https://doi.org/10.31891/2307-5732-2026-361-70
УДК 004.9

СВИРИДОВ АРТЕМ
Харківський національний університет радіоелектроніки
https://orcid.org/0000-0002-9830-4103
e-mail: SvyrydovArtem@gmail.com
СЄВЄРІНОВ ОЛЕКСАНДР
Харківський національний університет радіоелектроніки
https://orcid.org/0000-0002-6327-6405
e-mail: oleksandr.sievierinov@nure.ua

МЕТОД ВИЯВЛЕННЯ АНОМАЛІЙ У KUBERNETES-КЛАСТЕРАХ З ВИКОРИСТАННЯМ LSTM AUTOENCODE

Об'єктом дослідження є інформаційна технологія виявлення аномальної та потенційно шкідливої активності у контейнеризованих середовищах на базі платформи Kubernetes. Предметом дослідження є методи аналізу багатовимірних телеметричних даних Kubernetes-кластера та їх використання у поєднанні з алгоритмами глибокого навчання для побудови інтелектуальних систем виявлення вторгнень. У роботі розглядається задача формування комплексного датасету, що поєднує мережеві характеристики трафіку, метрики контейнерів та показники стану Kubernetes-кластера. Особливу увагу приділено попередній обробці даних, зокрема імпутації пропущених значень, стандартизації простору ознак та формуванню часових послідовностей, необхідних для моделювання динамічної поведінки системи. З урахуванням дисбалансу між кількістю прикладів нормальної роботи та атакувальних сценаріїв, а також обмеженості множини відомих типів атак, обґрунтовано доцільність застосування підходу виявлення аномалій на основі навчання виключно на даних нормальної поведінки. Для розв'язання поставленої задачі запропоновано використання LSTM Autoencoder, який дозволяє моделювати часові залежності телеметричних даних та виявляти відхилення від вивченої нормальної поведінки шляхом аналізу помилки реконструкції. Запропоновано формальний підхід до визначення порогу аномальності на основі оптимізації F1-міри, що забезпечує збалансоване співвідношення між повнотою виявлення атак та кількістю хибних спрацювань. У ході експериментального дослідження проведено оцінювання ефективності запропонованого підходу з використанням стандартних метрик якості класифікації, зокрема precision, recall та F1-score. Отримані результати свідчать про здатність моделі виявляти до 95% атакувальних сценаріїв за умов відсутності попередньої інформації про їх типи, що підтверджує придатність підходу для виявлення zero-day атак. Практична значущість роботи полягає у можливості застосування запропонованого методу для побудови стійких та адаптивних систем безпеки Kubernetes-кластерів, орієнтованих на аналіз поведінкових аномалій у реальному часі.

Ключові слова: *Kubernetes, виявлення аномалій, система виявлення вторгнень, LSTM Autoencoder, глибоке навчання, телеметричні дані, контейнеризовані середовища, часові ряди, кібербезпека, zero-day атаки.*

SVYRYDOV ARTEM
Kharkiv National University of Radio Electronics
SIEVIERINOV OLEKSANDR
Kharkiv National University of Radio Electronics

A METHOD FOR ANOMALY DETECTION IN KUBERNETES CLUSTERS USING AN LSTM AUTOENCODER

The object of the study is an information technology for detecting anomalous and potentially malicious activity in containerized environments based on the Kubernetes platform. The subject of the research comprises methods for analyzing multidimensional telemetry data of a Kubernetes cluster and their application in combination with deep learning algorithms for building intelligent intrusion detection systems. The paper addresses the problem of constructing a comprehensive dataset that integrates network traffic characteristics, container-level metrics, and Kubernetes cluster state indicators. Special attention is given to data preprocessing, including missing value imputation, feature space standardization, and the formation of time sequences required to model the dynamic behavior of the system. Considering the imbalance between the number of normal operation samples and attack scenarios, as well as the limited set of known attack types, the study substantiates the feasibility of using an anomaly detection approach based on training exclusively on normal behavior data. To solve the stated problem, the use of an LSTM Autoencoder is proposed, which enables modeling temporal dependencies in telemetry data and detecting deviations from learned normal behavior through reconstruction error analysis. A formal approach to determining the anomaly threshold based on F1-score optimization is proposed, ensuring a balanced trade-off between attack detection recall and the number of false positives. During the experimental study, the effectiveness of the proposed approach is evaluated using standard classification performance metrics, including precision, recall, and F1-score. The obtained results demonstrate the model's ability to detect up to 95% of attack scenarios in the absence of prior information about their types, confirming the suitability of the approach for zero-day attack detection. The practical significance of the work lies in the possibility of applying the proposed method to build robust and adaptive security systems for Kubernetes clusters, focused on real-time behavioral anomaly analysis.

Keywords: *Kubernetes, anomaly detection, intrusion detection system, LSTM Autoencoder, deep learning, telemetry data, containerized environments, time series, cybersecurity, zero-day attacks.*

Стаття надійшла до редакції / Received 12.12.2025
Прийнята до друку / Accepted 11.01.2026
Опубліковано / Published 29.01.2026

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Сьогоднішній розвиток технологій віртуалізації таких, як контейнеризація та оркестрація контейнерів, значно спростив процес розгортання, масштабування та управління сучасними програмними застосунками.

Однією з найпопулярніших технологій віртуалізації є Kubernetes [1]. Кластери Kubernetes все частіше стають цілями різноманітних кібератак, що призводить до витоку даних, несанкціонованого використання обчислювальних ресурсів та інших видів загроз [2-3].

Основними факторами успішних кібератак є:

- помилки в конфігураційних файлах;
- наявність вразливостей в компонентах контейнерних платформ;
- недостатній рівень моніторингу та контролю внутрішніх процесів у кластерах.

Велику небезпеку становлять складні приховані атаки, пов'язані з впровадженням шкідливого програмного забезпечення у контейнерні образи та Pod. Також використання зловмисниками методів ухилення від виявлення та горизонтального переміщення всередині кластера значно підвищує ймовірність успішної атаки.

Таким чином актуальною науково-практичною проблемою є розробка та вдосконалення методів виявлення, запобігання та протидії кіберзагрозам у середовищі Kubernetes. Це передбачає розвиток нових підходів до моніторингу внутрішніх комунікацій, поведінкового аналізу контейнерів, а також інтеграції сучасних засобів інфокомунікаційної безпеки в контейнерні платформи.

Аналіз досліджень та публікацій

Проблематика безпеки контейнерних середовищ таких, як кластерів Kubernetes, на сьогодні є об'єктом активних наукових досліджень та практичних розробок. Зі зростанням попиту на використання контейнеризації в корпоративних і хмарних інфраструктурах зростає й інтерес науковців та фахівців з інформаційної безпеки до питання протидії кібератакам у середовищі Kubernetes.

У роботі [3] автори пропонують вдосконалений метод виявлення атак з використанням наївного байєсівського алгоритму, доповнений комплексною інженерією ознак та зменшенням розмірності за допомогою нейронних мереж. Проте такий підхід має ряд недоліків, які можуть вплинути на можливість практичного застосування такого підходу. По-перше, наївний байєсівський алгоритм припускає те, що ознаки мають умовну незалежність, що в реальних задачах кібербезпеки часто не виконується. По-друге використання нейроних мереж для зменшення розмірності призведе до підвищення обчислювальної складності методу та посилює вимоги до обсягу навчальних даних.

У роботі [4] запропоновано метод оцінювання функціональності кожного контейнера та порядок призначення динамічного ризику на основі потенційних загроз, таких як криптоджекінг, крадіжка даних та атаки типу «відмова в обслуговуванні» (DoS), які можуть використовувати ресурси контейнера. Бал виводиться з економічної цінності, пов'язаної з неправильним використанням ресурсів контейнера, на основі ринкових даних. Наприклад, контейнер, який може розміщувати майнер Monero, буде оцінюватися на основі його процесорної потужності та потенційного доходу від видобутої криптовалюти. Аналогічно, контейнери, що обробляють конфіденційні дані, такі як інформація про кредитні картки, оцінюються на основі ринкової вартості таких даних. Кількісно оцінюючи ризик таким чином, автори прагнуть запропонувати масштабований, керований даними підхід до аналізу kill-chain, який мінімізує залежність від експертної думки та підвищує точність пріоритезації загроз. Така зміна парадигми може дозволити проводити більш дієві та узагальнені оцінки ризиків, адаптовані до конкретного операційного контексту сучасних хмарних інфраструктур.

У публікації [5] проаналізовано сучасний стан безпеки середовищ оркестрації контейнерів Kubernetes, який на сьогодні є найбільш поширеним рішенням для розгортання cloud-native застосунків. Автори акцентують увагу на тому, що динамічний характер Kubernetes та складні сценарії атак роблять традиційні засоби безпеки, орієнтовані на статичні правила, малоефективними для забезпечення належного рівня захисту. Основний акцент у роботі зроблено на використанні контейнерних агентів безпеки з інтегрованими технологіями штучного інтелекту, машинного та глибинного навчання (AI/ML/DL), які забезпечують автономний та інтелектуальний механізм захисту. Авторами розглянуто архітектуру та принципи реалізації AI-орієнтованих систем безпеки в середовищі Kubernetes, а також проаналізовано їхню ефективність на практиці.

В статті [6] представлено особливості виявлення загроз в безпеці Kubernetes на основі штучного інтелекту, з особливим акцентом на його ролі у виявленні аномальної поведінки. Застосування алгоритмів штучного інтелекту дає змогу обробляти великі масиви телеметричних даних, які генеруються кластерами Kubernetes, у режимі реального часу, що забезпечує виявлення закономірностей і аномальних відхилень, які можуть свідчити про потенційні загрози безпеці або збої в роботі системи. Впровадження методів виявлення загроз на основі штучного інтелекту передбачає систематичний підхід, що охоплює збір даних, навчання моделей, інтеграцію з платформами оркестрації Kubernetes, механізми сповіщення та постійний моніторинг. Виявлення загроз на основі штучного інтелекту надає численні переваги, включаючи прогнозне виявлення загроз, підвищену точність і масштабованість, коротший час реагування та здатність адаптуватися до загроз, що розвиваються. Однак це також створює такі проблеми, як забезпечення якості даних, управління складністю моделі, зменшення хибнопозитивних результатів, задоволення потреб у ресурсах та підтримка стандартів безпеки та конфіденційності.

Проведений аналіз засвідчує, що стрімке зростання складності Kubernetes-інфраструктур вимагає переходу від традиційних rule-based підходів до інтелектуальних систем захисту, здатних працювати в умовах високої динаміки та великого обсягу телеметричних даних. Інтеграція алгоритмів штучного інтелекту, машинного та глибинного навчання у механізми моніторингу та виявлення загроз дозволяє значно підвищити точність і швидкість реагування, а також суттєво зменшити кількість хибнопозитивних спрацювань.

Методи поведінкового аналізу, машинної класифікації та аналізу мережевої взаємодії — зокрема на

основі VAE, CNN та GNN – демонструють високу ефективність у виявленні як відомих, так і нульових (zero-day) загроз. Водночас застосування інструментів, таких як eBPF і KubeArmor, забезпечує глибшу інтеграцію політик безпеки та контроль на рівні ядра, що підсилює можливості автономного захисту контейнерів.

Формулювання цілей статті

Мета дослідження полягає у розробленні та обґрунтуванні підходу до виявлення вторгнень у Kubernetes-кластерах на основі застосування алгоритму AdaBoost. Об'єктом дослідження є процеси забезпечення безпеки та моніторингу подій у середовищі Kubernetes, тоді як предметом дослідження є методи обробки та аналізу телеметричних і мережевих даних із використанням алгоритмів машинного навчання для виявлення аномалій та потенційних атак. Досягнення поставленої мети передбачає створення пайплайна збору, нормалізації та класифікації даних, навчання моделі AdaBoost на основі ознак, що характеризують поведінку контейнерів і ресурсів Kubernetes, а також оцінювання точності, повноти та адаптивності моделі для застосування в реальному середовищі. Запропонований підхід спрямований на підвищення ефективності виявлення вторгнень та зменшення кількості хибнопозитивних спрацювань у динамічних хмарних інфраструктурах.

Виклад основного матеріалу

Kubernetes – це система оркестрації контейнерів, яка автоматизує розгортання, масштабування та керування контейнеризованими застосунками. Вона забезпечує стабільну роботу сервісів, розподіляючи навантаження між вузлами та автоматично відновлюючи збої. Kubernetes має багаторівневу структуру, що складається з Control Plane та Worker Nodes (робочі вузли) і зображена на рисунку 1.

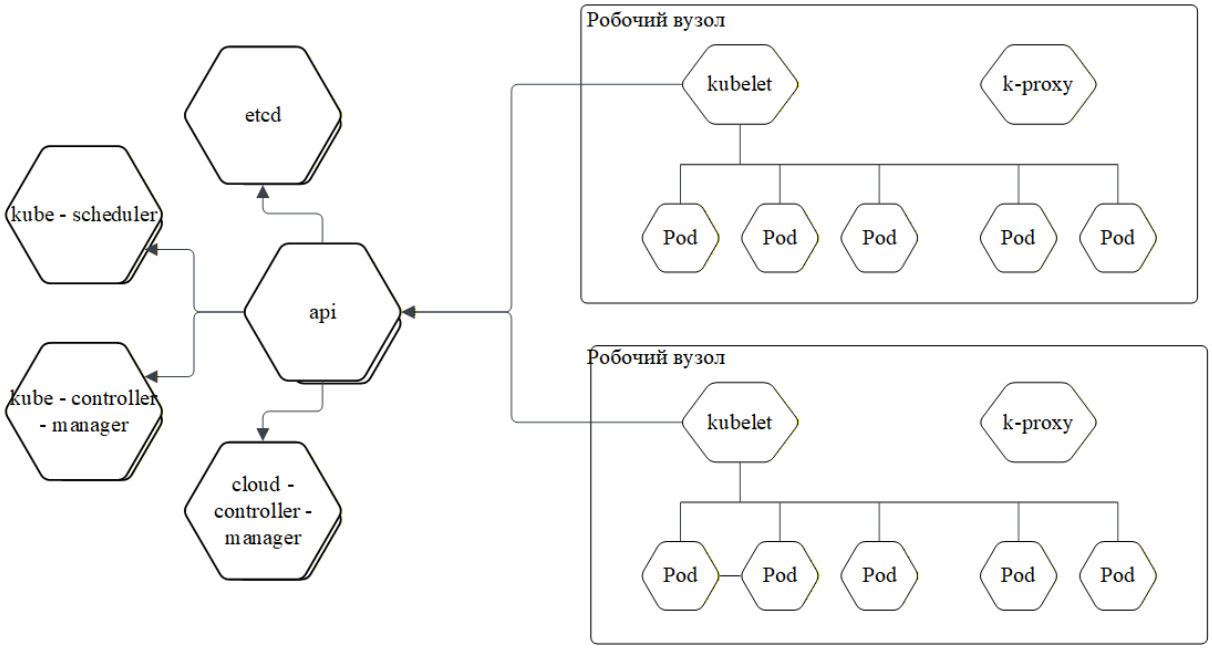

Рис. 1. Структура Kubernetes

Control Plane відповідає за управління кластером: планування подів, контроль стану, масштабування та підтримання узгодженого функціонального стану всієї системи.

У структурі Kubernetes важливу роль відіграють робочі вузли (Worker Nodes), які забезпечують виконання обчислювального навантаження. Саме на робочих вузлах розгортаються та функціонують поди, у яких запускаються контейнери застосунків. Кожен вузол містить необхідні системні компоненти — такі як kubelet, kube-proxy та контейнерний рантайм, що забезпечують керування життєвим циклом контейнерів, взаємодію з Control Plane та маршрутизацію мережевого трафіку. Таким чином, робочі вузли формують розподілене обчислювальне середовище, яке масштабовано обробляє запити та виконує сервіси користувача. Kubernetes може відновлювати збої завдяки вбудованим механізмам контролю стану подів та вузлів. Якщо контейнер або Pod виходить із ладу, Kubernetes автоматично перезапускає його, створює новий екземпляр або переносить навантаження на інший здоровий вузол. Така самовідновлювана архітектура підвищує надійність роботи застосунків, забезпечує стійкість до відмов та зменшує час простою системи.

Kubernetes Pod виступає базовою одиницею розгортання та виконання застосунків. Pod функціонує в межах робочого вузла (Worker Node) і спільно з іншими Pod-ами використовує мережевий простір імен вузла для комунікацій.

Спостереження за мережевим простором імен конкретного робочого вузла дає змогу отримати повну картину мережевого трафіку, що циркулює між Pod-ами на цьому вузлі. Це робить можливим виявлення нетипових або потенційно шкідливих мережевих взаємодій у кластері.

У випадку, якщо зловмисник отримує доступ до кластера, він змушений здійснювати внутрішнє переміщення, ідентифікувати цінні інформаційні ресурси та ініціювати їх переміщення за межі інфраструктури. Усі ці дії неминуче пов'язані з використанням внутрішньої мережі кластера, тому моніторинг трафіку на міжкомпонентному рівні є ключовим механізмом для своєчасного виявлення аномалій і потенційних порушень

безпеки.

Усередині мережевого простору імен кожного робочого вузла Kubernetes існує спеціальний віртуальний мережевий міст (virtual bridge), який виступає центральним елементом маршрутизації трафіку між Pod-ами та зовнішньою мережею. Цей міст забезпечує прозору комунікацію між усіма Pod-ами, розгорнутими на вузлі, і виконує функцію агрегатора мережевих потоків. Віртуальний міст підключений до всіх віртуальних Ethernet-інтерфейсів контейнерів, що входять до складу подів, і відповідає за передачу пакетів як всередині вузла, так і до інших вузлів кластера через оверлейні або фізичні мережеві інтерфейси.

Як показано на рисунку 2, віртуальний мережевий міст має прямий доступ до всіх підключених до нього інтерфейсів подів, що дозволяє контролювати весь трафік, що надходить і виходить із подів.

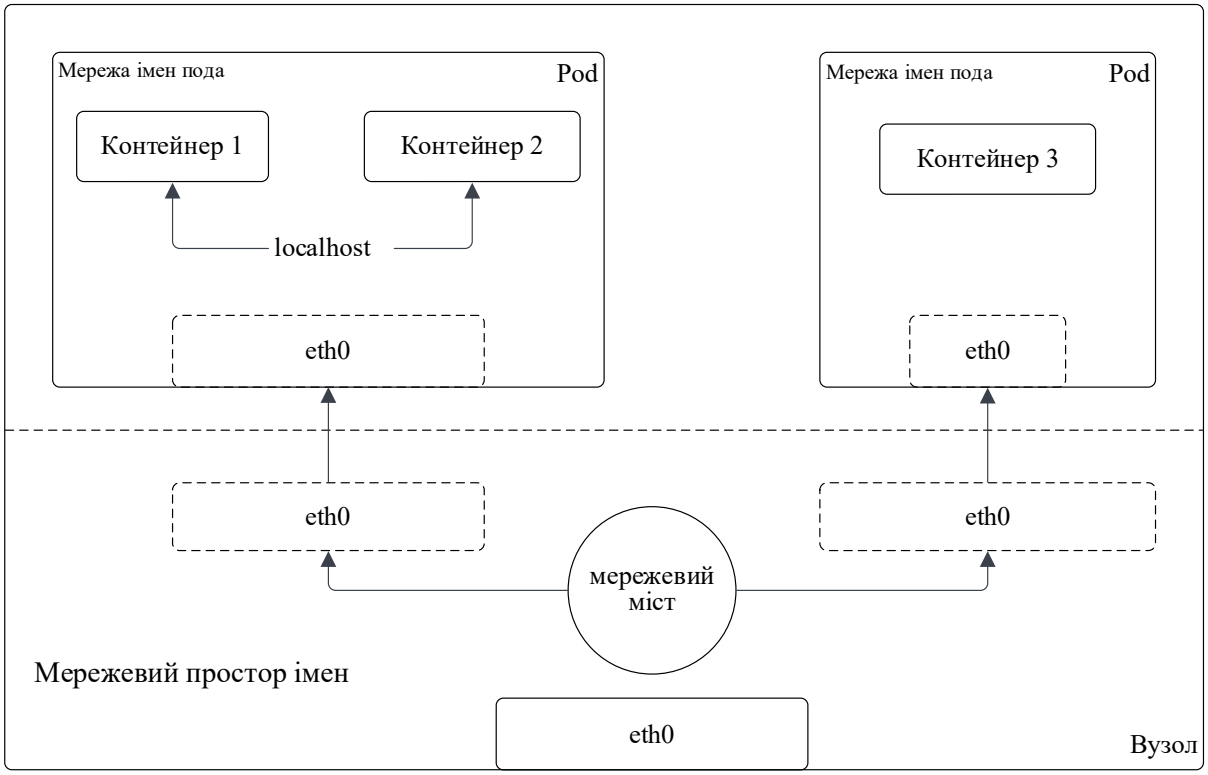

Рис. 2. Мережі між Pod-ами у вузлі

Прослуховування мережевих інтерфейсів на рівні віртуального мосту дозволяє здійснювати всебічний моніторинг всього трафіку, що проходить через робочий вузол, включаючи як внутрішньокластерні комунікації між Pod-ами, так і зовнішні з'єднання. Захоплений трафік може бути переданий до спеціалізованого Pod-а, на якому реалізовано алгоритм виявлення вторгнень, що здійснює детальний аналіз мережевих потоків. Такий підхід дозволяє не лише виявляти аномальні або підозрілі патерни в пакеті даних, але й проводити глибинний аналіз взаємодії компонентів кластера, визначати потенційні загрози та своєчасно реагувати на них. Впровадження подібних механізмів підвищує прозорість внутрішньої мережі Kubernetes, забезпечує контроль над безпекою на рівні вузлів та сприяє підвищенню загальної надійності та стійкості кластерної інфраструктури

У межах кластера Kubernetes мережевий трафік між Pod-ами та зовнішніми сервісами передається за допомогою набору стандартних мережевих протоколів, які забезпечують ефективну комунікацію, маршрутизацію та передачу даних.

Для дослідження було обрано набір даних [7] містить два набори даних для виявлення вторгнень, зібрані в Kubernetes-кластері. Дані були отримані шляхом симуляції нормальної та шкідливої активності на двох різних вебзастосунках, розміщених у кластері. Кожен набір даних містить:

– мережеві дані, файли перехоплення пакетів (PCAP) та витягнуті TCP-потоки;

– метрики контейнерів, метрики cAdvisor, такі як використання CPU контейнера, кеш-пам'ять, відкриті мережеві сокети, кількість процесів і потоків тощо;

– метрики кластера Kubernetes, метрики KubeStateMetrics та інші показники, зокрема кількість доступних Pod для мікросервісу та квоти ресурсів.

Ці дані є частиною статті [8].

У процесі навчання моделі виявлення аномалій використовується багатовимірний простір ознак, що описує стан мережевої взаємодії, контейнерів та Kubernetes-кластера. Формально повний простір ознак можна подати у вигляді об'єднання трьох множин:

$$X=F_{net} \cup F_{ctr} \cup F_{k8s}, \quad (1)$$

де F_{net} – множина мережевих ознак,

F_{ctr} – множина метрик контейнера,

F_{k8s} – множина системних показників Kubernetes-кластера.

Множену мережних показників можна описати наступним чином:

$$F_{net} = \{f_1^{net}, f_2^{net}, \dots, f_m^{net}\}, \quad (2)$$

де кожен елемент відповідає певній характеристиці TCP-трафіку або статистичному параметру мережевих потоків, отриманих із PCAP-даних. До цієї множини входять такі ознаки, як:

- кількість пакетів у прямому та зворотному напрямках,
- обсяг переданих байтів,
- інтенсивність трафіку (bytes rate, packets rate),
- характеристики TCP-заголовків (flags, window size, segment size),
- статистичні показники на кшталт варіації розміру сегментів або зміни довжини заголовка.

Ця група ознак описує поведінку мережевої взаємодії між сервісами у кластері.

Множина контейнерних метрик представляється наступним чином:

$$F_{net} = \{f_1^{ctr}, f_2^{ctr}, \dots, f_k^{ctr}\}, \quad (3)$$

і включає характеристики, що збираються агентом cAdvisor. До неї належать:

- використання CPU та доступні обмеження,
- обсяг виділеної та кешованої пам'яті,
- кількість відкритих дескрипторів,
- кількість мережевих сокетів,
- число процесів та потоків усередині контейнера.

Ці ознаки відображають поведінку контейнера як ізольованої обчислювальної одиниці.

Множина метрик стану кластера визначається як:

$$F_{net} = \{f_1^{k8s}, f_2^{k8s}, \dots, f_p^{k8s}\}, \quad (4)$$

куди входять дані, отримані за допомогою KubeStateMetrics та контролерів Kubernetes. До цієї групи належать:

- кількість доступних та бажаних Pod для кожного мікросервісу,
- статуси контейнерів та реплік,
- ліміти ресурсів та квоти (CPU, RAM),
- кількість перезапусків контейнерів,
- показники готовності та доступності сервісів.

Ці параметри описують стан оркестраційного середовища в цілому.

Після формування багатовимірного простіру ознак, що описує стан мережевої взаємодії, контейнерів та Kubernetes-кластера, було використано операцію імпутації пропущених значень. Для цього спочатку було обчислено вектори середніх значень для кожної ознаки j∈{1,…,n}, де визначається середнє значення за всіма наявними (непропущеними) елементами нормальної вибірки:

$$\mu_j = \frac{1}{|X_j|}\sum_{x \in X_j} x_j, \quad (5)$$

де

$$X_j = \{x \in X \mid x_j \neq \emptyset\}. \quad (6)$$

Після цього було виконано операцію заповнення пропущених значень, яку можна формально можна задати як відображення:

$$\Phi: X \to \mathbb{R}^n, \quad (7)$$

де для кожного вектора x∈X та кожної координати j:

$$\Phi(x)_j = \begin{cases} x_j, & \text{якщо } x_j \neq \emptyset, \\ \mu_j, & \text{якщо } x_j = \emptyset. \end{cases} \quad (8)$$

Операція імпутації для повної вибірки буде мати наступний вигляд

$$\mathrm{X}^{\mathrm{imp}} = \{\Phi(x) \mid x \in \mathrm{X}\} \quad (9)$$

Наступним етапом підготовки даних було застосовано операцію масштабування (стандартизації) для повної імпутованої вибірки.

Нехай після виконання операції імпутації отримано множину:

$$X^{imp} = \{\Phi(x) \mid x \in X\} \subset \mathbb{R}^n, \quad (10)$$

де кожен елемент $x^{imp} = (x_1^{imp}, \dots, x_n^{imp})$ є повністю визначеним вектором ознак. Для кожної ознаки j∈{1,…,n} визначаються середнє значення та стандартне відхилення:

$$\mu_j^{\mathrm{imp}} = \frac{1}{|X^{\mathrm{imp}}|}\sum_{x \in X^{\mathrm{imp}}} x_j \quad (11)$$

$$\sigma_j^{\mathrm{imp}} = \sqrt{\frac{1}{|X^{\mathrm{imp}}|}\sum_{x \in X^{\mathrm{imp}}} \left(x_j - \mu_j^{\mathrm{imp}}\right)^2} \quad (12)$$

Операцію масштабування визначимо як відображення:

$$\Psi: \mathbb{R}^n \to \mathbb{R}^n, \quad (13)$$

де для кожного вектора $x \in X^{imp}$ та кожного індексу j виконується наступна операція:

$$\Psi(x)_j = \frac{x_j - \mu_j^{\mathrm{imp}}}{\sigma_j^{\mathrm{imp}}} \quad (14)$$

Після застосування оператора Ψ до всієї імпутованої вибірки отримуємо масштабовану множину:

$$X^{\mathrm{scaled}} = \{\Psi(x) \mid x \in X^{\mathrm{imp}}\} \quad (15)$$

Таким чином, операція стандартизації переводить простір імпутованих ознак у нормалізований евклідів простір із нульовим математичним сподіванням та одиничною дисперсією для кожної координати, що забезпечує коректну та стабільну роботу моделей машинного та глибокого навчання.

Після виконання всіх етапів попередньої обробки даних, зокрема імпутації пропущених значень та стандартизації простору ознак, формується узгоджена та нормалізована вибірка. Яка вже придатна для подальшого моделювання. Водночас у задачах виявлення вторгнень спостерігається суттєвий дисбаланс між кількістю прикладів нормальної поведінки та атакувальних сценаріїв, а також обмеженість і неповнота множини відомих типів атак. З огляду на те, що в реальних умовах спектр атак постійно розширюється, а нові вектори загроз можуть бути відсутні у навчальних даних, доцільним є перехід до підходу виявлення аномалій.

У межах такого підходу для навчання моделі використовується виключно підмножина спостережень, що відповідають нормальній роботі системи. Тоді як усі відхилення від вивченої нормальної поведінки розглядаються як потенційні ознаки кібернетичного впливу на систему.

Для виявлення аномалій в функціонування Kubernetes запропоновано використовувати LSTM Autoencoder, який є однією з найкращих моделей для виявлення аномалій.

Після виділення множини ознак нормального функціонування Kubernetes було зроблено sliding window, тобто згрупування даних у вікна по N кроків. Це було зроблено тому, що мережні ознаки не є часовим рядом самі по собі. А LSTM Autoencoder – це різновид автоенкодера, побудований на основі рекурентних нейронних мереж з довготривалою короткочасною пам’яттю (Long Short-Term Memory, LSTM). Його основне призначений для моделювання та відтворення часових послідовностей даних.

Одже потрібно зробити sliding window, тобто групувати дані у вікна по N кроків. Було обрано ширину вікна 10, тобто модель бачить 10 послідовних записів як один приклад. Це ключовий момент до того, щоб LSTM навчилася ловити поведінкові патерни.

Для виявлення аномалій в функціонування Kubernetes було розроблено модель LSTM Autoencoder представлену на рисунку 3

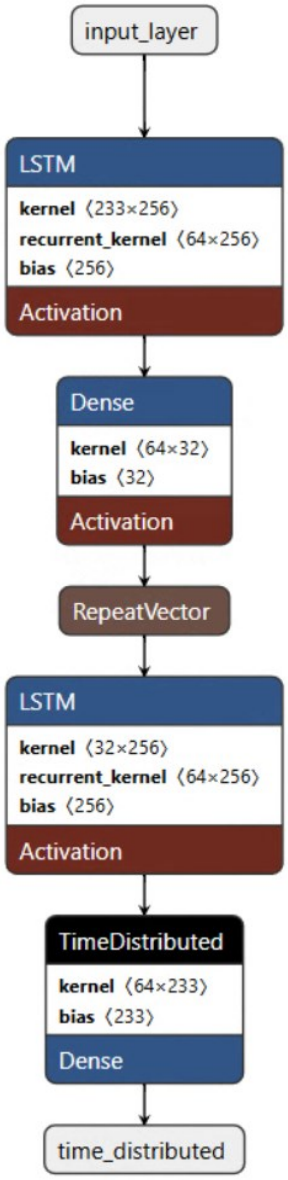

Рис. 3. Модель LSTM Autoencoder

Після навчання LSTM Autoencoder на підмножині даних, що відповідають нормальній роботі системи, модель використовується для відновлення (реконструкції) вхідних часових послідовностей ознак. Нехай $X^{\text{scaled}} = \{X^{(1)}, X^{(2)}, \ldots, X^{(N)}\}$ – множина масштабованих часових послідовностей, де кожен елемент $X^{(i)} \in \mathbb{R}^{T \times n}$ є послідовністю довжини T із n ознак.

Для кожної послідовності $X^{(i)}$ LSTM Autoencoder обчислює її реконструкцію

$$\hat{X}^{(i)} = \mathrm{D}\left(\mathrm{E}\left(X^{(i)}\right)\right), \tag{16}$$

де E та D відповідно позначають відображення енкодера та декодера.

Відхилення між вхідною послідовністю та її реконструкцією кількісно оцінюється за допомогою середньоквадратичної помилки:

$$\varepsilon^{(i)} = \frac{1}{T \cdot n} \sum_{t=1}^{T} \sum_{j=1}^{n} \left(X_{t,j}^{(i)} - \hat{X}_{t,j}^{(i)}\right)^2 \tag{17}$$

Отримане значення $\varepsilon^{(i)}$ інтерпретується як міра аномальності відповідної часової послідовності.

Для відокремлення нормальної та аномальної поведінки вводиться порогове значення θ. Значення порогу визначається на основі розподілу помилок реконструкції, отриманих для нормальної валідаційної вибірки. Формально поріг задається як:

$$\theta = \arg\max_{\tau} F_1\left(\mathrm{I}(\varepsilon > \tau), y\right) \tag{18}$$

де $\mathrm{I}()$ – індикаторна функція,

y – істинні мітки класів (норма / атака) на валідаційній множині,

F_1 – гармонічне середнє точності та повноти.

Такий підхід дозволяє підібрати поріг, який забезпечує оптимальний баланс між виявленням атак і кількістю хибних спрацювань.

Для кожної нової часової послідовності рішення про її належність до аномальної поведінки приймається за правилом:

$$\hat{y}^{(i)} = \begin{cases} 1, & \varepsilon^{(i)} > \theta, \\ 0, & \varepsilon^{(i)} \le \theta. \end{cases} \tag{19}$$

де $\hat{y}^{(i)} = 1$ відповідає аномальній або потенційно шкідливій активності, а $\hat{y}^{(i)} = 0$ – нормальній роботі системи.

Таким чином, процес реконструкції та порогової класифікації дозволяє звести задачу виявлення вторгнень до аналізу відхилень від вивченої нормальної поведінки. Оскільки модель навчається виключно на нормальних даних, підхід не залежить від повноти множини атак та зберігає здатність виявляти раніше невідомі або модифіковані загрози.

Експериментальне дослідження було проведено з метою оцінки ефективності запропонованого підходу виявлення аномалій у контейнеризованому середовищі Kubernetes на основі LSTM Autoencoder. Навчання моделі здійснювалося виключно на підмножині даних, що відповідають нормальній роботі системи, тоді як оцінка якості виконувалася на повній вибірці, яка містила як нормальні, так і атакувальні сценарії.

Після завершення етапу навчання для кожної часової послідовності було обчислено помилку реконструкції, на основі якої здійснювалася класифікація за допомогою порогового правила. Значення порогу аномальності визначалося шляхом максимізації F1-міри на валідаційній вибірці. У результаті оптимальний поріг було встановлено на рівні θ=0.47, що забезпечило найкращий баланс між точністю та повнотою виявлення атак.

Отримані результати класифікації свідчать про високу ефективність запропонованого підходу. Зокрема, для класу атак досягнуто значення recall = 0.95, що означає виявлення 95% усіх атакувальних послідовностей. Значення precision = 0.73 вказує на прийнятний рівень хибних спрацювань, що є типовим компромісом для систем виявлення вторгнень, орієнтованих на мінімізацію кількості пропущених атак. Відповідна F1-міра для атак склала 0.83, що є високим показником для unsupervised anomaly detection-підходів.

Загальна точність класифікації становила 0.76, а зважене середнє значення F1-міри – 0.74. Аналіз матриці помилок показав, що більшість помилок пов'язані з хибною класифікацією нормальної активності як аномальної, тоді як кількість пропущених атак є відносно низькою. Така поведінка моделі є очікуваною та прийнятною для систем безпеки, де пріоритетом є своєчасне виявлення потенційно шкідливої активності.

В таблиці 1 представлені основні показники якості

Таблиця 1

Показники якості виявлення аномалій за допомогою LSTM Autoencoder

Клас	Precision	Recall	F1-score	Кількість зразків
Нормальна активність (0)	0.86	0.45	0.59	26 679
Атакувальна активність (1)	0.73	0.95	0.83	42 705
Macro average	0.80	0.70	0.71	69 384
Weighted average	0.78	0.76	0.74	69 384

Загальна точність (Accuracy)	—	—	0.76	69 384

Отримані експериментальні результати підтверджують, що використання LSTM Autoencoder дозволяє ефективно моделювати часову динаміку нормальної поведінки Kubernetes-кластера та виявляти аномалії без необхідності попереднього знання всіх можливих типів атак. Це робить запропонований підхід придатним для практичного застосування в умовах постійної еволюції загроз та обмеженої доступності розмічених атакувальних даних.

Висновки з даного дослідження
і перспективи подальших розвідок у даному напрямі

В даному дослідженні було розроблено та експериментально обґрунтовано підхід до виявлення аномальної та потенційно шкідливої активності у Kubernetes-кластерах на основі методів глибокого навчання. Запропоноване рішення ґрунтується на використанні LSTM Autoencoder, який моделює часову динаміку нормальної поведінки контейнеризованого середовища та дозволяє виявляти відхилення без необхідності попереднього знання повної множини атак.

У ході роботи було формалізовано простір ознак, що поєднує мережеві характеристики, метрики контейнерів та показники стану Kubernetes-кластера, а також детально описано всі етапи попередньої обробки даних, включаючи імпутацію пропущених значень, стандартизацію та формування часових послідовностей. Навчання моделі здійснювалося виключно на даних нормальної роботи системи, що дозволило усунути проблему дисбалансу класів та забезпечити здатність моделі до виявлення раніше невідомих або модифікованих атак.

Експериментальні результати показали, що після оптимізації порогу аномальності за критерієм максимізації F1-міри LSTM Autoencoder забезпечує високий рівень виявлення атак (recall = 0.95) при F1-значенні 0.83. Такий результат підтверджує ефективність використання часових залежностей телеметричних даних для задач виявлення вторгнень і свідчить про доцільність застосування unsupervised-підходів у динамічних та слабо розмічених середовищах.

Перспективи подальших досліджень у даному напрямі пов’язані з удосконаленням архітектури моделі та розширенням її функціональних можливостей. Зокрема, доцільним є дослідження глибших та багатошарових LSTM Autoencoder, а також використання варіаційних автоенкодерів і гібридних підходів, що поєднують anomaly detection з керованими методами машинного навчання. Окрему увагу може бути приділено адаптивному визначенню порогу аномальності в режимі реального часу, інтеграції моделі з eBPF- та runtime-засобами безпеки Kubernetes, а також оцінці ефективності підходу в умовах edge- та multi-cloud-інфраструктур.

Література

1. Cherukuri, B. R. (2024). Containerization in cloud computing: Comparing Docker and Kubernetes for scalable web applications. International Journal of Scientific Research and Application, 13(01), 3302–3315. https://doi.org/10.30574/ijsra.2024.13.1.2035

2. Lin, L., Xiong, K., Wang, G., & Shi, J. (2024). AI-enhanced security for large-scale Kubernetes clusters: Advanced defense and authentication for national cloud infrastructure. Journal of Theory and Practice of Engineering Science, 4(12), 07. https://doi.org/10.53469/jtpes.2024.04(12).07

3. Aly, A., Fayez, M., Al-Qutt, M., & Hamad, A. M. (2024). Multi-class threat detection using neural network and machine learning approaches in Kubernetes environments. 6th International Conference on Computing and Informatics (ICCI), 103–108. https://doi.org/10.1109/ICCI61671.2024.10485133

4. Simonetto, S., & Bosch, P. (2024). Quantifying risk in the kill-chain: Automating threat prioritization for Kubernetes clusters. 10th Annual Cyber Security Next Generation Workshop, CSNG 2024. [Електронний ресурс]. Режим доступу: https://research.utwente.nl/files/468902786/csng_simonetto_1_.pdf (дата звернення: 08.12.2025)

5. Kulkarni, S. V. (2025). Securing Kubernetes: AI-powered container security agents. International Journal of AI, BigData, Computational and Management Studies, 6(1), 124–136. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I1P113

6. Bhardwaj, A. K., Dutta, P., & Chintale, P. (2024). AI-powered anomaly detection for Kubernetes security: A systematic approach to identifying threats. Babylonian Journal of Machine Learning, 142–148. https://doi.org/10.58496/BJML/2024/014

7. Morsli, R. (2025). Kubernetes Intrusion Detection Datasets (Kube-IDS0) [Електронний ресурс]. Режим доступу: https://www.kaggle.com/datasets/redamorsli/kube-ids0/code (дата звернення: 10.12.2025)

8. Morsli, R., Kara, N., Ould-Slimane, H., & Lahlou, L. (2025). Multidimensional intrusion detection system for containerized environments. IEEE 11th International Conference on Network Softwarization (NetSoft), 546–554. https://doi.org/10.1109/NetSoft64993.2025.11080585

References

1. Cherukuri, B. R. (2024). Containerization in cloud computing: Comparing Docker and Kubernetes for scalable web applications. International Journal of Scientific Research and Application, 13(01), 3302–3315. https://doi.org/10.30574/ijsra.2024.13.1.2035

2. Lin, L., Xiong, K., Wang, G., & Shi, J. (2024). AI-enhanced security for large-scale Kubernetes clusters: Advanced defense and authentication for national cloud infrastructure. Journal of Theory and Practice of Engineering Science, 4(12), 07. https://doi.org/10.53469/jtpes.2024.04(12).07

3. Aly, A., Fayez, M., Al-Qutt, M., & Hamad, A. M. (2024). Multi-class threat detection using neural network and machine learning approaches in Kubernetes environments. 6th International Conference on Computing and Informatics (ICCI), 103–108. https://doi.org/10.1109/ICCI61671.2024.10485133

4. Simonetto, S., & Bosch, P. (2024). Quantifying risk in the kill-chain: Automating threat prioritization for Kubernetes clusters. 10th Annual Cyber Security Next Generation Workshop, CSNG 2024. [Online resource]. Available at: https://research.utwente.nl/files/468902786/csng_simonetto_1_.pdf (Accessed: 08.12.2025).

5. Kulkarni, S. V. (2025). Securing Kubernetes: AI-powered container security agents. International Journal of AI, BigData, Computational and Management Studies, 6(1), 124–136. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I1P113

6. Bhardwaj, A. K., Dutta, P., & Chintale, P. (2024). AI-powered anomaly detection for Kubernetes security: A systematic approach to identifying threats. Babylonian Journal of Machine Learning, 142–148. https://doi.org/10.58496/BJML/2024/014

7. Morsli, R. (2025). Kubernetes Intrusion Detection Datasets (Kube-IDS0) [Online resource]. Available at: https://www.kaggle.com/datasets/redamorsli/kube-ids0/code (Accessed: 10.12.2025).

8. Morsli, R., Kara, N., Ould-Slimane, H., & Lahlou, L. (2025). Multidimensional intrusion detection system for containerized environments. IEEE 11th International Conference on Network Softwarization (NetSoft), 546–554. https://doi.org/10.1109/NetSoft64993.2025.11080585