чверть запитів (22 %) залишаються необробленими через перевантаження системи. Впровадження балансування на одному вузлі знижує цей показник до 8 %, що підтверджує ефективність навіть мінімального масштабування. При розгортанні трьох вузлів у Kubernetes кількість відхилених запитів зменшується до 2 %. Найкращий результат забезпечує комбінація кешування з балансуванням – відсоток відхилених запитів складає лише 1 %.

Отримані результати експериментів підтвердили, що поєднання балансування навантаження, масштабування контейнерів у Kubernetes та кешування є найбільш ефективним методом захисту систем штучного інтелекту від DDoS-атак. Застосування даних механізмів забезпечує оптимальний розподіл ресурсів, високу продуктивність, мінімізацію затримок та відмов у обслуговуванні, що робить такі підходи перспективними для впровадження в сучасних інформаційних системах.

Проведення експериментального тестування дозволяє отримати наступні очікувані результати: виявити вузькі місця в архітектурі, визначити ефективність балансування навантаження, перевірити, чи зберігається доступність сервісів під час високого навантаження, оцінити вплив авторизації (Keycloak) та кешування на зниження навантаження. До переваг проведеного дослідження з використанням Gatling слід віднести: автоматизацію (можливість інтеграції в CI/CD-процеси [22]), гнучкість (підтримку складних сценаріїв, включаючи стрес-тести), масштабованість (тестування великої кількості одночасних з'єднань) та візуалізацію (зручні графіки і звіти для аналізу результатів).

Таким чином, використання Gatling у комплексі з Kubernetes та інструментами балансування навантаження забезпечує практичну перевірку стійкості систем ШІ до DDoS-атак і дозволяє оптимізувати архітектуру ще на етапі проектування.

Висновки

Таким чином, проведено дослідження, у процесі якого розглянуто сучасні підходи до підвищення стійкості систем ШІ до DDoS-атак з особливим акцентом на балансування навантаження. Проведений аналіз показав, що поєднання архітектурних рішень, механізмів контролю доступу та оптимізації продуктивності є ключовим для забезпечення безперервної роботи інформаційних систем.

Основні висновки проведеного дослідження:

– мікросервісна архітектура та сайдкар-патерн забезпечують гнучкість системи, дозволяючи ізолювати критичні сервіси та зменшувати наслідки атак на окремі компоненти;

– контейнери та Kubernetes забезпечують динамічне масштабування, розподіл ресурсів і балансування навантаження, що є ефективним механізмом протидії DDoS;

– авторизація через Keycloak і попередня валідація запитів дозволяють відсіяти частину шкідливого трафіку ще на ранньому етапі, підвищуючи безпеку системи;

– кешування зменшує повторні обчислення мо-

делей ШІ, що знижує навантаження на обчислювальні ресурси та підвищує продуктивність;

– експериментальне тестування з Gatling дозволяє оцінити ефективність балансування та масштабованості, виявити вузькі місця та перевірити стійкість системи під час пікових навантажень.

Згідно проведеному дослідженню запропоновані наступні рекомендації для практичного застосування:

– використовувати комбінацію архітектурних та програмних рішень для захисту систем ШІ;

– забезпечити горизонтальне масштабування мікросервісів та автоматичне балансування навантаження;

– впроваджувати авторизацію та обмеження швидкості запитів на рівні API;

– використовувати кешування для повторних обчислень моделей ШІ;

– регулярно проводити навантажувальні тестування за допомогою Gatling та інших open-source інструментів.

У підсумку слід відзначити, що інтеграція балансування навантаження, масштабованої архітектури та комплексних заходів безпеки дозволяє значно підвищити стійкість систем штучного інтелекту до DDoS-атак та забезпечити стабільну роботу інформаційних систем у сучасних умовах високого навантаження.

Список літератури

1. Kubernetes connectivity and security for AI/ML workloads. *F5* : web site. URL: https://www.f5.com/pdf/solution-overview/kubernetes-connectivity-and-security-for-ai-ml-workloads.pdf (accessed 08.09.2025).

2. Turin G., Borgarelli A., Donetti S., Damiani F., Johnsen E. Br., Tarifa S. L. T. Predicting resource consumption of Kubernetes container systems using resource models. *Journal of Systems and Software*. 2023. Vol. 203. Art. 111750. https://doi.org/10.1016/j.jss.2023.111750.

3. Berardi D., Giallorenzo S., Mauro J., Melis A., Montesi F., Prandini M. Microservice security: a systematic literature review. *PeerJ Computer Science*. 2022. No. 8. Art. e779. https://doi.org/10.7717/peerj-cs.779.

4. Chatterjee A., Prinz A. Applying Spring Security Framework with KeyCloak-Based OAuth2 to Protect Microservice Architecture APIs: A Case Study. *Sensors*. 2022. No. 22(5). Art. 1703. https://doi.org/10.3390/s22051703.

5. Chuang Y.-T., Tu C.-H. Mitigating DDoS attacks in containerized environments: A comparative analysis of Docker and Kubernetes. *Journal of Parallel and Distributed Computing*. 2025. Vol. 204. Art. 105130. http://doi.org/10.1016/j.jpdc.2025.105130.

6. Ouhssini M., Afdel K., Akouhar M., Agherrabi E., Abarda A. Advancements in detecting, preventing, and mitigating DDoS attacks in cloud environments: A comprehensive systematic review of state-of-the-art approaches. *Egyptian Informatics Journal*. 2024. Vol. 27. Art. 100517. https://doi.org/10.1016/j.eij.2024.100517.

7. Sadiq A., Syed H. J., Ansari A. A., Ibrahim A. O., Alohaly M., Elsadig M. Detection of Denial of Service Attack in Cloud Based Kubernetes Using eBPF. *Applied Sciences.* 2023. Vol. 13. No. 8. Art. 4700. https://doi.org/10.3390/app13084700.

8. Lavoie C. Application layer DDoS attack protection with HAProxy. *HAPROXY* : web site. URL: https://www.haproxy.com/blog/application-layer-ddos-attack-protection-with-haproxy (accessed 08.09.2025).

9. Abdulkareem N. M., Zeebaree S. R. M. Optimization of Load Balancing Algorithms to Deal with DDoS Attacks Using Whale optimization Algorithm. *Journal of Duhok University*. 2022. No. 25(2). P. 65–85. https://doi.org/10.26682/sjuod.2022.25.2.7.

10. Bhayangkara D. S., Ashari W. M. Load Balancing Effect on DDoS Attacks Using Nginx: Pengaruh Load Balancing Pada Serangan DDoS Menggunakan Nginx. *The Indonesian Journal of Computer Science*. 2024. No. 13(4). https://doi.org/10.33022/ijcs.v13i4.4118.

11. DDoS attack on Kubernetes: What's the best solutions. *CloudAutoCraft* : web site. URL: https://cloudautocraft.com/ddos-attack-on-kubernetes-whats-the-best-solutions (accessed 08.09.2025).

12. Chandramouli R., Butcher Z., Callaghan J. Service Mesh Proxy Models for Cloud-Native Applications. NIST Special Publication (SP) NIST SP 800-233. Gaithersburg, MD : National Institute of Standards and Technology, 2024. 43 p. https://doi.org/10.6028/NIST.SP.800-233.

13. Using service mesh within your Kubernetes environment. *Kong* : web site. URL: https://konghq.com/blog/engineering/using-service-mesh-in-kubernetes-enviroment (accessed 08.09.2025).

14. Aggarwal S. Load balancing traffic to applications in Kubernetes cluster. *A10 Networks* : web site. URL: https://www.a10networks.com/blog/load-balancing-traffic-to-applications-in-kubernetes-cluster (accessed 08.09.2025).

15. Load testing and microservices architecture. *Gatling* : web site. URL: https://gatling.io/blog/load-testing-and-microservices-architecture (accessed 08.09.2025).

16. Distributed load testing with Gatling and Testkube on Kubernetes. *Testkube* : web site. URL: https://testkube.io/learn/distributed-load-testing-with-gatling-and-testkube-on-kubernetes (accessed 08.09.2025).

17. Gatling (software). *Wikipedia* : web site. URL: https://en.wikipedia.org/wiki/Gatling_%28software%29 (accessed 08.09.2025).

18. Christie M. A., Bhandar A., Nakandala S., Marru S., Abeysinghe E., Pamidighantam S., Pierce M. E. Managing authentication and authorization in distributed science gateway middleware. *Future Generation Computer Systems*. 2020. Vol. 111. P. 780–785. https://doi.org/10.1016/j.future.2019.07.018.

19. Meadows C., Hounsinou S., Wood T., Bloom G. Sidecar-based Path-aware Security for Microservices. *Proceedings of the 28th ACM Symposium on Access Control Models and Technologies (SACMAT '23).* Trento, Italy, 7–9 June 2023. P. 157–162. https://doi.org/10.1145/3589608.359474.

20. Hannousse A., Yahiouche S. Securing Microservices and Microservice Architectures: A Systematic Mapping Study. *Computer Science Review*. 2021. Vol. 41. Art. 100415. https://doi.org/10.1016/j.cosrev.2021.100415.

21. Nasab A. R., Shahin M., Raviz S.A.H., Liang P., Mashmool A., Lenarduzzi V. An Empirical Study of Security Practic-

es for Microservices Systems. *arXiv* : web site. 2022. https://doi.org/10.48550/arXiv.2112.14927.

22. Jani Y. Implementing continuous integration and continuous deployment (CI/CD) in modern software development. *International Journal of Science and Research*. 2023. Vol. 12. No. 6. P. 2984–2987. https://doi.org/10.21275/SR24716120535.

Надійшла до редколегії 12.09.2025
Схвалена до друку 23.12.2025

Відомості про авторів:

Главчев Максим Ігорович
кандидат економічних наук доцент
професор
Національного технічного університету
"Харківський політехнічний інститут",
Харків, Україна
https://orcid.org/0000-0001-9670-9118

Главчев Дмитро Максимович
доктор філософії
доцент
Національного технічного університету
"Харківський політехнічний інститут",
Харків, Україна
https://orcid.org/0000-0003-4248-4819

Панченко Володимир Іванович
старший викладач
Національного технічного університету
"Харківський політехнічний інститут",
Харків, Україна
https://orcid.org/0000-0003-3364-3398

Information about the authors:

Maksym Glavchev
PhD of Economic Sciences Associate Professor
Professor
of National Technical University
"Kharkiv Polytechnic Institute",
Kharkiv, Ukraine
https://orcid.org/0000-0001-9670-9118

Dmytro Hlavchev
PhD
Associate Professor
of National Technical University
"Kharkiv Polytechnic Institute",
Kharkiv, Ukraine
https://orcid.org/0000-0003-4248-4819

Volodymyr Panchenko
Senior Lecturer
of National Technical University
"Kharkiv Polytechnic Institute",
Kharkiv, Ukraine
https://orcid.org/0000-0003-3364-3398

LOAD BALANCING OF AN ARTIFICIAL INTELLIGENCE SYSTEM TO PREVENT DDOS ATTACKS

M. Glavchev, D. Hlavchev, V. Panchenko

The article addresses the urgent problem of protecting artificial intelligence (AI) systems from distributed denial-of-service (DDoS) attacks, which has become increasingly significant due to the rapid expansion of intelligent technologies across various domains of human activity. With the growing number of users and the increasing complexity of digital infrastructures, the vulnerability of such systems to external threats rises substantially, which makes the application of proactive and comprehensive solutions a necessity. The study emphasizes the integration of modern architectural approaches and performance optimization methods to simultaneously enhance the security, resilience, and scalability of AI systems. In particular, the use of microservices architecture is proposed, as it enables component isolation, minimizes the risks of cascading failures, and simplifies the update and maintenance of individual modules. Additionally, the importance of applying the sidecar pattern for traffic management, along with containerization and orchestration in Kubernetes, is highlighted. These techniques allow for automated scaling and rapid recovery of system functionality following attacks or technical failures. Special attention is given to load balancing as a critical mechanism against overload. The even distribution of incoming traffic across system components ensures stability even under intensive DDoS attacks. Other important protective measures include caching to significantly reduce database load, and centralized authorization through Keycloak, which strengthens access control and resource security. The proposed approach also incorporates request pre-validation and dynamic allocation of computing resources, contributing to higher processing efficiency. To objectively evaluate the effectiveness of the proposed solutions, load simulation experiments were carried out using the Gatling tool. This made it possible not only to identify potential bottlenecks in the architecture but also to confirm the practical applicability of the solutions under realistic operating conditions. The results confirm the relevance and effectiveness of a comprehensive approach that combines modern architectural design with security mechanisms and performance optimization techniques. It is concluded that such an approach enables the development of secure, scalable, and high-performance AI systems that can effectively withstand cyber threats while meeting the demands of modern users. This article addresses the challenge of protecting AI systems from DDoS attacks through modern architectural approaches and performance optimization techniques. It proposes the use of microservice architecture, the Sidecar pattern, containerization, and orchestration in a Kubernetes environment to enhance resilience and scalability. Special attention is devoted to load balancing as a key mechanism for preventing overload and ensuring uninterrupted operation of information systems. Additional protection mechanisms are implemented through caching, authorization (Keycloak), request pre-validation, and resource allocation. To evaluate the effectiveness of the proposed approaches, experimental load testing was conducted with Gatling, which made it possible to identify bottlenecks and confirm the efficiency of the solutions. The results obtained confirm the relevance of a comprehensive approach to building secure and high-performance artificial intelligence systems.

Keywords: *artificial intelligence; data protection; Gatling; Keycloak; Kubernetes; load balancing; resource allocation; sidecar.*