МАТЕРІАЛИ XI ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: «ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ»

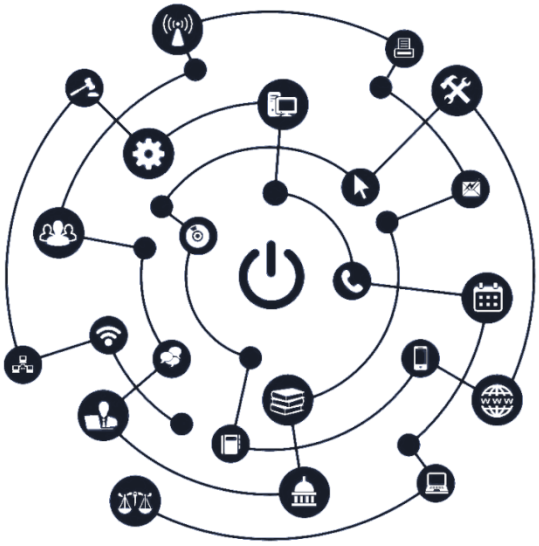
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

ОДЕСА
22 травня 2026 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Одеська юридична академія»
Факультет кібербезпеки та інформаційних технологій
Кафедра програмної інженерії



ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ

МАТЕРІАЛИ XI ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

22 травня 2026 року, м. Одеса

Одеса, 2026

УДК 340.12:316.3

Відповідальній редактор:
Н. І. Логінова, завідувачка кафедри програмної інженерії, канд. пед. наук, доцент
https://orcid.org/0000-0002-9475-6188

Матеріали видано в авторській редакції.
Повну відповідальність за достовірність та якість поданого матеріалу несуть учасники конференції, їхні наукові керівники, які рекомендували ці матеріали до друку

Друкується за рішенням кафедри програмної інженерії Національного університету «Одеська юридична академія» (протокол засідання кафедри № 5 від 18.05.2026 р.)

Інформаційне суспільство : проблеми та перспективи : Матеріали XI Всеукраїнської науково-практичної конференції (м. Одеса, 22 травня 2026 р.). Одеса : Національний університет «Одеська юридична академія», 2026. 175 с.
URL: ; DOI:

Всеукраїнська науково-практична конференція «Інформаційне суспільство: проблеми та перспективи» присвячена розгляду актуальних питань розвитку інформаційного суспільства в Україні та у світі.

До участі у конференції запрошені здобувачі вищої освіти, аспіранти, молоді вчені, викладачі та науковці. Конференція проводиться на базі Національного університету «Одеська юридична академія».

ПЕРЕДМОВА

Сучасний розвиток інформаційного суспільства характеризується стрімким поширенням цифрових технологій, які суттєво впливають на науку, освіту, економіку, державне управління та повсякденне життя. Особливого значення набувають дослідження у сферах штучного інтелекту, аналізу даних, програмної інженерії, хмарних обчислень, кібербезпеки та цифрової трансформації суспільства.

Матеріали, представлені у цьому збірнику, відображають широкий спектр актуальних наукових і практичних проблем сучасної ІТ-галузі. У роботах висвітлено фундаментальні аспекти інформатики та математичного забезпечення інформаційних технологій, сучасні підходи до розробки програмного забезпечення, створення веб- і мобільних застосунків, побудови інформаційних систем, використання баз даних і мережевих технологій.

Окремий блок досліджень присвячено штучному інтелекту, машинному навчанню, комп'ютерному зору, генеративним моделям та інтелектуальним системам. Значна увага приділяється також розробці комп'ютерних ігор, автоматизації бізнес-процесів та впровадженню цифрових сервісів у різних сферах діяльності.

Важливе місце у збірнику займають питання інформаційної безпеки та захисту інформації, зокрема кіберзлочинності, управління ризиками, криптографічного захисту, безпеки мереж і критичної інфраструктури. Поряд із технічними аспектами розглядаються правові, соціальні та етичні виклики цифровізації, регулювання цифрових платформ, використання штучного інтелекту та забезпечення цифрового суверенітету.

Представлені матеріали засвідчують міждисциплінарний характер сучасних досліджень у галузі інформаційних технологій та їхню вагому роль у розвитку суспільства. Сподіваємося, що результати, викладені у роботах учасників конференції, сприятимуть подальшому розвитку наукових досліджень, професійної співпраці та обміну досвідом між молодими науковцями, викладачами й фахівцями ІТ-сфери.

Секція 1. СУЧАСНІ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В ЖИТТІ ЛЮДИНИ ТА СУСПІЛЬСТВА

Підсекція 1. Теоретичні та фундаментальні основи ІТ

APPLICATION OF INFORMATION GEOMETRY TO CYBER INCIDENT ANALYSIS

Chepurna Olena

PhD in Physics and Mathematics, Associate Professor,
Department of Cybersecurity, National University «Odessa Law Academy»

Operational intrusion detection systems today process tens of thousands of network events per second yet still report false-positive rates that make automated response impractical. The surveys by Okolie et al. [1] and Sami [2] document this consistently across machine-learning-based approaches from 2019 to 2023: the underlying difficulty is not computational but conceptual. Most systems measure deviation in feature space using Euclidean distance, implicitly treating all coordinate directions as equally significant. This is not the case for probability distributions, which have a natural, coordinate-independent geometry that ordinary Euclidean distance ignores. When an attack manifests as a shift in the statistical character of traffic – a change in the shape of inter-arrival time histograms or in the entropy rate of payload bytes – a threshold on a scalar feature vector will miss the event or fire too late. What is needed is a principled metric on the space of distributions themselves.

Such a metric is provided by information geometry. Given a parametric family of densities $p(x;\,\theta), \theta \in \Theta \subseteq \mathbb{R}^{\mathrm{n}}$, the Fisher information matrix $g_{\mathrm{ij}}(\theta) = E_{\mathrm{x}}[\partial_{\mathrm{i}}\, log\, p \cdot \partial_{\mathrm{j}}\, log\, p]$ equips Θ with a Riemannian structure, turning it into a statistical manifold. As Nielsen explains in [3], this metric is the unique (up to positive scaling) Riemannian metric preserved under reparametrisation by sufficient statistics – that is, under any reformulation of the model that retains all statistical information in the data. The geodesic distance between two distributions on the manifold is therefore a measure of their intrinsic statistical distinguishability, not of their coordinate proximity, and an anomaly score based on this distance will not change merely because the analyst chose a different encoding for the traffic features. Within the exponential family – encompassing Gaussian, Poisson, and gamma distributions – the manifold admits a dually flat structure with analytically tractable geodesics and a Pythagorean identity for KL divergences.

Network events – connection attempts, DNS queries, ICMP packets – produce inter-arrival time sequences that are well described by the two-parameter gamma family $f(t;\,\mu,\kappa)$. The information geometry of this family is fully worked out, and the sub-manifold $\kappa = 1$ corresponds to the Poisson process, i.e., to memoryless, temporally unstructured arrivals. Geodesic distance from the Poisson sub-manifold

therefore provides a natural scalar anomaly score per monitoring window: $\kappa < 1$ indicates clustering of events in time, characteristic of DDoS burst traffic and port-scanning sweeps, while $\kappa > 1$ signals anomalous regularity typical of automated command-and-control heartbeat traffic. Bounding and computing these geodesic distances for the gamma and related families in practical implementations is addressed in [4], where approximation techniques are developed for parametric models where closed-form expressions are unavailable.

For high-dimensional feature vectors – incorporating packet sizes, inter-arrival times, port usage histograms, and flow-level statistics – dimensionality reduction on the statistical manifold reveals global topological structure that local methods cannot detect. Multi-dimensional scaling applied to the matrix of pairwise symmetrized KL divergences between successive monitoring windows embed all observed distributions in a low-dimensional Euclidean space while preserving their information-geometric separations. Nielsen [5] develops a systematic account of f-divergences, of which the symmetrized KL divergence is a special case, establishing their ordering properties relative to geodesic distance on non-flat manifolds. Anomalous windows appear as spatial outliers in the MDS embedding, separable by standard nearest-neighbour methods, whereas standard PCA applied to raw feature matrices produces flat Euclidean projections that discard the curvature of the distribution space and collapse statistically distinct distributions to nearby points.

Table 1. Information-geometric instruments for cyber incident analysis

Method / instrument	**Geometric meaning**	**Target incident class**
Fisher–Rao geodesic distance	Arc length on statistical manifold	Abrupt shift in traffic distribution
Symmetrised KL divergence	Approx. geodesic on non-flat manifold	Incremental log-based anomaly scoring
Gamma manifold / Poisson distance	Geodesic on 2-D gamma manifold	Event clustering or suspicious regularity
MDS on pairwise IG distances	Isometric embedding in Euclidean space	Topology anomalies, cluster separation
Maximum entropy projection	m-geodesic onto exponential sub-family	Baseline deviation in flow statistics

From the perspective of operational security architectures, the information-geometric scoring layer described above is conceptually compatible with entropy-based anomaly detection frameworks for cloud and IoT environments. Tosi and Pazzi [6] demonstrate that quantifying deviations from a trained distributional baseline via information-theoretic divergences achieves sub-second detection latency on large-scale heterogeneous networks. The information-geometric perspective adds formal justification for the choice of divergence: the symmetrized KL divergence is not

merely one heuristic among others but a computable approximation to the geodesic distance on the statistical manifold, and therefore the unique divergence that respects the intrinsic geometry of the distribution space. Placing such a geometry-grounded filter upstream of standard deep-learning classifiers reduces the dimensionality of the problem before it reaches the neural component and provides anomaly scores with a clear statistical interpretation, unlike the opaque feature distances typical of end-to-end architectures. Extending these methods to non-stationary baselines and encrypted traffic where only timing and size are observable remains the central open problem.

References:

1. Okolie S. A. Amadi C. A. Odii J. N. Nwokorie E. C. Onyemauche U. C. Anomaly detection in heterogeneous cybersecurity data. Franklin Open. 2025. Vol. 13. Art. 100426. DOI: 10.1016/j.fraope.2025.100426.
2. Sami A. A Comprehensive Investigation of Anomaly Detection Methods in Deep Learning and Machine Learning: 2019–2023. IET Information Security. 2024. P. 1–26. DOI: 10.1049/2024/8821891.
3. Nielsen F. The Many Faces of Information Geometry. Notices of the American Mathematical Society. 2022. Vol. 69, No. 1. P. 36–45. DOI: 10.1090/noti2403.
4. Nielsen F. Approximation and bounding techniques for the Fisher-Rao distances between parametric statistical models. Handbook of Statistics. Amsterdam : Elsevier, 2024. Vol. 51. P. 67–116. DOI: 10.1016/bs.host.2024.06.004.
5. Nielsen F. On f-Divergences Between Cauchy Distributions. IEEE Transactions on Information Theory. 2023. Vol. 69, No. 5. P. 3150–3171. arXiv:2101.12459.
6. Tosi D. Pazzi R. $(H\text{-}DIR)^2$: A Scalable Entropy-Based Framework for Anomaly Detection and Cybersecurity in Cloud IoT Data Centers. Sensors. 2025. Vol. 25, No. 15. Art. 4841. DOI: 10.3390/s25154841.

Keywords: information geometry, Fisher-Rao metric, statistical manifold, anomaly detection, f-divergence, gamma distribution, cyber incident analysis

Ключові слова: інформаційна геометрія, метрика Фішера-Рао, статистичний многовид, виявлення аномалій, f-дивергенція, розподіл гамма, аналіз кіберінцидентів

STATISTICAL MANIFOLDS AND THEIR APPLICATIONS IN DATA ANALYSIS

Cherevko Yevhen

PhD in Physics and Mathematics, Associate Professor,
Department of Artificial Intelligence and Mathematical Modeling,
National University «Odessa Law Academy»

A recurring difficulty in modern data analysis is that standard algorithms treat data as points in a Euclidean space, measuring distances with the ℓ- norm, while the

objects of actual interest are often probability distributions – empirical frequency histograms, fitted parametric models, covariance matrices from sensor arrays. Forcing such objects into Euclidean space by vectorizing their parameters discard the geometric structure that makes statistical distances meaningful. The Manifold Hypothesis, surveyed by Meilă and Zhang [1], asserts that nominally high-dimensional data concentrate near a low-dimensional nonlinear structure rather than filling the ambient space uniformly, so the relevant geometry is that of a curved manifold. The theory of statistical manifolds identifies, for any smooth parametric family of probability distributions, exactly this canonical Riemannian structure.

Let $\{p(x;\,\theta) : \theta \in \Theta \subseteq \mathbb{R}^n\}$ be a regular parametric statistical model. The Fisher information matrix $g_{ij}(\theta) = E_x[\partial_i \log p(x;\,\theta)\, \partial_j \log p(x;\,\theta)]$ equips Θ with a Riemannian metric known as the Fisher–Rao metric, turning the parameter space into a statistical manifold. As Nielsen shows in [2], this metric is the unique (up to positive scaling) Riemannian metric that is invariant under reparametrisation by sufficient statistics, i.e., it does not change when the model is reformulated in new coordinates if no statistical information is lost. The geodesic distance between two distributions on the manifold measures their intrinsic statistical distinguishability: its square equals the Rao lower bound on the expected squared error of any unbiased estimator, and geodesic balls of a given radius describe the set of distributions indistinguishable at a given significance level – a direct connection between differential geometry and hypothesis testing.

Alongside the Riemannian structure, the statistical manifold carries a family of α-connections with a duality between the (α)- and (−α)-connections with respect to the Fisher metric. The exponential family of distributions – encompassing Gaussian, Poisson, gamma, Bernoulli, and multinomial models – forms a dually flat manifold admitting two globally defined affine coordinate systems, the natural parameters η and the expectation parameters μ, linked by the Legendre–Fenchel transform of the cumulant function. In these coordinates the KL divergence between any two members of the family equals the Bregman divergence generated by the cumulant function, geodesics of each dual connection are straight lines, and the orthogonality of the two dual foliations implies a Pythagorean identity for KL divergences – the algebraic engine behind most practical applications of statistical manifolds in data analysis.

In classification and dimensionality reduction, replacing Euclidean pairwise distances with Fisher–Rao geodesic distances produce embeddings that preserve the statistical distinguishability of the input distributions rather than merely their coordinate proximity. Akgüller, Balcı, and Cioca [3] demonstrated this on a medical imaging dataset: converting grayscale MRI scans into Gaussian statistical manifolds via estimated mean vectors and covariance matrices, they computed pairwise geodesic distances with the Fisher information metric and used these as edge weights in Graph Neural Network classifiers. The geodesic distance accounts for differences in both mean and covariance structure simultaneously and is invariant to affine transformations of the feature space. All three graph architectures tested – GCN, GAT, and

GraphSAGE – achieved perfect precision and recall on four-class Alzheimer's severity data, while standard PCA-based representations produced substantial overlap between adjacent severity classes, a result explained by the non-zero curvature of the Gaussian manifold that PCA ignores.

In clustering, the duality between Bregman divergences and exponential families enables a principled generalization of k-means to non-Gaussian data. Standard k-means minimizes squared Euclidean distance, which is equivalent to minimizing the KL divergence between a Gaussian mixture and its components – appropriate for Gaussian clusters but poor for Poisson, Bernoulli, or gamma-distributed data. Vellal, Chakraborty, and Xu [4] developed Bregman power k-means, which replaces the squared Euclidean loss with the Bregman divergence of the target exponential family, maintaining closed-form centroid updates while relaxing bounded-support assumptions of prior work. In geometric terms, the centroid update is the m-geodesic projection of the cluster points onto the sub-manifold of the family, and the assignment step is the e-geodesic nearest-neighbour computation – exact geodesic geometry on the statistical manifold without approximation. Experiments on Poisson count data and rainfall intensity records showed substantially lower clustering error than Euclidean k-means under distributional shift.

Table 1. Information-geometric instruments in common data analysis tasks

Data analysis task	**Information-geometric instrument**
Dimensionality reduction	MDS / UMAP on Fisher–Rao pairwise distances
Clustering	Bregman divergence k-means on exponential family
Density estimation	m-geodesic projection (maximum entropy)
Classification	Geodesic distance as class-separability metric
Hypothesis testing	Log-likelihood ratio as manifold statistic

The instruments listed in Table 1 address standard data analysis tasks while replacing the implicit Euclidean assumption with a geometry derived from the statistical structure of the problem. In AI applications, the same curved geometry underlies natural gradient descent, where the Euclidean gradient of the training loss is replaced by its Fisher–Rao counterpart, accounting for the curvature of the parameter manifold and substantially accelerating convergence of neural network training. The main computational obstacle for extending these methods beyond the exponential family is the absence of closed-form geodesic expressions for general mixture and heavy-tailed distributions; bounding and approximation techniques for Fisher–Rao distances in such cases are the focus of ongoing work documented in [5]. Treating data as points on a curved statistical manifold and computing distances accordingly consistently improves over Euclidean baselines across tasks – from image classification

to count-data clustering to density estimation – whenever the data-generating family admits a characterizable manifold geometry.

References

1. Meilă M. Zhang H. Manifold Learning: What, How, and Why. Annual Review of Statistics and Its Application. 2024. Vol. 11. P. 393–417. DOI: 10.1146/annurev-statistics-040522-115238.
2. Nielsen F. The Many Faces of Information Geometry. Notices of the American Mathematical Society. 2022. Vol. 69, No. 1. P. 36–45. DOI: 10.1090/noti2403.
3. Akgüller Ö. Balcı M. A. Cioca G. Information Geometry and Manifold Learning: A Novel Framework for Analyzing Alzheimer's Disease MRI Data. Diagnostics. 2025. Vol. 15, No. 2. Art. 153.
4. Vellal A. Chakraborty S. Xu J. Bregman Power k-Means for Clustering Exponential Family Data. Proceedings of the 39th International Conference on Machine Learning. PMLR 162, 2022. URL: https://proceedings.mlr.press/v162/vellal22a.html (дата звернення: 14.05.2026).
5. Nielsen F. Approximation and bounding techniques for the Fisher-Rao distances between parametric statistical models. Handbook of Statistics. Amsterdam: Elsevier, 2024. Vol. 51. P. 67–116. DOI: 10.1016/bs.host.2024.06.004.

Keywords: statistical manifold, Fisher-Rao metric, information geometry, Bregman divergence, manifold learning, exponential family, data analysis

Ключові слова: статистичний многовид, метрика Фішера-Рао, інформаційна геометрія, дивергенція Брегмана, навчання на многовидах, експоненційна сімейство розподілів, аналіз даних

МАТЕМАТИЧНИЙ ФУНДАМЕНТ ШІ: ВІД ЛІНІЙНОЇ АЛГЕБРИ ДО ГЛИБОКОГО НАВЧАННЯ

Баландіна Наталія

старший викладач кафедри штучного інтелекту та математичного моделювання Національного університету «Одеська юридична академія»

Сучасний штучний інтелект (ШІ) перетворився з академічної новинки на технологічну основу багатьох галузей. Ефективність алгоритмів машинного навчання (ML) у аналізі складних даних, розпізнаванні патернів та прогнозуванні часто сприймається як обчислювальна алхімія, проте ця ілюзія приховує жорсткий математичний фундамент. Лінійна алгебра становить основну частину цього фундаменту.

Sadudeen M. [1] підкреслює, що лінійна алгебра є невід'ємною мовою машинного навчання, оскільки представлення даних, функціонування моделей та оптимізація є внутрішньо лінійно-алгебраїчними операціями: множення матриць, перетворення векторних просторів та маніпуляції з тензорами.

Метою роботи є систематичний аналіз застосування концепцій лінійної алгебри в алгоритмах машинного та глибокого навчання – від базових операцій з

матрицями та векторами до сучасних архітектур трансформерів і нейронних мереж на графах, з акцентом на матричні операції та матричне числення для оптимізації.

У машинному навчанні дані представлені у вигляді матриці X, де кожен рядок відповідає окремому об'єкту (зображенню, тексту, сигналу), а кожен стовпець – окремій ознаці. Модель ML виконує перетворення цієї матриці через множення на вагові матриці та додавання векторів зміщення.

Sadudeen M. [1] систематично демонструє, що базові алгоритми ML, такі як лінійна регресія, метод опорних векторів (SVM) та метод головних компонент (PCA), повністю описуються через лінійно-алгебраїчні операції. Зокрема, PCA базується на спектральному розкладі коваріаційної матриці та задачі на власні значення, що є класичною задачею лінійної алгебри. Автор наголошує, що навіть нейронні мережі, які часто сприймаються як "чорна скринька", по суті є послідовністю матричних множень з нелінійними активаціями, а їх навчання – оптимізацією у багатовимірному просторі параметрів.

Глибоке навчання базується на послідовності афінних перетворень з нелінійними активаціями. Baggag A. і Saad Y. [2] подають загальний вигляд шару нейронної мережі як

$$Z_l = \sigma(Z_{l-1}W_l + b_l), \quad (1)$$

де Z_{l-1}– виходи попереднього шару, W_l – матриця вагів, b_l – вектор зміщення, $\sigma(\cdot)$ – нелінійна функція активації. Множення матриць $Z_{l-1}W_l$ є центральною обчислювальною операцією, що визначає продуктивність глибоких мереж.

Baggag A. і Saad Y. [2] показують, що майже всі важливі сучасні вдосконалення алгоритмів ШІ базуються на простих ідеях чисельної лінійної алгебри (NLA). Метод низькорангової адаптації (LoRA) використовує низькорангову апроксимацію вагових матриць, що дозволяє знизити витрати на навчання та інференс на порядки величин без втрати точності. Це робить можливим тренування моделей з мільярдами параметрів.

Механізм уваги, який є основою архітектури трансформерів, описується формулою, наведеною Baggag A. і Saad Y. [2]:

$$\text{Attention}(Q, K) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right), \quad (2)$$

де Q – матриця запитів, K – матриця ключів, d_k – розмірність ключів. Операція QK^T– матричне множення, що обчислює скалярні добутки між запитами та ключами, визначаючи важливість різних частин входу. Функція softmax нормалізує ці значення у ймовірнісний розподіл. Автори також досліджують лінійну увагу (Linear Attention), яка апроксимує цей нелінійний механізм лінійним, зменшуючи обчислювальну складність з $O(n^2)$ до $O(n)$.

Порівняння обчислювальної складності та областей застосування основних архітектур ML представлено в табл. 1.

Таблиця 1 – Лінійно-алгебраїчні операції в архітектурах машинного навчання

Архітектура	**Базова операція**	**Складність**	**Застосування**
Лінійна регресія / SVM	Матричне множення	$O(nm)$	Прогнозування
PCA	Спектральний розклад	$O(m^3)$	Редукція розмірності
MLP (FFN)	Афінне перетворення з активацією	$O(nd^2)$	Класифікація
Transformer	Механізм уваги (формула 2)	$O(n^2d)$	NLP, Vision
GNN	Згортка на графі	$O(\|E\|d^2)$	Графові дані

Навчання нейронних мереж відбувається через алгоритм зворотного поширення помилки (backpropagation), який обчислює градієнти функції втрат за параметрами моделі. Bright P., Edelman A. і Johnson S. G. [3] у лекційних нотатках з матричного числення для машинного навчання систематизують методи диференціювання функцій на загальних векторних просторах. Зокрема, автори розглядають диференціювання функцій, які приймають матрицю як вхід та повертають іншу матрицю (наприклад, обернену матрицю або матричну факторизацію), що є нетривіальним розширенням класичного диференціального числення.

Bright P., Edelman A. і Johnson S. G. [3] вводять поняття "спряженого" (adjoint) або "зворотного режиму" диференціювання, яке у ML відоме як backpropagation. Цей підхід дозволяє ефективно обчислювати градієнти навіть для складних обчислювальних графів з мільярдами параметрів через послідовні матрично-векторні добутки за ланцюговим правилом. Автори також розглядають диференціювання рішень звичайних диференціальних рівнянь (ODE) та стохастичних похідних випадкових функцій, що є основою сучасних методів, таких як нейронні ODE (Neural ODEs) та стохастичні градієнтні методи навчання.

Sadudeen M. [1] узагальнює, що розуміння математичних основ ML не є суто академічною вправою, а є необхідною умовою для практичного опанування галузі. Дослідники, які глибоко розуміють лінійно-алгебраїчну природу алгоритмів, можуть створювати нові архітектури, а практики – ефективніше налагоджувати та оптимізувати свої ML-конвеєри. Це особливо актуально для нейронних мереж на графах (GNN), які Baggag A. і Saad Y. [2] описують як природне розширення трансформерів на нерегулярні структури даних, де матриця суміжності графа використовується для агрегації інформації від сусідніх вершин, що ґрунтується на спектральному аналізі та методах вбудовування (embeddings).

У роботі проведено систематичний аналіз застосування лінійної алгебри як математичного фундаменту алгоритмів машинного та глибокого навчання. Показано, що всі ключові операції в ML – від лінійної регресії та PCA до трансформерів і нейронних мереж на графах – базуються на матричних операціях: множенні матриць, спектральних розкладах та операціях у векторних просторах. Механізм уваги, основа сучасних моделей обробки мови та комп'ютерного зору,

повністю описується через матричні добутки та функцію softmax. Методи низькорангової адаптації (LoRA) та лінійної уваги демонструють, що прості ідеї чисельної лінійної алгебри призводять до значних покращень ефективності без втрати точності. Матричне числення забезпечує формальний апарат для зворотного поширення помилки та автоматичного диференціювання, що є критичним для навчання глибоких мереж з мільярдами параметрів. Подальші дослідження повинні бути спрямовані на інтеграцію складніших методів чисельної лінійної алгебри в архітектури ШІ для досягнення нових проривів у продуктивності та масштабованості.

Список використаних джерел

1. Sadudeen M. Linear Algebra as a Mathematical Foundation for Machine Learning Algorithms. International Journal of Innovative Science and Research Technology. 2025. Vol. 10, No. 9. P. 2257–2262. DOI: 10.38124/ijisrt/25sep829.
2. Baggag A., Saad Y. Deep learning, transformers and graph neural networks: a linear algebra perspective. Numerical Algorithms. 2025. Vol. 100, No. 4. P. 2095–2134. DOI: 10.1007/s11075-025-02218-2.
3. Bright P., Edelman A., Johnson S. G. Matrix Calculus (for Machine Learning and Beyond) (Version 1). arXiv. 2025. DOI: 10.48550/ARXIV.2501.14787.

Ключові слова: лінійна алгебра, машинне навчання, глибоке навчання, трансформери, матричне числення

Keywords: linear algebra, machine learning, deep learning, transformers, matrix calculus

STD::VECTOR ПРИ ОПРАЦЮВАННІ БАГАТОВИМІРНИХ ДАНИХ

Кунченко Богдан

студент 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У сучасній інженерії програмного забезпечення швидкодія прикладних програм дедалі більше визначається взаємодією процесора з підсистемою пам’яті (феномен «Memory Wall»). У мові C++ використання контейнерів STL, зокрема `std::vector`, стало еталоном безпеки завдяки парадигмі RAII, що нівелює витоки пам’яті [1]. Однак високий рівень абстракції маскує низькорівневі апаратні процеси. Застосування вкладених структур для моделювання багатовимірних даних може призводити до затримок кешування та виділення пам’яті, хоча ця модель залишається зручною та гнучкою для задач із нерівномірною довжиною рядків або динамічною зміною структури матриці.

На рівні компілятора `std::vector` є обгорткою над низькорівневим динамічним масивом суміжного розташування і в більшості сучасних реалізацій стандартної бібліотеки складається з трьох базових вказівників (на початок

блоку, на кінець фактичних даних та на кінець зарезервованої пам'яті). Одновимірний вектор забезпечує час доступу $O(1)$. Проте при моделюванні двовимірних матриць стандартним підходом є конструкція `std::vector<std::vector<T>>`.

З погляду системної архітектури зовнішній вектор зберігає лише набір дескрипторів внутрішніх векторів. Кожен рядок такої матриці алокує пам'ять у купі (heap) незалежно. Як наслідок, для окремих рядків не гарантується суміжне розташування в пам'яті, що може знижувати ефективність просторової локальності та збільшувати затримки доступу до пам'яті.

Передача даних між оперативною пам'яттю (ОП) та кристалом процесора здійснюється фіксованими блоками – кеш-лініями (зазвичай 64 байти). Ефективність обчислень базується на принципі просторової локальності (Spatial Locality): якщо програма звернулася до адреси X, наступним очікується звернення до сусідньої адреси $X + 1$.

При обході матриці `vector<vector<T>>` перехід між рядками може порушувати цей принцип та збільшувати ймовірність промаху (Cache Miss), оскільки адреса нового рядка фізично не суміжна з попереднім [2, стор. 34]. Процесор може витрачати десятки або сотні тактів на завантаження нової кеш-лінії з ОП. У високонавантажених алгоритмах велика кількість Cache Miss збільшує затримки виконання обчислень, паралельно роблячи апаратний префетчер (Hardware Prefetcher) неефективним через непослідовне розташування блоків пам'яті.

Попри геометричне зростання ємності (`capacity`) та амортизовану константну складність операції додавання, при вичерпанні поточного буфера виклик `push_back()` спричиняє реаллокацію: виділення нової пам'яті, копіювання елементів та знищення старого масиву [3, стор. 898]. Якщо розмірність матриці заздалегідь невідома, у структурі вкладених векторів ці ресурсомісткі операції реаллокації виконуються багаторазово для кожного окремого рядка, збільшуючи навантаження на систему керування пам'яттю.

Крім того, структурна модифікація вектора (видалення елементів методом `.erase()`) потребує зсуву наступних елементів контейнера, що супроводжується додатковими витратами часу. Ця операція має лінійну складність $O(N)$. За умови частого застосування таких операцій структурної модифікації у вкладених циклах багатовимірних масивів, складність алгоритму може наближатися до квадратичної $O(M * N^2)$, що є неприпустимим для обчислювальних систем.

Для усунення вищезазначених проблем у системах High-Performance Computing застосовують метод лінеаризації. Двовимірна матриця розміром $M * N$ розгортається в один плоский одновимірний вектор `std::vector<T>` загальною довжиною $M * N$. Перерахунок двовимірних координат (i, j) у лінійний індекс здійснюється програмно за формулою:

$$Index = i * N + j$$

Переваги лінеаризації є комплексними:

- максимізація Cache Hit Rate: елементи контейнера розташовуються в одному суцільному блоці пам'яті. Це зменшує ймовірність промахів кешу при послідовному обході даних;
- активація Hardware Prefetcher: крок доступу до пам'яті стає строго лінійним, що дозволяє процесору успішно прогнозувати і підвантажувати дані заздалегідь;
- оптимізація алокацій: замість сотень запитів виділяється лише один загальний блок пам'яті;
- резервування: виклик `matrix.reserve(M*N)` одразу виділяє потрібний обсяг пам'яті, запобігаючи повторним реаллокаціям під час подальшої ініціалізації (наприклад, через `push_back`).

Практичні мікробенчмарки в задачах послідовного обходу матриць демонструють, що лінеаризоване подання даних зазвичай забезпечує меншу кількість Cache Miss та нижчу латентність доступу до пам'яті, порівняно з вкладеними структурами `vector<vector<T>>`. Порівняльні характеристики розглянутих підходів наведено у табл. 1.

Таблиця 1 – Порівняльний аналіз архітектурних підходів обробки матриць

Критерій порівняння	**Вкладені вектори `vector<vector<T>>`**	**Лінеаризований вектор `vector<T>`**
Топологія в ОП	Фрагментація, рядки розкидані в купі	Суцільний, неперервний блок пам'яті
Ефективність кешування CPU	Низька (висока частота Cache Misses)	Максимальна (висока частота Cache Hits)
Кількість алокацій пам'яті	Щонайменше M+1 алокацій, при реалокації зовнішнього вектора кількість може зрости	Єдине виділення суцільного блоку пам'яті
Робота Hardware Prefetcher	Суттєво втрачає ефективність через нерегулярний характер доступу до пам'яті	Працює більш ефективно
Складність доступу за індексом	Два рівні розіменування вказівників	Прямий розрахунок адреси за формулою

Контейнер `std::vector` є потужним інструментом безпечного управління ресурсами, проте використання вкладених структур (`vector<vector<T>>`) без урахування особливостей доступу до пам'яті знижує ефективність просторової локальності, збільшує затримки доступу до пам'яті та кількість операцій виділення пам'яті. Для створення високоефективного та масштабованого програмного забезпечення у задачах, чутливих до продуктивності пам'яті, доцільно використовувати лінеаризацію багатовимірних даних в одновимірні суцільні контейнери з попереднім резервуванням пам'яті. Це дозволяє поєднати безпеку сучасного стандарту C++ із максимальною продуктивністю на рівні апаратного забезпечення. Попри переваги лінеаризації з погляду

продуктивності, вкладені вектори можуть забезпечувати кращу читабельність коду, простішу індексацію та вищу гнучкість при роботі зі структурами змінної розмірності.

Варто зазначити, що стандарт C++23 пропонує сучасний підхід до роботи з багатовимірними даними за допомогою абстракції `std::mdspan` [4]. Цей інструмент є невласницьким (non-owning) багатовимірним поданням (view) над суцільним одновимірним масивом даних. Використання `std::mdspan` дозволяє поєднати високу ефективність просторової локальності плоского масиву зі зручністю багатовимірної індексації без потреб ручного обчислення лінійного індексу. Це частково усуває проблему зниження читабельності коду, характерну для класичної лінеаризації.

Список використаних джерел

1. RAII. *cppreference.com*. URL: https://en.cppreference.com/cpp/language/raii (дата звернення: 22.05.2026).
2. Drepper U. What Every Programmer Should Know About Memory. Red Hat, Inc., 2007. 114 p. URL: https://people.freebsd.org/~lstewart/articles/cpumemory.pdf
3. Stroustrup B. The C++ Programming Language. 4th ed. Upper Saddle River: Addison-Wesley, 2013. 1366 p. URL: https://chenweixiang.github.io/docs/The_C++_Programming_Language_4th_Edition_Bjarne_Stroustrup.pdf
4. ISO/IEC 14882:2024(E) – Programming Language C++ 23. URL: https://isocpp.org/std/the-standard (дата звернення: 22.05.2026).

Ключові слова: `std::vector`, `std::mdspan`, багатовимірні дані, просторова локальність, промах кешу, кеш-ефективність, високопродуктивні обчислення, лінеаризація даних, C++23

Keywords: `std::vector`, `std::mdspan`, multidimensional data, spatial locality, cache miss, cache efficiency, high-performance computing, data linearization, C++23

Науковий керівник: доцент Трофименко О.Г.

КЛАСИФІКАЦІЯ ТОПОЛОГІЙ «P2P» МЕРЕЖ У СУЧАСНИХ СИСТЕМАХ ЗВ'ЯЗКУ

Максименко Артем

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Сьогодні Peer-to-Peer мережі є фундаментом для багатьох децентралізованих систем: від протоколів обміну файлами до захищених месенджерів та блокчейн-технологій. На відміну від традиційної клієнт-серверної моделі, де ресурси зосереджені на виділених серверах, P2P-архітектура розподіляє навантаження між рівноправними учасниками [1].

Термін «P2P» охоплює широкий спектр технологій. В академічній літературі виділяють різні види P2P-архітектур, кожна з яких має свої переваги та недоліки залежно від умов застосування:

1. *Централізовані P2P-мережі*. У цій моделі є центральний сервер, але він не бере участі в передачі самих даних. Сервер зберігає лише індексну базу, як адресна книга, яка дозволяє вузлам знаходити один одного [2]. Після отримання IP-адреси співрозмовника передача інформації відбувається напряму. Історичним прикладом є перша версія мережі Napster. Головний недолік архітектури – наявність єдиної точки відмови на рівні індексації.

2. *Неструктуровані або «чисті» P2P-мережі*. У таких мережах взагалі немає виділених серверів, а всі вузли абсолютно рівноправні. Пошук інформації або співрозмовника здійснюється методом «затоплення» (flooding), вузол надсилає запит усім своїм сусідам, ті – своїм, і так далі. Хоча ця архітектура є максимально стійкою до цензури, вона генерує величезні обсяги надлишкового мережевого трафіка та погано масштабується [3].

3. *Структуровані P2P-мережі*. Вони вирішують проблему надлишкового трафіка шляхом використання розподілених хеш-таблиць, протоколів Chord або Kademlia. Кожен пристрій отримує унікальний ідентифікатор, а топологія мережі строго організована. Це гарантує пошук будь-якого вузла за логарифмічний час [1]. Структуровані мережі ідеально підходять для глобальних файлових систем та маршрутизації у блокчейнах.

4. *Мережі з супервузлами*. Ця архітектура враховує той факт, що не всі пристрої в мережі мають однакову пропускну здатність і стабільність. Найбільш потужні комп'ютери з відкритими IP-адресами стають «супервузлами», які беруть на себе функції локальної індексації та маршрутизації для групи звичайних, слабших клієнтів, наприклад, мобільних пристроїв [4]. Класичним прикладом є протокол Skype до 2012 року.

5. *Гібридні P2P-мережі*. Це сучасний, найбільш прагматичний підхід, який гнучко поєднує елементи P2P та клієнт-серверної архітектури для подолання реальних обмежень інтернету, зокрема трансляції мережевих адрес. У гібридній моделі допоміжний сервер може виступати не лише як індексатор, а і як резервний ретранслятор, якщо пряме з'єднання між вузлами встановити технічно неможливо [5]. Саме цей підхід використовується під час проєктування сучасних захищених месенджерів із наскрізним шифруванням.

Загалом еволюція P2P-мереж демонструє перехід від спроб створити абсолютно децентралізовані, але енергозатратні системи, до прагматичних гібридних та супервузлових моделей. Саме останні типи архітектур дозволяють забезпечити високу швидкість, економію заряду акумуляторів та надійність з'єднання, зберігаючи при цьому ключову перевагу P2P конфіденційну передачу даних без збереження вмісту на сторонніх серверах.

Список використаних джерел

1. Hassanzadeh-Nazarabadi Y., Taheri-Boshrooyeh S., Otoum S., Ucar S., Özkasap Ö. DHT-based Communications Survey: Architectures and Use Cases. *arXiv*:2109.10787. 2021. DOI: https://doi.org/10.48550/arXiv.2109.10787
2. Masinde N., Graffi K. Peer-to-Peer-Based Social Networks: A Comprehensive Survey. *SN Computer Science*. 2020. Vol. 1, No. 5. Art. 299. DOI: https://doi.org/10.1007/s42979-020-00315-8
3. Fazea Y., Attarbash Z.S., Mohammed F., Abdullahi I. Review on Unstructured Peer-to-Peer Overlay Network Applications. *2021 International Conference of Technology, Science and Administration (ICTSA)*. Taiz, Yemen, 22–24 March 2021. DOI: https://doi.org/10.1109/ICTSA52017.2021.9406524
4. Amirazodi N., Saghiri A.M., Meybodi M.R. A self-adaptive algorithm for super-peer selection considering the mobility of peers in cognitive mobile peer-to-peer networks. *International Journal of Communication Systems*. 2021. Vol. 34. e4661. DOI: https://doi.org/10.1002/dac.4661
5. Wanli Zh., Fujita S. Hashgraph Over the Edge: Achieving Byzantine Fault Tolerance in Decentralized P2P Frameworks. *IEEE Access*. 2025. Vol. 13. P. 153036-153055. DOI: https://doi.org/10.1109/access.2025.3604512

Ключові слова: P2P-мережі, структуровані мережі, неструктуровані мережі, гібридна архітектура, супервузли, розподілені хеш-таблиці

Keywords: P2P networks, structured networks, unstructured networks, hybrid architecture, super-peers, distributed hash tables

Науковий керівник: доцент Трофименко О.Г.

Підсекція 2. Архітектури, інфраструктура та технології розробки

ДОЦІЛЬНІСТЬ ВИКОРИСТАННЯ ДЕКІЛЬКОХ БАЗ ДАНИХ В ОДНОМУ ПРОЄКТІ

Карагуц Олександр

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У сучасній розробці програмного забезпечення дедалі частіше застосовується підхід polyglot persistence – використання декількох типів баз даних та сервісів збереження інформації в межах одного проєкту. Такий підхід дозволяє ефективніше вирішувати різні задачі: зберігання структурованих та неструктурованих даних, кешування, повнотекстовий пошук, обробку потоків подій, роботу з файлами, зображеннями, логами, метриками, тощо.

Сучасні вебсистеми та мікросервісні архітектури мають високі вимоги до продуктивності, масштабованості та швидкості обробки даних. Використання лише однієї бази даних часто не дозволяє ефективно вирішити всі задачі проєкту.

Саме тому набуває популярності підхід комбінування різних технологій збереження даних.

Актуальність використання polyglot persistence підтверджується сучасними тенденціями у сфері розробки програмного забезпечення. Згідно з дослідженнями архітектур мікросервісів [1], близько 52% мікросервісних систем використовують декілька типів баз даних одночасно. Найчастіше комбінуються реляційні, документно-орієнтовані, бази даних ключ-значення та пошукові бази.

Аналітики також відзначають швидке зростання популярності polyglot persistence у cloud-native системах. За оцінками Gartner та IDC, до 2027 року близько 65% нових enterprise-застосунків будуть використовувати polyglot-архітектури, а понад 80% великих компаній уже використовують декілька платформ збереження даних для різних типів навантаження [2].

Основною перевагою використання різних баз даних є можливість використовувати оптимальну технологію для конкретного типу задачі. Різні бази даних мають власні сильні сторони, тому одна універсальна система часто не може ефективно покривати всі потреби застосунку. Використання декількох баз даних дозволяє: підвищити продуктивність системи, зменшити навантаження на основну базу даних, масштабувати окремі сервіси незалежно, покращити відмовостійкість, спростити обробку великих обсягів даних, оптимізувати витрати ресурсів [3].

Разом із перевагами використання кількох баз даних створює і певні труднощі: ускладнюється підтримка інфраструктури, зростають вимоги до DevOps-процесів, моніторингу та синхронізації даних між сервісами. Тому застосування polyglot persistence повинно бути обґрунтованим і відповідати реальним потребам проєкту.

У розподілених системах часто підтримка миттєвої узгодженості даних між усіма сервісами є складним та ресурсозатратним завданням. У високонавантажених мікросервісних архітектурах пріоритет часто надається масштабованості, доступності та відмовостійкості системи. Саме тому сучасні системи дедалі частіше використовують модель eventual consistency – поступової узгодженості даних.

Eventual consistency досягається завдяки асинхронній синхронізації даних між окремими сервісами або базами даних. Основним способом реалізації eventual consistency є використання event-driven architecture, де сервіси обмінюються подіями через брокери повідомлень, наприклад Apache Kafka [4]. Apache Kafka часто сприймається не лише як брокер повідомлень, а і як спеціалізована розподілена база даних для роботи з потоками подій. Це пов’язано з тим, що Kafka не просто передає повідомлення між сервісами, а зберігає їх протягом тривалого часу та дозволяє повторно читати дані.

Відомими прикладами використання polyglot persistence є великі технологічні компанії, зокрема Netflix та Uber. Через величезну кількість користувачів і високі навантаження такі компанії використовують декілька типів баз даних та

систем обробки даних одночасно. Так, Netflix використовує одразу декілька спеціалізованих систем зберігання та обробки даних, що є прикладом polyglot persistence: Apache Kafka використовується для потокової передачі та зберігання мільйонів подій між мікросервісами, Apache Flink застосовується для обробки потоків даних у реальному часі, Apache Cassandra використовується як масштабоване розподілене сховище даних, графові бази даних – для аналізу зв'язків між користувачами, контентом та подіями, бази даних типу «ключ-значення» – для надшвидкого доступу до інформації [5]. Також Netflix активно використовує event-driven architecture та eventual consistency для забезпечення масштабованості і відмовостійкості системи.

Згідно зі статтею, Uber побудував власну високонавантажену систему зберігання даних під назвою Docstore, яка обслуговує десятки мільйонів запитів щосекунди. Основою цієї системи є MySQL, поверх якого реалізована масштабована архітектура для роботи мікросервісів. Для обробки великої кількості запитів Uber використовує: Redis – як високошвидкісний кеш, CacheFront – внутрішню систему інтегрованого кешування, Flux – систему CDC (Change Data Capture), яка синхронізує зміни між MySQL та кешем Redis. Така архітектура дозволила Uber обробляти понад 40 мільйонів читань за секунду та значно зменшити навантаження на основну базу даних [6].

Polyglot persistence поступово стає стандартом для побудови сучасних високонавантажених інформаційних систем, оскільки дозволяє поєднувати переваги різних моделей зберігання даних та ефективно адаптувати архітектуру системи до вимог конкретного бізнес-застосунку. Комбінування реляційних, документо-орієнтованих, широко-колонкових, пошукових та інших спеціалізованих баз даних забезпечує високу продуктивність, масштабованість, відмовостійкість та гнучкість системи. Водночас впровадження polyglot persistence потребує складнішої інфраструктури, додаткового моніторингу та грамотного проєктування архітектури системи. Подальший розвиток мікросервісних і cloud-native архітектур сприятиме активнішому впровадженню спеціалізованих баз даних і систем потокової обробки даних у межах одного проєкту.

Список використаних джерел

1. André M., Raglianti M., et al. An Empirical Study on Database Usage in Microservices. *Arxiv*. 2025. DOI: https://doi.org/10.48550/arXiv.2510.20582
2. The Future is Polyglot: Why Businesses Use Multiple Databases. URL: https://technotes.in/the-future-is-polyglot-why-businesses-use-multiple-databases
3. Risi C. Modernizing Data: From Relational Databases to Polyglot Persistence. URL: https://www.linkedin.com/pulse/modernizing-data-from-relational-databases-polyglot-persistence-risi-lr9gf/
4. Rahul K. Eventual Consistency in Microservices. *Medium*. URL: https://medium.com/@27.rahul.k/eventual-consistency-in-microservices-83f3a7b82e2f

5. How and Why Netflix Built a Real-Time Distributed Graph: Part 1 Ingesting and Processing Data Streams at Internet Scale. *Medium*. URL: https://netflixtechblog.com/how-and-why-netflix-built-a-real-time-distributed-graph-part-1-ingesting-and-processing-data-80113e124acc
6. Pozniansky E., Patel P., Narayanareddy P. Preetham Narayanareddy How Uber Serves Over 40 Million Reads Per Second from Online Storage Using an Integrated Cache. *Uber Blog*. URL: https://www.uber.com/us/en/blog/how-uber-serves-over-40-million-reads-per-second-using-an-integrated-cache

Ключові слова: поліглотна стійкість, кінцева консистенція, мікросервісна архітектура, бази даних, поліглот-архітектура

Keywords: polyglot resilience, eventual consistency, microservice architecture, databases, polyglot architecture

Науковий керівник: доцент Трофименко О.Г.

СУЧАСНІ ПРОБЛЕМИ ВИКОРИСТАННЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ

Щербина Юрій

кандидат технічних наук, доцент,
доцент кафедри програмної інженерії
Національного університету «Одеська юридична академія»

Не зважаючи на те, що протягом останніх десятиліть було створено декілька типів баз даних, орієнтованих на роботу в хмарних технологіях з мало структурованими даними, в основу яких закладено теорему CAP (Consistency, Availability, Partition tolerance) [1], основним робочим інструментом, що використовується для зберігання та управління важливою інформацією залишаються такі реляційні бази даних, як Oracle, MySQL, Microsoft SQL Server та PostgreSQL, що побудовані на основі теореми ACID (Atomicity, Consistency, Isolation, Durability) [2]. Вони надійніше забезпечують роботу в сценаріях, що вимагають обслуговування складних запитів, цілісності даних та узгодженості транзакцій.

Системи управління реляційними базами даних (СУБД) передбачають роботу з великими об'ємами даних і, в основному, використовуються крупними корпоративними об'єднаннями, що включають велику кількість відділів і підрозділів. Інформація, якою вони користуються зберігається в декількох сховищах та використовується різними командами фахівців.

У порівнянні з базами даних типу NoSQL, реляційні бази даних мають суттєві переваги у тих ситуаціях, де робота з даними вкрай формалізована і висуваються високі вимоги до їх захисту. До складу таких переваг відносять:

- простоту моделі, що будується на використанні достатньо простих запитів, написаних мовою SQL;
- простоту використання, що забезпечується високою швидкістю виконання транзакцій, яка не залежить від складності моделі;

- відсутність дублювання даних, що гуртується на способі побудови архітектури реляційних баз даних;
- забезпечення цілісності даних, що гуртується на забезпеченні узгодженості між усіма зв'язаними таблицями;
- нормалізацію, яка передбачає розділення даних на окремі частини для зменшення розміру необхідних об'ємів пам'яті;
- можливість забезпечення з боку СУБД одночасного доступу до бази даних кількох користувачів;
- забезпечення розподіленого доступу до даних з боку користувачі, з метою захисту даних.

Що стосується обмежень, притаманних реляційним базам даних, то вони включають:

- ускладнення підтримки бази даних по мірі збільшення об'ємів інформації, яка з часом накопичується в її пам'яті;
- відносно висока вартість розгорнення та обслуговування;
- необхідність великої кількості фізичної пам'яті для зберігання даних, розподілених у таблицях;
- обмеженість можливості масштабування на кількох серверах, що обмежує швидкість роботи по мірі зростання загального об'єму даних;
- відносне ускладнення структури через наявність великої кількості зв'язків між табличними даними.

Через постійне надмірне зростання об'ємів інформації, що зберігається і обробляється в реляційних базах даних, виникають нові унікальні проблеми, які можуть знижувати продуктивність обслуговування клієнтів і поставити під загрозу безпеку. Для усунення таких проблем мають бути знайдені ефективні рішення, що забезпечують зменшення ймовірності конфлікту блокувань та оптимізацію ефективності запитів, а також мають бути знайдені ефективні стратегії підтримки працездатності баз даних. Як показано в роботі [3], для вирішення проблем мають бути проведені дослідження в наступних напрямах.

Першою і головною проблемою є проблема конфлікту блокувань, коли одночасний доступ до спільних ресурсів зменшує продуктивність роботи. Така проблема виникає під час пікового навантаження, що приводить до перекриття транзакційних запитів. Її рішенням може бути використання інструментів моніторингу та виявлення процесів що сприяють блокуванню, а також оптимізація запитів, що передаються в середовищах з високим трафіком.

Наступною проблемою є надмірне навантаження, яке апаратні обмеження можуть створювати для великих реляційних баз даних. Зі збільшенням розміру наборів даних вони вимагають більше пам'яті та обчислювальних ресурсів, що може призводити до зниження продуктивності та потенційних збоїв системи, якщо їх не контролювати належним чином. Ця проблема може бути усунена регульованою оцінкою та оновленням обладнання відповідно до зростання

обсягів даних. Більше того, необхідне попереднє планування інвестицій у розвиток апаратної інфраструктури.

Ще одним завданням досліджень в напрямі оптимізації роботи реляційних баз даних є впровадження надійності стратегії резервного копіювання, яке забезпечує цілісність даних. Такі стратегії діють як запобіжні заходи від випадкового видалення, пошкодження або інших непередбачених подій. Їх використання, включаючи регулярні оновлення та наявність резервних сховищ, гарантує, що дані залишаться цілісними та їх можна буде відновити у разі виникнення аварійних ситуацій.

Удосконалення управління обробленням даних вимагає, також, оптимізації продуктивності повільних запитів, які виникають в результаті їх недосконалого програмування, відсутністю індексації або проблемами з проектуванням бази даних. Регулярна перевірка їх продуктивності та внесення необхідних змін може покращити час відгуку.

Зниження продуктивності запитів вимагає їх оптимізації. Управління великою реляційною базою даних часто пояснюється проблемами, пов'язаними з повільною продуктивністю запитів. Це може бути спричинено неефективністю програмування запитів, відсутністю індексації або проблемами з проектуванням баз даних. Регулярна перевірка продуктивності запитів та внесення необхідних оптимізації може значно покращити час відгуку.

Великі реляційні бази даних часто містять конфіденційну та цінну інформацію, що робить їх основними цілями для кібератак. Їх захист вимагає таких заходів як, шифрування, контроль доступу та регулярні оцінки захищеності. Оскільки кібератаки використовують слабкі, з точки зору захисту, місця, своєчасне усунення вразливостей допомагає захиститися від порушень конфіденційності та втрати даних. Впровадження комплексної стратегії захисту, включаючи сучасне програмне та апаратне забезпечення, зменшує рівень потенційних загроз.

Головною умовою підтримки безперебійної роботи великих реляційних баз даних є постійний моніторинг обміну даними, мета якого полягає у ранньому виявленні будь-яких проблем. Такий моніторинг показників продуктивності і виявлення помилок та аналіз моделей використання дозволяє своєчасно корегувати процеси обміну даними в режимі реального часу та мінімізувати час простою.

Із сказаного витікає, що сучасні реляційні бази даних, попри довгий термін розвитку і надзвичайну популярність, вимагають подальшого удосконалення. Причинами такого стану є постійне і невпинне зростання обсягів даних, що використовуються великою кількістю користувачів і, відповідно, зростання складності процесів, пов'язаних з узгодженням дій між суб'єктами інформаційних процесів. Забезпечення постійного моніторингу подій в мережній та програмно-апаратній частині баз даних само по собі вимагає значних обчислювальних ресурсів та фінансових витрат. Тому пошук рішень, що дозволяють оптимізувати роботу сучасних реляційних баз даних є нагальною задачею.

Список використаних джерел

1. Pradeep Bhosale. Data Consistency Models in Distributed Systems: CAP Theorem Revisited. *International Journal on Science and Technology* (IJSAT), Volume 14, Issue 3, July-September 2023. URL: https://www.ijsat.org/papers/2023/3/1408.pdf.
2. Ginard S. Guaki. A Comparative Analysis of Relational, NoSQL and NewSQL Database System. July 2024. DOI:10.13140/RG.2.2.27904.03849. URL: https://www.researchgate.net/publication/382457773_A_Comparative_Analysis_of_Relational_NoSQL_and_NewSQL_Database_System.
3. Ted Dunning Ellen Friedman. Time Series Databases: New Ways to Store and Access Data. O'Reilly Report, 2021. URL : https://vdoc.pub/documents/time-series-databases-new-ways-to-store-and-access-data-3hrf3sgae1og.

Ключові слова: реляційні бази даних, бази даних типу NoSQL, реляційна модель, архітектури бази даних

Key words: relational databases, NoSQL databases, relational model, database architectures

БІБЛІОТЕКИ ОБ'ЄКТНО-РЕЛЯЦІЙНОГО ВІДОБРАЖЕННЯ: ПЕРЕВАГИ, ОБМЕЖЕННЯ ТА ЗАСТОСУВАННЯ

Лобода Юлія

кандидат педагогічних наук, доцент,
доцент кафедри програмної інженерії
Національного університету «Одеська юридична академія»

Взаємодія між прикладним програмним кодом і реляційною базою даних є одним із базових завдань сучасної веброзробки. Реляційні СКБД зберігають дані у вигляді таблиць, рядків, первинних і зовнішніх ключів, тоді як програмний код зазвичай працює з об'єктами, класами або структурами даних. Через структурну невідповідність між двома моделями виникає потреба у перетворенні даних між реляційним і об'єктним рівнями.

Традиційний підхід передбачає написання SQL-запитів безпосередньо в коді програми. Він забезпечує повний контроль над роботою з базою даних, однак у великих проєктах призводить до дублювання однотипного коду, ускладнює супровід і підвищує ризик помилок. Особливо помітно недоліки такого підходу проявляються під час реалізації типових CRUD-операцій та роботи зі зв'язками між таблицями.

Одним із поширених способів розв'язання зазначеної проблеми є використання Object-Relational Mapping (ORM) – проміжного шару між прикладною логікою та реляційною базою даних. Розробник описує структуру даних у вигляді моделей, а ORM формує SQL-запити, виконує операції з базою даних і перетворює результати у зручні для програми об'єкти [1]. Водночас застосування ORM не скасовує потреби у знанні SQL і реляційної моделі. Ефективна робота з

бібліотекою потребує розуміння того, які запити вона генерує, як функціонують зв'язки між таблицями, індекси, транзакції та обмеження цілісності.

Sequelize є ORM-бібліотекою для Node.js, яка підтримує роботу з MySQL, PostgreSQL, SQLite та іншими реляційними СКБД. У разі використання MySQL модель Sequelize виступає програмним представленням таблиці бази даних. Кожен екземпляр моделі відповідає одному рядку таблиці, а методи моделі забезпечують виконання типових операцій з даними.

Основним елементом Sequelize є модель, у якій описуються поля таблиці, типи даних, обмеження, правила валідації та додаткові параметри. Для виконання CRUD-операцій використовуються методи create, findAll, findByPk, update, destroy та інші. Опис зв'язків між моделями реалізується через методи hasOne, hasMany, belongsTo та belongsToMany. Вони дозволяють відобразити реляційні відношення один-до-одного, один-до-багатьох і багато-до-багатьох у програмному коді. У базі даних такі зв’язки реалізуються через зовнішні ключі та проміжні таблиці.

Sequelize підтримує «eager loading» через параметр include. Завдяки цьому пов’язані дані завантажуються разом з основною сутністю в межах одного запиту із використанням JOIN. До функціональних можливостей Sequelize належать міграції схеми, транзакції та валідація даних. Міграції дозволяють фіксувати зміни структури бази даних у вигляді окремих файлів і застосовувати їх послідовно. Транзакції забезпечують виконання групи операцій як єдиного цілого, що важливо для збереження узгодженості даних у MySQL [2]. Дослідження підтверджують, що Sequelize ефективно працює в сценаріях з одиночними запитами та низьким рівнем конкурентності, тоді як під паралельним навантаженням альтернативні бібліотеки можуть демонструвати вищу продуктивність [2].

Основною перевагою ORM є скорочення обсягу шаблонного коду. Типові операції з базою даних виконуються через методи моделей, тому розробнику не потрібно вручну писати SQL-запити для кожної стандартної дії. Такий підхід прискорює розробку та робить код більш структурованим.

Другою перевагою є підвищення підтримуваності програмного забезпечення. Робота з даними описується через сутності предметної області, наприклад User, Order, Ticket, а не через набір SQL-рядків, розміщених у різних частинах програми. Подібний підхід спрощує зміну логіки доступу до даних і полегшує командну розробку.

Третьою перевагою є зменшення ризику SQL-ін’єкцій. ORM-бібліотеки зазвичай використовують параметризовані запити та автоматичне екранування значень [3]. Застосування ORM не скасовує потреби у перевірці вхідних даних, але знижує ризик помилок, пов’язаних із ручним формуванням SQL-запитів.

Разом із перевагами ORM має суттєві обмеження. Додатковий шар абстракції може створювати накладні витрати на виконання запитів, використання пам’яті та процесорного часу. Дослідження продуктивності ORM-фреймворків

показують, що зручність розробки може супроводжуватися зниженням ефективності порівняно з оптимізованими SQL-запитами [4].

Однією з найпоширеніших практичних проблем ORM є N+1-проблема. Вона виникає тоді, коли спочатку виконується один запит для отримання основного набору даних, а потім для кожного запису окремо виконується додатковий запит до пов’язаної таблиці. У невеликих наборах даних така проблема може бути непомітною, однак зі збільшенням кількості записів вона різко підвищує навантаження на базу даних.

Дослідження показують, що усунення N+1-проблеми може суттєво прискорити роботу застосунків – до 7,67 разу, а на великих базах даних – до 38,58 разу [5]. У Sequelize для зменшення цього ризику доцільно використовувати «eager loading», аналізувати сформовані SQL-запити та профілювати роботу застосунку.

Ще одним обмеженням ORM є складність реалізації нетривіальних SQL-запитів. Агрегації, підзапити, складні умови фільтрації або аналітичні вибірки іноді важко виразити засобами бібліотеки ORM. У таких випадках Sequelize дозволяє використовувати raw SQL-запити через метод query, що дає змогу поєднувати переваги ORM для типових операцій і гнучкість SQL для складних сценаріїв.

ORM є ефективним засобом абстракції над реляційними базами даних, який спрощує взаємодію між прикладним кодом і СКБД, зменшує кількість шаблонного SQL-коду та підвищує підтримуваність програмного забезпечення. Sequelize є поширеним ORM-інструментом для Node.js, який підтримує опис моделей, зв'язків, транзакцій, міграцій і валідації даних, що робить його доцільним вибором для вебзастосунків середньої складності.

Разом із тим ORM не є повноцінною заміною SQL. Ефективне використання Sequelize потребує розуміння реляційної моделі, аналізу згенерованих запитів і контролю продуктивності. Для складних аналітичних або високонавантажених операцій доцільно поєднувати ORM із прямими SQL-запитами.

Список використаних джерел

1. Majerik F., Borkovcova M. Design of Data Access Architecture Using ORM Framework. *2023 34th Conference of Open Innovations Association (FRUCT)*. 2023, P. 93-99. DOI: https://doi.org/10.23919/FRUCT60429.2023.10328151.
2. Zhadko-Bazilevych S. Analysis of ORM framework approaches for Node.js. *Journal of Computer Sciences Institute*. Vol. 37, P. 426–430. https://doi.org/10.35784/jcsi.7951
3. Bonvoisin A., Quinton C., Rouvoy R. Understanding the Performance-Energy Tradeoffs of Object-Relational Mapping Frameworks. *2024 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 2024, P. 626–636. DOI: https://doi.org/10.1109/saner60148.2024.00069
4. Turcotte A., Aldrich M., Tip F. reformulator: Automated Refactoring of the N+1 Problem in Database-Backed Applications. *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*. 2022, Vol. 84, P. 1-11. DOI: https://doi.org/10.1145/3551349.3556911

Ключові слова: ORM, об'єктно-реляційне відображення, Sequelize, MySQL, Node.js, абстракція даних

Keywords: ORM, object-relational mapping, Sequelize, MySQL, Node.js, data abstraction

ВИКОРИСТАННЯ REDIS ДЛЯ КЕШУВАННЯ АНАЛІТИЧНИХ ЗАПИТІВ У СИСТЕМАХ УПРАВЛІННЯ ЗВЕРНЕННЯМИ

Лобода Юлія

кандидат педагогічних наук, доцент,
доцент кафедри програмної інженерії
Національного університету «Одеська юридична академія»

Саблуков Костянтин

студент 4-го курсу факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Системи управління зверненнями технічної підтримки накопичують значні обсяги структурованих даних про заявки, користувачів, категорії звернень, статуси виконання та дії працівників служби підтримки. Аналіз таких даних використовується для оцінювання навантаження на фахівців, контролю швидкості опрацювання заявок, виявлення проблемних категорій звернень і формування управлінських звітів.

Аналітичні запити в подібних системах зазвичай передбачають фільтрацію, групування, агрегацію та об'єднання кількох таблиць реляційної бази даних. На відміну від операцій перегляду або оновлення окремих записів, такі запити мають вищу обчислювальну складність і можуть створювати додаткове навантаження на сервер бази даних. Особливо відчутною ця проблема стає при частому формуванні аналітичних звітів і одночасному зверненні кількох користувачів до статистики системи.

Одним із підходів до зменшення навантаження на реляційну базу даних є кешування результатів аналітичних запитів у сховищі типу «ключ–значення». Redis може використовуватися як проміжний шар між серверною частиною вебзастосунку та базою даних PostgreSQL, оскільки зберігає дані в оперативній пам'яті, підтримує швидкий доступ за ключем і різні структури даних для зберігання результатів обчислень [1, 2].

Для систем управління зверненнями придатною є стратегія «cache-aside». Суть цієї стратегії полягає в тому, що серверна частина спочатку перевіряє наявність результату аналітичного запиту в Redis. Якщо дані знайдено, відповідь повертається клієнту без звернення до основної бази даних. Такий підхід дає змогу кешувати не всі дані системи, а лише ті результати, які реально запитуються користувачами. Якщо запис у кеші відсутній, сервер виконує запит до бази даних

PostgreSQL, зберігає отриманий результат у Redis із визначеним часом життя та повертає його користувачу.

Дослідження показують, що застосування стратегії «cache-aside» може знизити середній час відповіді системи з 1146 до 323 мс, а в мікросервісних системах – забезпечити рівень cache hit до 92 %. Загалом кешування аналітичних запитів засобами Redis може скоротити час відповіді у 3–4 рази та зменшити навантаження на реляційну базу даних [2, 3].

Важливим елементом такої організації є механізм Time to Live (TTL). TTL визначає час життя запису в кеші. Після завершення цього часу Redis автоматично видаляє відповідний ключ, тому застарілі дані не зберігаються безстроково. Наприклад, для сторінки із загальною статистикою звернень може бути встановлено TTL 60–300 секунд, а для менш динамічних тижневих або місячних звітів – довший період зберігання. Отже, TTL обмежує час актуальності кешованого результату та зменшує ризик показу застарілих аналітичних даних.

Ключовою проблемою при кешуванні аналітичних запитів є забезпечення актуальності даних. У системах управління зверненнями аналітичні показники змінюються після створення заявки, зміни її статусу, призначення відповідального працівника, додавання коментаря або закриття звернення. Тому поряд із TTL варто використовувати інвалідацію кешу – примусове видалення кешованих записів після зміни даних, що впливають на аналітичні результати [2]. Якщо, наприклад, змінено статус заявки, серверна частина може видалити ключі, пов’язані з аналітикою за статусами, навантаженням працівників і загальною статистикою системи [4].

Структура ключів має формуватися з урахуванням типу аналітичного запиту та параметрів фільтрації. Для системи управління зверненнями можуть використовуватися ключі виду analytics:tickets:status, analytics:tickets:category, analytics:agents:workload, analytics:dashboard. Якщо аналітичний запит має параметри, їх також варто включати до ключа, наприклад період, категорію або ідентифікатор відповідального працівника. Така структура полегшує оновлення та видалення кешованих результатів.

Комбінація TTL та інвалідації кешу дає змогу збалансувати продуктивність і актуальність даних. TTL забезпечує автоматичне очищення кешу після завершення заданого часу, а інвалідація дозволяє оновити дані раніше, якщо в системі відбулася важлива зміна. Для аналітичних сторінок, де допустима невелика затримка оновлення показників, такий підхід є практичним і не потребує значного ускладнення архітектури серверної частини.

Практичні дослідження показують, що застосування Redis для кешування результатів запитів до реляційних баз даних може скоротити час відповіді вебзастосунку та зменшити кількість звернень до основної бази даних [3–5]. Для систем управління зверненнями це особливо важливо в аналітичних модулях, де одні й ті самі агреговані показники можуть багаторазово запитуватися різними користувачами.

Отже, використання Redis для кешування аналітичних запитів у системах управління зверненнями є обґрунтованим архітектурним рішенням. Запропонований підхід передбачає застосування стратегії cache-aside, формування структурованих ключів для аналітичних маршрутів API, встановлення TTL та інвалідацію кешу після змін у заявках. Така організація зменшує навантаження на реляційну базу даних, прискорює роботу аналітичного модуля та підтримує прийнятний рівень актуальності показників.

Список використаних джерел

1. Redis Documentation. URL: https://redis.io/docs
2. Kaur G., Kaur J. In-Memory Data processing using Redis Database. *International Journal of Computer Applications*. 2018, Vol. 180, p. 26–31. DOI: https://doi.org/10.5120/ijca2018916589
3. Zulfa M., Fadli A., Wardhana A. Application caching strategy based on in-memory using Redis server to accelerate relational data access. *Jurnal Teknologi dan Sistem Komputer*. 2020, Vol. 8, p. 157–163. DOI: https://doi.org/10.14710/jtsiskom.8.2.2020.157-163
4. Privalov M., Stupina M. Improving web-oriented information systems efficiency using Redis caching mechanisms. *Indonesian Journal of Electrical Engineering and Computer Science*. 2024. DOI: https://doi.org/10.11591/ijeecs.v33.i3.pp1667-1675
5. Prasetyo D., Fatoni F., Nasir M., Effendy I. Implementasi Redis Untuk Mempercepat Pengambilan Data. *Explore: Jurnal Sistem Informasi dan Telematika*. 2024. DOI: https://doi.org/10.36448/jsit.v15i2.3948
6. Aglibar K., Rodelas N. Impact of Critical and Auto Ticket: Analysis for Management and Workers Productivity in using a Ticketing System. *ArXiv*. 2022. DOI: https://doi.org/10.48550/arxiv.2203.03709

Ключові слова: Redis, кешування, аналітичні запити, TTL, інвалідація кешу

Keywords: Redis, caching, analytical queries, TTL, cache invalidation

СУЧАСНІ АРХІТЕКТУРНІ ПІДХОДИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБПЛАТФОРМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

Максименко Олександр

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Сучасний розвиток IT-технологій диктує нові вимоги до проєктування платформ електронної комерції. Традиційні монолітні рішення та «коробкові» CMS не завжди відповідають вимогам сучасних вебплатформ електронної комерції, особливо коли йдеться про високу інтерактивність інтерфейсу, гнучке управління даними та масштабування системи [1]. Необхідність повного перезавантаження сторінок при кожній взаємодії із сервером призводить до навантаження

на мережу та погіршення користувацького досвіду (UX), що критично для утримання клієнтів.

Для вирішення цих проблем доцільно використовувати архітектуру односторінкових застосунків (SPA). Реалізація клієнтської частини на базі React дозволяє перенести обчислювальну логіку на сторону браузера, а концепція Virtual DOM забезпечує точкове оновлення компонентів без повторного рендерингу всієї сторінки [2]. Це створює інтерфейси з миттєвим відгуком, що за швидкістю імітують нативні застосунки.

Ефективна робота SPA-архітектури вимагає надійного та масштабованого бекенду. Оптимальним вибором для систем електронної комерції є середовище виконання Node.js у поєднанні з мінімалістичним фреймворком Express.js [3]. Асинхронна, подієво-орієнтована модель з неблокуючим вводом-виводом (non-blocking I/O) дозволяє серверу паралельно обробляти тисячі одночасних з'єднань, що є життєво необхідним під час пікових навантажень (наприклад, у періоди акцій чи розпродажів). Додатковою перевагою є використання мови програмування TypeScript як на клієнті, так і на сервері, що реалізує концепцію «JavaScript Everywhere». Статична типізація суттєво мінімізує кількість помилок ще на етапі компіляції, спрощує рефакторинг великих кодових баз та гарантує строгу цілісність даних при їх передачі через RESTful API [4].

Критично важливим аспектом проєктування є вибір системи управління базами даних. Для платформ, що реалізують товари зі складною, багаторівневою або динамічною структурою атрибутів (наприклад, товари з різними варіантами фасування, нестандартними характеристиками та гнучким ціноутворенням), традиційні реляційні бази даних (SQL) виявляються недостатньо гнучкими. Застосування документоорієнтованих NoSQL рішень, таких як MongoDB, дозволяє зберігати дані у форматі BSON (бінарному представленні JSON) [5]. Це дає змогу природно та ефективно описувати ієрархічні структури товарів без необхідності створення складних зв'язків (JOIN) між десятками таблиць, що значно пришвидшує виконання запитів на читання та запис у базі даних.

Управління станом у складних клієнтських застосунках також вимагає сучасних інженерних підходів. Розділення стану на клієнтський (наприклад, тимчасові дані кошика чи UI-елементів) та серверний (кешовані дані каталогу товарів) є галузевим стандартом сучасної розробки. Використання інструментів глобального управління станом (на кшталт Redux Toolkit) забезпечує передбачувану мутацію клієнтського стану, тоді як інструменти смарт-кешування (наприклад, React Query) беруть на себе завдання автоматичного фонового оновлення серверних даних, пагінації та синхронізації з бекендом. Такий підхід зменшує кількість надлишкових HTTP-запитів та зменшує навантаження на серверну інфраструктуру.

Питання кібербезпеки є найвищим пріоритетом для платформ електронної комерції, оскільки вони обробляють конфіденційні дані користувачів. Реалізація механізмів автентифікації на базі JSON Web Tokens (JWT) дозволяє відмовитися від зберігання сесій на сервері, що робить архітектуру stateless (без стану) та

значно полегшує горизонтальне масштабування серверних потужностей [7]. Додатковий рівень безпеки забезпечується інтеграцією протоколів відкритої авторизації (наприклад, OAuth 2.0), надійним криптографічним хешуванням паролів за допомогою алгоритму bcrypt та обов'язковим впровадженням проміжного програмного забезпечення (middleware) для захисту від brute-force атак, DDoS-атак та міжсайтового скриптингу (XSS).

Отже, проєктування сучасних систем електронної комерції потребує комплексного інженерного підходу, що охоплює вибір архітектури клієнтської частини, серверної логіки, бази даних, засобів управління станом і механізмів безпеки. Технологічний стек MERN у поєднанні з TypeScript, SPA-архітектурою, MongoDB та сучасними інструментами управління станом є доцільним рішенням для розробки гнучких вебплатформ із динамічним каталогом товарів. Такий підхід дозволяє забезпечити зручну взаємодію користувача з інтерфейсом, ефективну обробку даних, масштабованість серверної частини та належний рівень захисту користувацької інформації.

Список використаних джерел

1. Laudon K. C., Traver C. G. E-commerce: business, technology, society. Pearson, 2021. 900 p. URL: https://surl.li/jjcquc
2. React Documentation: Building User Interfaces. URL: https://react.dev/
3. Node.js Official Documentation. URL: https://nodejs.org/docs/latest/api/
4. TypeScript Handbook. URL: https://surl.li/abryja
5. MongoDB.URL: https://www.mongodb.com/docs/development/
6. JSON Web Token (JWT). URL: https://datatracker.ietf.org/doc/html/rfc7519

Ключові слова: вебзастосунок, електронна комерція, SPA-архітектура, NoSQL, MERN-стек, TypeScript, бази даних

Keywords: web application, e-commerce, SPA architecture, NoSQL, MERN stack, TypeScript, databases

Науковий керівник: доцент Лобода Ю.Г.

АРХІТЕКТУРА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ВІДСТЕЖЕННЯ ГЕОЛОКАЦІЇ З ВИКОРИСТАННЯМ ПРОТОКОЛІВ MQTT ТА WEBSOCKET

Гура Володимир

кандидат технічних наук,
доцент кафедри штучного інтелекту та математичного моделювання
Національного університету «Одеська юридична академія»

Папіж Віолета

студентка 4-го курсу факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Стрімкий розвиток ринку телеметричних послуг формує стійкий попит на ефективні програмні засоби моніторингу геолокації транспортних засобів. Комерційні рішення (Wialon, GeoTab, Bitrek) обмежені закритою архітектурою та високою вартістю, тоді як відкриті альтернативи (Traccar) мають слабку підтримку мобільних клієнтів і не використовують сучасних механізмів передачі даних. Актуальною є розробка універсальної масштабованої системи з надійною доставкою координат за умов нестабільних мобільних мереж та інтерактивною візуалізацією у режимі реального часу.

Метою роботи є розробка архітектурного рішення та програмна реалізація системи відстеження геолокації транспортних засобів у реальному часі з використанням протоколів MQTT та WebSocket. Завдання дослідження охоплюють: обґрунтування комбінації протоколів передачі даних, проектування мікросервісної декомпозиції системи та підсистеми зберігання геопросторових і темпоральних даних, реалізацію клієнтських застосунків з інтерактивною картою, експериментальну перевірку прототипу за показниками продуктивності та стійкості.

Розроблене рішення запропонованої архітектури (рис. 1) базується на мікросервісній архітектурі з подвійним каналом передачі даних. GPS-трекери та мобільний застосунок водія публікують координати у брокер Eclipse Mosquitto за протоколом MQTT, що оптимально для нестабільних мобільних мереж [1–3]. Шлюз WebSocket забезпечує низхідну доставку оновлень інтерфейсам користувачів. Обробний конвеєр об'єднує п'ять мікросервісів: Ingestion (прийом MQTT), Tracking (геозони), Route (історія маршрутів), Notification та Auth [2,4]. Підсистема зберігання поєднує PostgreSQL з розширенням PostGIS для геопросторових запитів, TimescaleDB для часових рядів координат та Redis для кешування і Pub/Sub-синхронізації [5]. Клієнтська частина: веб-інтерфейс на React з рушієм Leaflet, крос-платформові мобільні застосунки на React Native [6].

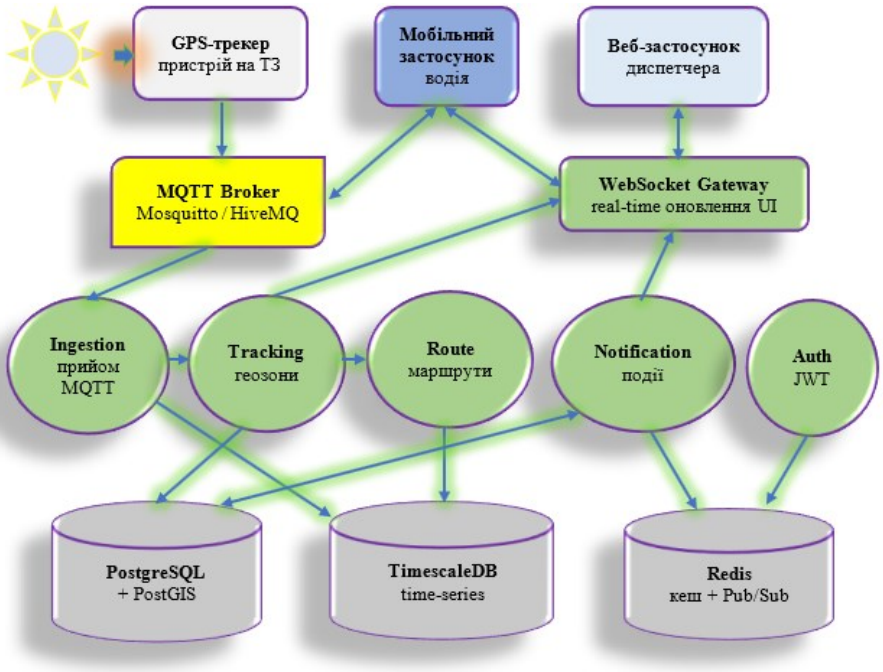


Рисунок 1 – Архітектура системи відстеження геолокації транспортних засобів

Експериментальне дослідження прототипу реалізовано засобами Node.js та FastAPI з контейнеризацією у Docker. Експериментальна перевірка підтвердила обслуговування 2358 одночасних GPS-пристроїв із затримкою доставки оновлень менше 143 мс; втрати координат за умов імітації мережі 3G/EDGE – менше 1% завдяки гарантіям QOS; точність геозонування на 4500 подій – понад 97% [5]. Порівняно з реалізацією на HTTP-опитуванні досягнуто зменшення мережевого трафіку у 7–9 разів та споживання заряду акумулятора втричі.

Розроблений прототип поєднує переваги мікросервісної декомпозиції, легковагового MQTT для висхідного каналу та WebSocket для оперативної доставки оновлень із спеціалізованими сховищами геопросторових даних. Експериментально підтверджено масштабованість і стійкість системи. Перспективи розвитку – інтеграція з моделями машинного навчання для прогнозування часу прибуття та виявлення аномалій водіння.

Список використаних джерел

1. Integrating Real-Time Wireless Sensor Networks into IoT Using MQTT-SN. Journal of Network and Systems Management (Springer). 2025. DOI: 10.1007/s10922-025-09916-1.
2. A Distributed Architecture for MQTT Messaging: The Case of TBMQ. Journal of Big Data (Springer). 2025. DOI: 10.1186/s40537-025-01271-x.
3. Comparison Between MQTT and WebSocket Protocols for IoT Applications. IEEE Xplore. URL: https://ieeexplore.ieee.org/document/8428348.
4. Tokmak A. V., Akbulut A., Catal C. Boosting the Visibility of Services in Microservice Architecture. Cluster Computing (Springer). 2024. Vol. 27. P. 3099–3111.

5. Vedantham A. K. Geofencing in IoT: Enhancing Location-Based Services. IJCET. 2024. Vol. 15, Iss. 6. P. 687–700.
6. Flutter vs React Native: Complete 2025 Framework Comparison Guide. Droids on Roids. 2025.

Ключові слова: геолокації, відстеження, Інтернет речей, MQTT, WebSocket, мікросервісна архітектура, геопросторова база даних

Keywords: geolocation, tracking, Internet of Things, MQTT, WebSocket, microservices architecture, geospatial database

ІНВЕНТАРИЗАЦІЯ ПРИСТРОЇВ У КОМП'ЮТЕРНИХ МЕРЕЖАХ

Гура Володимир

кандидат технічних наук,
доцент кафедри штучного інтелекту та математичного моделювання
Національного університету «Одеська юридична академія»

Гришин Богдан

студент 4-го курсу факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Стрімке зростання кількості мережевих пристроїв у корпоративних та IT-інфраструктурах формує нові виклики для адміністрування та кібербезпеки. Сучасні комп'ютерні мережі об'єднують серверне обладнання, робочі станції, активне мережеве обладнання, мобільні та IoT-пристрої, причому, за прогнозами аналітичних агентств, кількість підключених пристроїв стримко збільшується у глобальному масштабі.

Ручне ведення інвентаризації в таких умовах стає неефективним, схильним до помилок та неспроможним відстежувати динамічні зміни. Окрему загрозу становлять несанкціоновано підключені пристрої (Shadow IT), які формують додаткову поверхню атаки. Наявні комерційні рішення (Lansweeper, SolarWinds NPM, Device42) мають високу вартість і часто потребують встановлення агентів на цільових вузлах, що ускладнює їх застосування у гетерогенних середовищах. Відкриті альтернативи (Nmap, Open-AudIT, NetBox, GLPI з агентом FusionInventory) не використовують методи машинного навчання та потребують значного ручного налаштування правил класифікації. Відтак актуальною є задача розробки гібридного методу та програмних засобів автоматичного виявлення та інвентаризації пристроїв, який поєднує безагентне сканування мережі з інтелектуальною класифікацією на основі машинного навчання.

Метою дослідження є розробка методу та програмних засобів автоматичного виявлення та інвентаризації пристроїв у TCP/IP-мережах з інтелектуальною класифікацією на основі методів машинного навчання. Для досягнення поставленої мети необхідно: проаналізувати існуючі підходи й інструменти виявлення мережевих пристроїв; розробити гібридний метод, що поєднує активне

сканування на основі протоколів ARP та ICMP з пасивним опитуванням через SNMP; сформувати множину інформативних ознак для побудови ML-класифікатора типів пристроїв; розробити архітектуру програмної системи та реалізувати її прототип; провести експериментальну перевірку точності класифікації на реальному мережевому середовищі.

Класичні методи виявлення пристроїв ґрунтуються на активному скануванні (Nmap, ZMap) та використовують особливості реалізації TCP/IP-стека для визначення типу операційної системи. Зокрема, Nmap аналізує понад 16 параметрів відгуків на TCP-, UDP- та ICMP-зонди для побудови фінгерпринту [1]. Однак такі методи мають обмежену точність при ідентифікації специфічних типів пристроїв (IoT, IP-камери, мережеві принтери) та повністю ігнорують сучасні алгоритми класифікації.

У сучасних дослідженнях акцент змістився на використання методів машинного навчання. Sivanathan A. та співавтори [1] запропонували мультиетапний ML-класифікатор для 28 типів IoT-пристроїв на основі статистичних характеристик трафіку (активність, набір портів, сигнальні шаблони), який досягає точності понад 99%. Chowdhury R. R. та інші [2] продемонстрували, що класифікатор Random Forest на основі чотирьох статистичних ознак забезпечує F1-score 99.81% на публічному датасеті UNSW. Hamad S. A. та співавтори [3] використали потокові ознаки мережевого трафіку та supervised ML для ідентифікації типів пристроїв. У роботі [4-6] показано ефективність гібридного підходу, де попереднє пасивне сканування дозволяє визначити топологію мережі, після чого активне сканування деталізує характеристики кожного вузла.

Зведений аналіз показує, що жодне з розглянутих рішень не поєднує одночасно: безагентне виявлення пристроїв, опитування через стандартні мережеві протоколи (SNMP), інтелектуальну класифікацію на основі ML та універсальність застосування у TCP/IP-мережах довільного типу. Саме це визначає наукову новизну запропонованого методу.

Запропонований метод складається з чотирьох послідовних етапів. На першому етапі виконується ARP-broadcast сканування на канальному рівні (L2), що дозволяє швидко виявити всі активні пристрої у локальній підмережі за MAC-адресами та визначити виробників обладнання за полем OUI [2,7]. На другому етапі здійснюється ICMP-ping sweep для підтвердження доступності вузлів та визначення базових параметрів (значення TTL, час відгуку), які несуть інформацію про тип операційної системи. На третьому етапі виконується SNMP-опитування керованого мережевого обладнання: зчитуються атрибути sysDescr, sysObjectID, ifTable, ipNetToMediaTable, а також ARP-таблиці комутаторів і маршрутизаторів, що дозволяє виявити пристрої, недоступні для прямого сканування. На четвертому етапі проводиться цілеспрямоване TCP-сканування портів для отримання банерів сервісів і визначення набору відкритих служб.

На основі зібраних даних формується вектор ознак для кожного виявленого пристрою, що включає: ідентифікатор виробника за MAC OUI, кількість та

конкретний набір відкритих портів, значення TTL, розмір TCP-вікна, ключові слова з SNMP sysDescr та HTTP-банерів, статистику часу відгуку. Класифікація виконується алгоритмом Random Forest, обраним на основі результатів робіт [2, 5], в яких доведено, що він забезпечує найкращий баланс між точністю та обчислювальною ефективністю порівняно з XGBoost, LightGBM та багатошаровими перцептронами [8]. Виокремлено вісім класів пристроїв: сервер, робоча станція, маршрутизатор, комутатор, мережевий принтер, IP-камера, IoT-пристрій, мобільний пристрій.

Структурно-функціональна схема системи побудована за 5-ти рівневим принципом (рис. 1).

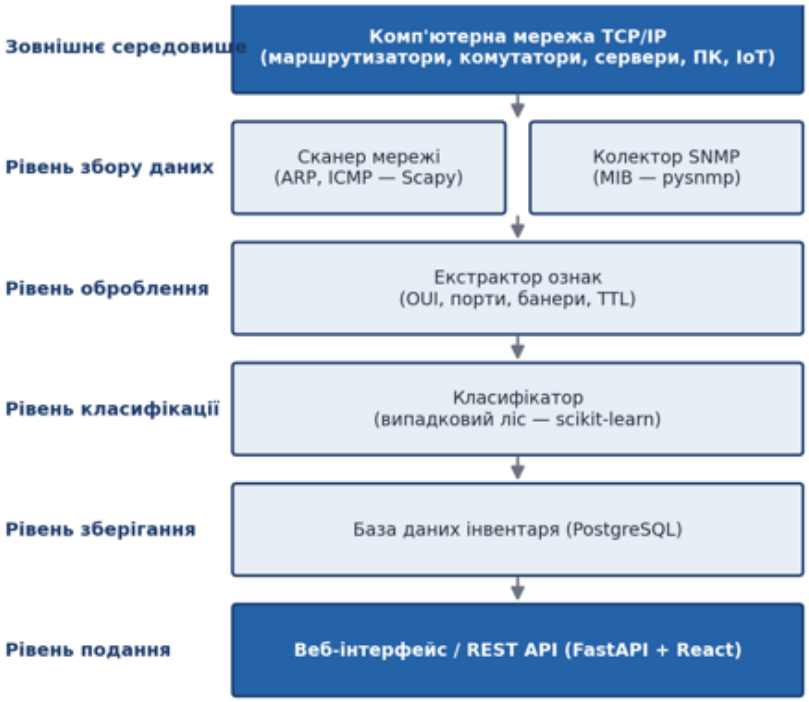


Рисунок 1 – Структурно-функціональна схема системи автоматичного виявлення та інвентаризації пристроїв

Зовнішнім середовищем виступає TCP/IP-мережа, яка містить досліджувані пристрої. Перший внутрішній рівень збору даних – складається з двох паралельних модулів: сканера мережі (виконує ARP- та ICMP-сканування з використанням бібліотеки Scapy) та SNMP-колектора (опитує MIB-таблиці засобами pysnmp). Другий рівень обробки – представлений модулем екстракції ознак, який нормалізує дані з обох джерел і формує вектори ознак. Третій рівень класифікації містить модуль ML-класифікатора на основі Random Forest, реалізованого засобами бібліотеки scikit-learn. Четвертий рівень зберігання та представлення – включає базу даних інвентаря (PostgreSQL) та веб-інтерфейс із REST API (FastAPI + React), через який адміністратор переглядає результати інвентаризації та ініціює нові цикли сканування.

Прототип системи реалізовано на Python із використанням бібліотек Scapy, pysnmp, scikit-learn, а також СКБД PostgreSQL для зберігання результатів інвентаризації. Експериментальне тестування проводилося у мережевому стенді з пристроями восьми вищезазначених класів. Середній час повного сканування підмережі /24 (256 IP-адрес) склав від 38 до 62 секунд залежно від кількості активних вузлів та налаштувань паралельності SNMP-опитування. Точність класифікації за метрикою macro F1-score становила 0.94, що на 12–15% перевищує rule-based методи (зокрема, нативне визначення типу пристрою у Nmap). Порівняння з Nmap показало переваги запропонованого методу у визначенні нестандартних IoT-пристроїв та мережевих принтерів, які класичним фінгерпринтингом часто ідентифікуються як generic-вузли.

Запропонований метод поєднує переваги активного та пасивного сканування мережі з інтелектуальною класифікацією на основі алгоритмів МН, що дозволяє автоматизовано формувати інвентаризацію мережевих пристроїв без використання агентів на цільових вузлах. Розроблена програмна система є масштабованою та універсальною для TCP/IP-мереж довільного типу та конфігурації. Експериментально підтверджено високу точність класифікації та прийнятну продуктивність сканування. Подальші напрями досліджень включають розширення множини класів пристроїв, інтеграцію з системами SIEM для автоматичного реагування на появу несанкціонованих вузлів, а також підвищення стійкості методу до спуфінгу MAC-адрес шляхом використання додаткових поведінкових ознак.

Список використаних джерел

1. Sivanathan A., Gharakheili H. H., Loi F., Radford A., Wijenayake C., Vishwanath A., Sivaraman V. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. IEEE Transactions on Mobile Computing. 2019. Vol. 18, No. 8. P. 1745–1759. DOI: 10.1109/TMC.2018.2866249.
2. Chowdhury R. R., Idris A. C., Abas P. E. Identifying SH-IoT Devices from Network Traffic Characteristics Using Random Forest Classifier. Wireless Networks. Springer, 2023. DOI: 10.1007/s11276-023-03478-3.
3. Hamad S. A., Zhang W. E., Sheng Q. Z., Nepal S. IoT Device Identification via Network-Flow Based Fingerprinting and Learning. Proceedings of 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2019. DOI: 10.1109/TrustCom/BigDataSE.2019.00023.
4. Cordeiro W., Wickboldt J. A., Granville L. Z. Automatically Determining a Network Reconnaissance Scope Using Passive Scanning Techniques. arXiv preprint arXiv:2106.14484. 2021.
5. Top-K Feature Selection for IoT Intrusion Detection: Contributions of XGBoost, LightGBM, and Random Forest. Future Internet (MDPI). 2025. Vol. 17, No. 11. Article 529. DOI: 10.3390/fi17110529.
6. Lyon G. F. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Sunnyvale: Insecure.Com LLC, 2009. 468 p. ISBN 978-0-9799587-1-7.

7. Plummer D. C. RFC 826: An Ethernet Address Resolution Protocol. IETF. November 1982. URL: https://www.rfc-editor.org/rfc/rfc826.
8. Методи функціонування пристроїв IoT з використанням машинного навчання. Системи управління, навігації та зв'язку. 2024. № 2. ISSN 2073-7394.

Ключові слова: автоматичне виявлення пристроїв, інвентаризація комп'ютерної мережі, протокол SNMP, протокол ARP, машинне навчання, класифікація мережевих пристроїв, Random Forest, безагентний моніторинг

Key words: automatic device discovery, network inventory, SNMP, ARP, machine learning, network device classification, Random Forest, agentless monitoring

Підсекція 3. Штучний інтелект, машинне навчання та комп'ютерний зір

МЕТОДИ МАШИННОГО НАВЧАННЯ В ОБРОБЦІ ЗОБРАЖЕНЬ

Логінова Наталія

кандидат педагогічних наук, доцент,
завідувачка кафедри програмної інженерії
Національного університету «Одеська юридична академія

Сучасний розвиток інформаційних технологій супроводжується стрімким поширенням систем штучного інтелекту (ШІ), здатних автоматично аналізувати великі обсяги даних та приймати рішення без безпосереднього втручання людини. Однією з найбільш перспективних галузей ШІ є комп'ютерний зір, основним завданням якого є отримання, аналіз та інтерпретація візуальної інформації у вигляді цифрових зображень і відео. Сьогодні системи комп'ютерного зору активно використовуються у медицині, транспорті, промисловості, системах безпеки, робототехніці та мобільних застосунках [1].

Важливу роль у розвитку комп'ютерного зору відіграє машинне навчання (МН), яке забезпечує можливість автоматичного виявлення закономірностей у даних та побудови моделей для класифікації, сегментації та розпізнавання об'єктів. На відміну від класичних методів обробки зображень, де правила та ознаки задавалися вручну, алгоритми МН здатні самостійно навчатися на прикладах, що значно підвищує точність та адаптивність систем [2].

Комп'ютерний зір і МН є взаємопов'язаними галузями. Якщо комп'ютерний зір забезпечує отримання та попередню обробку візуальної інформації, то МН виконує функцію її інтерпретації та аналізу. Завдяки цьому сучасні системи здатні не лише «бачити» об'єкти, а й розуміти їхній зміст, прогнозувати поведінку та приймати рішення на основі виявлених закономірностей.

Процес побудови системи комп'ютерного зору на основі МН включає кілька послідовних етапів (рис. 1).

Рисунок 1 – Процес побудови системи комп'ютерного зору на основі МН

Першим етапом є формування набору даних, який може містити фотографії, відео, медичні знімки або супутникові зображення. Якість і різноманітність даних безпосередньо впливають на ефективність навчання моделі.

Наступним етапом є виділення ознак. У традиційних системах комп'ютерного зору ознаки задавалися вручну та включали контури, текстури, кольорові характеристики або геометричні параметри. Сучасні алгоритми МН здатні автоматично визначати найбільш важливі характеристики зображення, що значно спрощує побудову моделей та підвищує їхню точність.

Після підготовки даних виконується навчання моделі. Для цього використовуються різні алгоритми МН, серед яких метод найближчих сусідів, метод опорних векторів, дерева рішень, логістична регресія та інші. Навчена модель повинна не лише запам'ятати навчальні приклади, а й узагальнювати отримані знання для роботи з новими даними [3].

Важливим етапом є оцінювання якості моделі. Найчастіше набір даних розділяють на навчальну та тестову вибірки за принципом train-test split. Навчальна вибірка використовується для навчання моделі, а тестова – для перевірки її здатності працювати з новими прикладами. Для більш об'єктивної оцінки застосовується крос-валідація, яка дозволяє оцінити стабільність результатів на різних частинах набору даних.

У комп'ютерному зорі використовуються різні типи МН. Серед яких найбільш поширеним є контрольоване навчання, або навчання з учителем. У цьому випадку модель навчається на попередньо розмічених даних, де кожному зображенню відповідає правильна мітка. Такий підхід широко застосовується для класифікації зображень, розпізнавання облич та виявлення об'єктів [4].

Одним із найпростіших алгоритмів контрольованого навчання є метод k-найближчих сусідів (k-NN). Його принцип роботи полягає у визначенні найближчих об'єктів до нового прикладу та віднесенні його до класу, який переважає серед сусідів. Метод є простим у реалізації та ефективним для невеликих наборів даних, однак потребує значних обчислювальних ресурсів при роботі з великими масивами інформації.

Іншим важливим алгоритмом є метод опорних векторів (SVM). Він базується на побудові оптимальної гіперплощини, яка максимально відокремлює класи даних. Перевагою SVM є висока точність класифікації та здатність працювати з високорозмірними просторами ознак. Цей алгоритм тривалий час залишався одним із основних методів класифікації зображень у системах комп'ютерного зору.

Для задач прогнозування числових характеристик використовують лінійну регресію, тоді як логістична регресія застосовується для задач бінарної класифікації. У комп'ютерному зорі ці методи використовуються для оцінки параметрів об'єктів, сегментації зображень та класифікації окремих елементів сцени.

Серед алгоритмів контрольованого навчання важливе місце займають дерева рішень, які формують ієрархічну структуру, де кожен вузол відповідає певній умові, а кінцеві листки визначають результат класифікації. Перевагою дерев рішень є простота інтерпретації та можливість пояснення прийнятих рішень.

Окрім контрольованого навчання, у комп'ютерному зорі активно застосовується неконтрольоване навчання, яке працює без попередньо розмічених даних. Основне завдання таких алгоритмів полягає у виявленні прихованих закономірностей та групуванні схожих об'єктів [5].

Одним із найпоширеніших методів неконтрольованого навчання є алгоритм k-means. Він використовується для кластеризації даних та сегментації зображень. Алгоритм поділяє зображення на k груп пікселів зі схожими характеристиками, що дозволяє спрощувати структуру зображення та виділяти основні області сцени. K-means широко застосовується для сегментації медичних знімків, стискання зображень та попередньої обробки даних.

Ще одним важливим методом є аналіз головних компонент (PCA), який використовується для зниження розмірності даних. Оскільки зображення можуть містити тисячі або навіть мільйони ознак, PCA дозволяє виділити найбільш інформативні компоненти та скоротити обсяг даних без суттєвої втрати інформації. Це значно підвищує швидкодію алгоритмів та зменшує ризик перенавчання моделей.

Для оцінювання якості моделей у комп'ютерному зорі використовують спеціальні метрики. Серед основних показників виділяють Accuracy, Precision, Recall та F1-score. Вони дозволяють оцінити точність класифікації та ефективність роботи моделі. Також широко застосовуються ROC-криві та показник AUC, які дають змогу оцінити якість класифікатора при різних порогах прийняття рішень [6].

Попри значні переваги, класичні методи МН мають певні обмеження. Їхня ефективність значною мірою залежить від якості ознак, які необхідно правильно виділити та підготувати. Крім того, традиційні алгоритми можуть втрачати ефективність при роботі з великими обсягами даних або складними структурами зображень. Саме тому в сучасних системах дедалі частіше використовуються методи глибокого навчання, здатні автоматично навчатися складним ієрархічним ознакам без ручного втручання.

Отже, МН є ключовою складовою сучасного комп'ютерного зору. Алгоритми МН забезпечують автоматичний аналіз зображень, виявлення закономірностей та прийняття рішень на основі візуальних даних. Використання методів контрольованого та неконтрольованого навчання дозволяє розв'язувати широкий спектр задач: від класифікації та сегментації зображень до робототехніки та автономної навігації. Подальший розвиток технологій ШІ сприятиме створенню

більш точних, швидких та інтелектуальних систем комп'ютерного зору, здатних ефективно працювати у реальних умовах.

Список використаних джерел

1. Nadeem Yousuf Khanday, Shabir Ahmad Sofi. Taxonomy, state-of-the-art, challenges and applications of visual understanding: A review, Computer Science Review, Volume 40, 2021, 100374, https://doi.org/10.1016/j.cosrev.2021.100374
2. Chen L., Li S., Bai Q., Yang J., Jiang S., Miao Y. Review of Image Classification Algorithms Based on Convolutional Neural Networks. Remote Sens. 2021, 13, 4712. https://doi.org/10.3390/rs13224712
3. Faisal, A.; Jhanjhi, N.Z.; Ashraf, H.; Ray, S.K.; Ashfaq, F. A Comprehensive Review of Machine Learning Models: Principles,Applications, and Optimal Model Selection. TechRxiv. 2025. DOI:10.36227/techrxiv.174285687.71966152/v1
4. Єрмоленко С. В. Машинне навчання: загальний огляд та застосування для розпізнавання текстових і візуальних об'єктів. Таврійський науковий вісник: Серія Технічні науки. 2025. № 3. С. 103-111. DOI:10.32782/tnv-tech.2025.3.11
5. Jing Wang, Filip Biljecki. Unsupervised machine learning in urban studies: A systematic review of applications. Cities, Volume 129, 2022, 103925,https://doi.org/10.1016/j.cities.2022.103925.
6. Rainio, O., Teuho, J. & Klén, R. Evaluation metrics and statistical tests for machine learning. Sci Rep 14, 6086 (2024). https://doi.org/10.1038/s41598-024-56706-x

Ключові слова: комп'ютерний зір, машинне навчання, обробка зображень, класифікація зображень, контрольоване навчання, неконтрольоване навчання, штучний інтелект

Key words: Computer vision, machine learning, image processing, image classification, supervised learning, unsupervised learning, artificial intelligence

СУЧАСНІ ПІДХОДИ ДО ВИЯВЛЕННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ У ЗАДАЧАХ КОМП'ЮТЕРНОГО ЗОРУ

Батін Олександр

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія

Комп'ютерний зір є одним із ключових напрямів розвитку сучасного штучного інтелекту, орієнтованим на автоматизовану обробку, аналіз та інтерпретацію візуальної інформації. Метою комп'ютерного зору є надання обчислювальним системам здатності сприймати та аналізувати цифрові зображення й відео подібно до людини [1]. Однім з головних завдань комп'ютерного зору є виявлення об'єктів на зображеннях. На відміну від задачі класифікації, де необхідно лише визначити клас об'єкта, детекція об'єктів передбачає також встановлення їх просторового розташування на зображенні.

Сучасні системи виявлення об'єктів активно застосовуються у різних областях знань. У сфері автономного водіння вони дозволяють транспортним

засобам розпізнавати пішоходів, дорожні знаки та інші автомобілі. У медичній галузі методи детекції використовуються для аналізу рентгенівських знімків, комп'ютерної томографії та виявлення патологій. У системах відеоспостереження технології детекції забезпечують автоматичне виявлення людей або підозрілих об'єктів у реальному часі.

Попри значний прогрес, задача виявлення об'єктів має низку складностей. Однією з основних проблем є варіативність зовнішнього вигляду об'єктів залежно від масштабу, освітлення, кута огляду чи часткового перекриття. Крім того, зображення можуть містити одночасно великі та дрібні об'єкти, що вимагає багатомасштабного аналізу. Додатковими викликами є наявність шуму, складного фону та необхідність забезпечення високої швидкості обробки у системах реального часу.

Першими підходами до виявлення об'єктів були класичні методи комп'ютерного зору, засновані на ручному конструюванні ознак. До таких методів належать алгоритми виявлення країв і контурів, методи виявлення особливих точок, перетворення Хафа, каскади Хаара та метод HOG/SVM.

Алгоритми виявлення країв дозволяють знаходити межі об'єктів шляхом аналізу змін інтенсивності пікселів. Одними з найпоширеніших є оператори Прюітта та Собеля, які використовують маски згортки для обчислення градієнтів зображення по горизонталі та вертикалі. Більш складним є алгоритм Кенні, який включає попереднє згладжування, обчислення градієнтів, немаксимальне пригнічення та порогову обробку. Такі методи дозволяють ефективно виділяти контури об'єктів, однак є чутливими до шуму та змін освітлення [2, 3].

Важливе місце серед класичних методів займають алгоритми виявлення особливих точок, зокрема SIFT та ORB. Вони дозволяють знаходити характерні локальні області зображення, стійкі до масштабування, обертання та змін освітлення. Отримані дескриптори використовуються для задач зіставлення зображень, побудови панорам та розпізнавання об'єктів [4].

Перетворення Хафа застосовується для виявлення геометричних примітивів, таких як прямі або кола. Метод базується на переході від простору зображення до параметричного простору, де накопичуються можливі параметри об'єктів. Незважаючи на високу стійкість до шуму, цей підхід потребує значних обчислювальних ресурсів [5].

Одним із перших ефективних алгоритмів детекції об'єктів у реальному часі став метод Віоли-Джонса, що використовує каскади Хаара. Його ключовими особливостями є інтегральне зображення та каскадна структура класифікаторів, що дозволяє швидко відкидати нерелевантні області. Алгоритм широко застосовувався для виявлення облич, однак поступово був витіснений більш точними нейромережевими методами [6].

Класичним методом детекції людей є HOG/SVM. HOG (Histogram of Oriented Gradients) формує описові характеристики на основі напрямків градієнтів, а класифікатор SVM визначає належність об'єкта до певного класу. Перевагою

підходу є відносно невисокі вимоги до обчислювальних ресурсів, однак його ефективність суттєво поступається сучасним нейромережевим моделям [7].

Справжній прорив у задачах комп'ютерного зору відбувся завдяки розвитку глибокого навчання та згорткових нейронних мереж (CNN). CNN є спеціалізованими нейронними мережами, оптимізованими для обробки зображень. Основою їх роботи є операція згортки, яка дозволяє автоматично виділяти просторові ознаки різного рівня складності. Нижні шари мережі виявляють прості структури, такі як краї та текстури, тоді як глибші шари формують високорівневі семантичні представлення.

Типова архітектура CNN складається зі згорткових шарів, функцій активації, шарів підвибірки (pooling) та повнозв'язних шарів. Важливою перевагою CNN є здатність автоматично навчати ознаки без необхідності ручного конструювання дескрипторів. Для підвищення ефективності сучасних моделей широко використовуються Feature Pyramid Network (FPN), які дозволяють формувати багатомасштабні ознакові представлення та покращують виявлення дрібних об'єктів [8].

Сучасні нейромережеві детектори об'єктів поділяються на двоетапні та одноетапні. Двоетапні детектори спочатку формують регіони-кандидати, а потім класифікують їх та уточнюють координати. Першою архітектурою цього типу стала R-CNN, яка використовувала Selective Search для генерації регіонів і CNN для виділення ознак. Недоліком підходу була надзвичайно низька швидкість роботи.

Подальшим розвитком стали Fast R-CNN та Faster R-CNN. Fast R-CNN виконувала згорткову обробку всього зображення лише один раз, що суттєво прискорило інференс. Faster R-CNN додатково замінила алгоритм Selective Search на Region Proposal Network (RPN), яка генерує регіони-кандидати безпосередньо нейронною мережею. Такі архітектури забезпечують високу точність, але залишаються відносно повільними для застосувань реального часу [9].

Одноетапні детектори виконують класифікацію та локалізацію за один прохід мережі. Однією з перших ефективних архітектур цього типу стала SSD (Single Shot MultiBox Detector), яка використовує багатомасштабні карти ознак для детекції об'єктів різних розмірів. RetinaNet запропонувала функцію втрат Focal Loss, що дозволила подолати проблему дисбалансу між фоновими та цільовими об'єктами [10].

Особливе місце серед одноетапних детекторів займає архітектура YOLO (You Only Look Once). На відміну від попередніх підходів, YOLO розглядає задачу виявлення об'єктів як єдину задачу регресії. Зображення поділяється на сітку, а мережа одночасно передбачає координати об'єктів, оцінки достовірності та ймовірності класів. Завдяки цьому забезпечується висока швидкість роботи та можливість обробки відео у реальному часі.

Архітектура YOLO пройшла значний шлях розвитку. YOLOv2 додала якірні прямокутники та пакетну нормалізацію, YOLOv3 реалізувала багатомасштабну детекцію, а YOLOv4 інтегрувала сучасні техніки аугментації та оптимізації. Версії

YOLOv5, YOLOv8 та YOLO11 значно спростили процес навчання й розгортання моделей, а також впровадили безякірний підхід і вдосконалені модулі обробки ознак [11].

Таким чином, сучасні методи виявлення об'єктів пройшли еволюцію від класичних алгоритмів на основі ручних ознак до високоефективних нейромережевих моделей глибокого навчання. Класичні методи залишаються актуальними для задач із обмеженими ресурсами або простими сценами, однак сучасні CNN-архітектури забезпечують значно вищу точність та універсальність. Найбільш перспективними напрямами розвитку є підвищення швидкодії моделей, зменшення обчислювальної складності, покращення багатомасштабної детекції та адаптація моделей до роботи в реальному часі на мобільних і вбудованих пристроях.

Список використаних джерел:

1. Sineglazov V.M. Applied artificial intelligence systems: computer vision. Visn. Nac. Akad. Nauk Ukr. 2024. (6): 43-48. https://doi.org/10.15407/visn2024.06.043
2. Шамуратов О. Ю., Шаховська Н. Б. Алгоритми контурного аналізу зображень. Науковий вісник НЛТУ України. 2019, т. 29, № 6. С. 123-127. . https://doi.org/10.15421/40290624
3. Canny edge detector. URL: https://scikit-image.org/docs/stable/auto_examples/edges/plot_canny.html
4. Білан С. М., Бережний Д. А. Надійний метод зіставлення зображень для сільськогосподарських знімків з БПЛА з використанням SIFT та ORB. Український журнал інформаційних технологій. 2025, т. 7, № 2. С. 58–66. https://doi.org/10.23939/ujit2025.02.058
5. Векторизація елементів растрового зображення. URL: https://pzs.dstu.dp.ua/ComputerGraphics/ vector/index.html
6. Ілащук М., Кушнір І., Мельничук С. Розпізнавання облич в реальному часі за допомогою бібліотеки OPENCV та мови програмування Python. Herald of Khmelnytskyi National University. Technical Sciences, 2024. 341(5), 140-144. https://doi.org/10.31891/2307-5732-2024-341-5-21
7. HoG & SVM. URL: https://robotcopper.github.io/Machine_Learning/HoG_and_SVM.html
8. Understanding Convolutional Neural Network (CNN) Architecture. URL: https://www.codecademy.com/article/understanding-convolutional-neural-network-cnn-architecture
9. R-CNN - Region-Based Convolutional Neural Networks. URL: https://www.geeksforgeeks.org/machine-learning/r-cnn-region-based-cnns/
10. Single Shot Multibox Detection. URL: https://d2l.ai/chapter_computer-vision/ssd.html
11. Горбань В., Левченко С. Порівняльний аналіз версій архітектури yolo для задач детекції об'єктів. Інноваційна наука: пошук відповідей на виклики сучасності (30 травня 2025 р., м. Київ). 2025. С. 374-377. DOI:10.62731/mcnd-30.05.2025.008

Ключові слова: комп'ютерний зір, детекція об'єктів, глибинне навчання, нейронні мережі, згорткові нейронні мережі, YOLO, комп'ютерний аналіз зображень, штучний інтелект

Key words: computer vision, object detection, deep learning, neural networks, convolutional neural networks, YOLO, computer image analysis, artificial intelligence

Науковий керівник: доцент Логінова Н.І.

ПОБУДОВА ПОВЕДІНКОВИХ ПРОФІЛІВ КОРИСТУВАЧІВ І СЕРВІСІВ ВЕБЗАСТОСУНКУ НА ОСНОВІ КЛАСТЕРИЗАЦІЇ СЕСІЙ

Дика Анастасія

асистент кафедри програмної інженерії
Національного університету «Одеська юридична академія»

Веборієнтовані інформаційні системи обслуговують різнорідні категорії користувачів – від анонімних відвідувачів до адміністраторів і автоматизованих сервісів. Кожна з цих категорій характеризується власним типовим патерном взаємодії із системою, а тому єдина узагальнена модель «нормальної» поведінки виявляється надмірно загальною та погано відрізняє легітимні відхилення від ознак атак.

Поширені підходи до виявлення аномалій у вебвзаємодіях здебільшого спираються на єдину ймовірнісну модель нормальної поведінки, у якій ступінь відхилення оцінюється через від'ємний логарифм правдоподібності сесії [1, 2]. Проте така єдина модель не враховує неоднорідності аудиторії: дії, типові для адміністратора, можуть бути аномальними для звичайного користувача, і навпаки, що призводить до зростання частки хибнопозитивних спрацювань.

Перспективним напрямом є побудова не однієї, а множини поведінкових профілів, кожен з яких описує «норму» окремої ролі або групи користувачів без попереднього розмічення даних [3]. Схожий підхід до ідентифікації трафік-профілів легітимних користувачів і ботів на основі навчання без учителя продемонстрував свою ефективність у роботі [4].

Досліджується підхід до побудови поведінкових профілів користувачів і сервісів вебзастосунку на основі кластеризації векторних представлень сесій. Висувається гіпотеза, що оцінювання аномальності відносно профілю найближчої групи забезпечує вищу точність і нижчий рівень хибних спрацювань, ніж оцінювання відносно єдиної усередненої моделі поведінки.

Поведінку вебсистеми подано на рівні сесій. Сесія визначається як впорядкована послідовність HTTP-запитів і переводиться у числовий вектор ознак за допомогою функції відображення $f : \sigma \to \mathbb{R}^d$. Оскільки сесія складається переважно з текстових і категоріальних даних (URL, методи, заголовки, параметри запитів, часові інтервали), функція f реалізується через попередню токенізацію запитів та векторизацію за схемою TF-IDF над символьними й токенними n-грамами; для врахування семантики послідовності дій можуть додатково застосовуватися нейромережеві векторні представлення або контекстні вкладення (Doc2Vec). Числові ознаки (тривалість сесії, частота запитів, обсяг переданих

даних) нормалізуються та конкатенуються з текстовими, утворюючи підсумковий вектор x_i, а вся навчальна вибірка позначається $X = \{x_1, \ldots, x_n\}$.

Задача профілювання полягає в автоматичному розбитті вибірки X на K груп (кластерів), кожна з яких відповідає окремому поведінковому профілю – ролі користувача або типу сервісу. Розбиття будується шляхом мінімізації сумарної внутрішньокластерної евклідової відстані між векторами сесій та центроїдами груп (алгоритм k-means). Центроїд μ_j кожної групи є середнім значенням усіх векторів, що до неї належать, і слугує еталонним профілем ролі.

Для кожної групи будується локальна модель нормальної поведінки $P(\sigma \mid \theta_j)$, параметри θ_j якої оцінюються лише за сесіями цієї групи. Поведінка системи в цілому подається як зважена суміш таких профілів із ваговими коефіцієнтами π_j, що відображають частку кожної ролі в загальній аудиторії.

Для нової сесії σ з вектором ознак x спочатку визначається найближчий профіль за евклідовою відстанню до центроїдів: $c(x) = \arg\min_j \|x - \mu_j\|$. Оцінка аномальності обчислюється відносно моделі саме цієї групи:

$$Score(\sigma) = -\ln P(\sigma \mid \theta_{c(x)}) \quad (1)$$

Сесія визнається аномальною, якщо її оцінка перевищує індивідуальний поріг τ_j, що калібрується окремо за валідаційною вибіркою кожної групи (наприклад, за ROC-аналізом). Індивідуалізація порогу дозволяє враховувати різну природну варіативність поведінки різних ролей [2].

Порівняно з єдиною моделлю, суміш профілів дозволяє відокремити поняття «норми» для кожної ролі: нетипова для однієї групи поведінка не маскується усередненою статистикою всієї аудиторії [3]. Кількість профілів K визначається автоматично за внутрішніми критеріями якості кластеризації – зокрема за коефіцієнтом силуету або індексом Девіса-Болдіна, що оцінюють компактність і роздільність груп [1]. Це усуває потребу в апріорному знанні рольової структури системи та дозволяє виявляти приховані категорії користувачів і автоматизованих сервісів. У проведеному експерименті оптимальне значення $K = 4$ було знайдено автоматично за максимумом коефіцієнта силуету.

Оскільки алгоритм k-means спирається на евклідову метрику і найкраще описує сферичні кластери, поряд із ним доцільно застосовувати алгоритм DBSCAN, що не потребує задання K, стійкий до груп довільної форми та природно відносить поодинокі викиди до шуму, що узгоджується із самою задачею виявлення аномалій [2, 5]. Додатковою перевагою є інтерпретованість результату: центроїди μ_j та частки π_j утворюють зрозумілий опис типових сценаріїв роботи системи, а раптова поява нового щільного кластера або зміщення центроїда у часі може сама по собі слугувати індикатором аномальної активності чи зміни характеру навантаження [2].

Для перевірки гіпотези проведено експериментальну апробацію на загальнодоступному наборі HTTP-запитів CSIC 2010 (загалом 72 000 нормальних і 25 000 аномальних запитів). Сесії векторизовано за схемою TF-IDF над символьними n-грамами, після чого порівняно дві конфігурації: базову однопрофільну

модель (K = 1) та запропонований профільний підхід із автоматично визначеною кількістю груп (K = 4). Схожі ненавчальні схеми виявлення аномалій у вебтрафіку демонструють високу ефективність у нещодавніх дослідженнях [5, 6]. Результати наведено в таблиці 1.

Таблиця 1 – Порівняння якості виявлення аномалій

Модель	**Точність (Precision)**	**Повнота (Recall)**	**F1-міра**
Базова (K = 1)	0,89	0,86	0,87
Профільна (запропонована, K = 4)	0,96	0,90	0,93

Профільний підхід забезпечив зростання F1-міри приблизно на 6 відсоткових пунктів переважно за рахунок підвищення точності (Precision), тобто скорочення хибнопозитивних спрацювань – саме того ефекту, що очікувалося від урахування рольової неоднорідності аудиторії. Слід зазначити, що отримані значення мають характер попередньої апробації та мають обмеження: набір CSIC 2010 є синтетичним і не відображає повною мірою реального трафіку сучасних вебзастосунків. Тому результати потребують подальшої валідації на ширшому наборі реальних журналів експлуатації.

Запропоновано підхід до побудови поведінкових профілів користувачів і сервісів вебзастосунку на основі кластеризації векторних представлень сесій. Поведінку системи представлено як зважену суміш локальних профілів з індивідуальними параметрами та порогами виявлення аномалій. Експериментальна апробація на наборі CSIC 2010 підтвердила доцільність урахування неоднорідності ролей користувачів та заклала основу для подальших досліджень у напрямі метрик відхилення поведінки веборієнтованих систем.

Список використаних джерел

1. Xu D., Tian Y. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*. 2015. Vol. 2, No. 2. P. 165-193. DOI: 10.1007/s40745-015-0040-1.
2. Aggarwal C. C. Outlier Analysis. 2nd ed. Cham : *Springer*, 2017. 466 p. DOI: 10.1007/978-3-319-47578-3.
3. Suchacka G., Iwanski J. Identifying legitimate Web users and bots with different traffic profiles – an Information Bottleneck approach. *Knowledge-Based Systems*. 2020. Vol. 197. Article 105875. DOI: 10.1016/j.knosys.2020.105875.
4. Gniewkowski M., Maciejewski H., Surmacz T., Walentynowicz W. Sec2vec: Anomaly Detection in HTTP Traffic and Malicious URLs. *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23)*. New York : ACM, 2023. P. 1154-1162. DOI: 10.1145/3555776.3577663.
5. Yılmazer Demirel D., Sandıkkaya M. T. ACUM: An Approach to Combining Unsupervised Methods for Detecting Malicious Web Sessions. *Proceedings of the 8th International Conference on Computer Science and Engineering (UBMK 2023)*. Burdur : IEEE, 2023. P. 288-293. DOI: 10.1109/UBMK59864.2023.10286727.

6. Benova L., Hudec L. Comprehensive Analysis and Evaluation of Anomalous User Activity in Web Server Logs. *Sensors*. 2024. Vol. 24, No. 3. Article 746. DOI: 10.3390/s24030746.

Ключові слова: поведінкові профілі, кластеризація сесій, веборієнтовані системи, виявлення аномалій, профілювання користувачів, машинне навчання

Keywords: behavioral profiles, session clustering, web-oriented systems, anomaly detection, user profiling, machine learning

Науковий керівник: доцент Трофименко О. Г.

АВТОМАТИЗАЦІЯ ПРОЄКТУВАННЯ ТА НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ У ЛОКАЛЬНОМУ ПРОГРАМНОМУ СЕРЕДОВИЩІ

Степанов Максим

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Розвиток інформаційних систем супроводжується накопиченням значних обсягів структурованих даних, які здебільшого зберігаються у табличному вигляді. Аналіз таких даних є важливою умовою автоматизації бізнес-процесів, підтримки прийняття управлінських рішень та побудови інтелектуальних програмних систем [1]. Водночас розроблення моделей машинного навчання, зокрема нейронних мереж, залишається складним і ресурсомістким процесом, що потребує навичок програмування, знання математичних основ та досвіду роботи зі спеціалізованими фреймворками.

Традиційний підхід до розв'язання задач класифікації та регресії за допомогою нейронних мереж передбачає написання програмного коду для підготовки даних, налаштування архітектури моделі, вибору параметрів навчання та оцінювання результатів. У результаті формується високий поріг входження для користувачів, які працюють з даними, але не мають достатньої підготовки у сфері програмування або глибокого навчання.

Одним із поширених способів спрощення цього процесу є використання хмарних платформ автоматизованого машинного навчання. Проте такі сервіси мають низку обмежень, серед яких варто виокремити комерційну вартість обчислень, залежність від інтернет-з'єднання та ризики, пов'язані з передаванням даних на сторонні сервери. Для підприємств, освітніх установ і дослідницьких проєктів, що працюють з конфіденційною або спеціалізованою інформацією, ці обмеження можуть бути суттєвими. Тому актуальним є розроблення локального програмного середовища, яке забезпечує автоматизацію проєктування, навчання та оцінювання нейронних мереж без передавання даних у хмарну інфраструктуру.

Математичною основою нейронних мереж є модель штучного нейрона, що виконує зважування вхідних сигналів і передає результат через функцію

активації. Для роботи з табличними даними доцільним є використання багатошарових перцептронів, які можуть застосовуватися для задач класифікації та регресії. Водночас локальне середовище автоматизованого проєктування може підтримувати не лише повнозв'язні архітектури, а й рекурентні шари RNN/LSTM або одномірні згорткові мережі 1D CNN [2]. Використання таких архітектур розширює можливості системи для аналізу послідовностей, часових залежностей і локальних закономірностей у даних.

Важливим етапом побудови нейронної мережі є попередня обробка даних. Алгоритми навчання, що базуються на градієнтній оптимізації, є чутливими до масштабу ознак, тому нормалізація та стандартизація даних мають бути автоматизовані [3]. Доцільним архітектурним рішенням є збереження параметрів масштабування разом із навченою моделлю. Збереження параметрів масштабування разом із моделлю забезпечує однакову обробку навчальних і нових даних під час її подальшого використання.

Для реалізації такого програмного середовища доцільно використовувати мову Python, яка є одним із основних інструментів у сфері Data Science та машинного навчання. Бібліотеки TensorFlow і Keras можуть забезпечувати побудову, навчання та збереження нейронних мереж, а PyQt6 – створення графічного інтерфейсу користувача. Використання архітектурного підходу Model-View-Controller дає змогу відокремити математичне ядро системи від візуальних компонентів, що підвищує модульність і зручність супроводу програмного коду.

Окрему увагу слід приділити забезпеченню стабільності та зручності роботи користувача. Оскільки навчання нейронної мережі може потребувати значного часу, цей процес доцільно виконувати в окремому фоновому потоці. Виконання навчання у фоновому потоці дозволяє уникнути блокування інтерфейсу та забезпечує коректне відображення проміжних результатів навчання. Для збереження історії експериментів, параметрів моделей і результатів оцінювання може використовуватися вбудована база даних SQLite.

Завершальним етапом автоматизованого циклу є оцінювання якості побудованої моделі. Для задач класифікації можуть використовуватися точність і функція втрат на основі перехресної ентропії, а для задач регресії – середньоквадратична помилка та інші метрики якості [4]. Візуалізація динаміки навчання дає змогу користувачеві аналізувати поведінку моделі, виявляти ознаки перенавчання та коригувати параметри без повного перезапуску процесу.

Отже, розроблення локального програмного середовища для автоматизованого проєктування та навчання нейронних мереж є актуальним завданням. Розроблене програмне середовище поєднує можливості попередньої обробки даних, налаштування архітектури моделі, навчання, оцінювання результатів і збереження побудованих моделей. Його використання може спростити роботу з нейронними мережами, підвищити автономність дослідницького процесу та зменшити залежність користувачів від хмарних сервісів.

Список використаних джерел

1. Bruno Lee. Data Processing: The Backbone of Modern Information Systems. *International Journal of Advancements in Technology*. 2024, Vol. 15(4). DOI: https:// 10.35841/0976-4860.24.15.296
2. Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, Daniel J. Inman. 1D Convolutional Neural Networks and Applications: A Survey. *Mechanical Systems and Signal Processing*. 2021. Vol. 151. P. 107398. DOI: https://doi.org/10.1016/j.ymssp.2020.107398
3. Xing Wan. Influence of feature scaling on convergence of gradient iterative algorithm. *Journal of Physics: Conference Series*. 2019, Vol. 1213(3). DOI: https://doi.org/10.1088/1742-6596/1213/3/032021
4. Classification vs Regression in Machine Learning. GeeksforGeeks. URL: https://surl.li/niczkf

Ключові слова: штучні нейронні мережі, автоматизація навчання, локальне програмне забезпечення, попередня обробка даних, графічний інтерфейс, MVC

Keywords: artificial neural networks, training automation, local software, data preprocessing, graphical user interface, MVC

Науковий керівник: доцент Лобода Ю. Г.

АДАПТИВНА МАРШРУТИЗАЦІЯ КОНТЕКСТУ В ДВОРІВНЕВИХ АГЕНТНИХ АРХІТЕКТУРАХ

Манаков Сергій

кандидат технічних наук, доцент,
доцент кафедри програмної інженерії
Національного університету «Одеська юридична академія»

Застосування великих мовних моделей для побудови автономних агентів породжує два фундаментально різних класи інтеграційних задач. Перший клас охоплює вертикальну інтеграцію агентів з інструментами та джерелами даних: базами знань, API-сервісами, файловими системами, обчислювальними середовищами. Другий клас стосується горизонтальної координації між спеціалізованими агентами, кожен з яких відповідає за окремий аспект складної задачі. Дослідження демонструють, що багатоагентні системи забезпечують підвищення продуктивності понад 70% порівняно з одноагентними підходами для широкого класу задач, включаючи програмування, аналіз даних та автоматизацію бізнес-процесів.

Індустрія відреагувала на ці виклики двома протоколами. Model Context Protocol (MCP), анонсований компанією Anthropic у листопаді 2024 року, став першим відкритим стандартом вертикальної інтеграції. Agent-to-Agent Protocol (A2A), представлений Google у квітні 2025 року та переданий під управління Linux Foundation, зосереджується на горизонтальній координації та підтримується

понад 100 провідними компаніями, серед яких AWS, Microsoft, Cisco та Salesforce. Проте існуючі рішення здебільшого розглядають ці протоколи ізольовано, тоді як реальні промислові системи потребують їх узгодженого функціонування.

Мета роботи – запропонувати практичну архітектуру дворівневої агентної системи з алгоритмом динамічного вибору протоколу та ефективними механізмами кешування контексту, що сумісна з наявними реалізаціями MCP та A2A без їх модифікації.

Аналіз специфікацій MCP (версія 2025-11-25) та A2A (версія 0.3.0) виявив їх комплементарний характер: кожен протокол оптимізовано для специфічного класу операцій (табл. 1).

Таблиця 1 – Порівняльна характеристика протоколів MCP та A2A

Аспект	**MCP**	**A2A**
Призначення	Доступ до інструментів та даних	Міжагентна координація
Модель взаємодії	Виклик функцій (RPC)	Діалогова (задачі зі станом)
Тривалість задач	Переважно синхронні	Синхронні та асинхронні
Стан між викликами	Stateless	Stateful (стан задачі)
Транспорт	JSON-RPC 2.0, Stdio/HTTP	JSON-RPC 2.0, gRPC, REST
Виявлення сервісів	Конфігурація клієнта	Agent Cards, реєстри
Безпека	OAuth 2.0, API keys	OAuth 2.0, mTLS, API keys

MCP визначає три типи серверних примітивів: Tools (виконувані функції), Resources (структуровані дані з URI-адресацією) та Prompts (шаблонні інструкції). Завдяки делегуванню обчислень MCP-серверу досягається редукція токенів до 98.7% порівняно з передачею проміжних результатів через контекст агента. A2A оперує концепцією задачі з восьми-станним життєвим циклом (submitted, working, input-required, auth-required, completed, canceled, failed, rejected) та підтримує потоковий вивід через Server-Sent Events для тривалих операцій. Ключовий інженерний висновок: ці протоколи є взаємодоповнювальними, а не конкурентними, що обґрунтовує доцільність дворівневої архітектури.

Запропонована архітектура базується на компоненті Context Router, що приймає вхідні запити та маршрутизує їх до відповідного рівня обробки. Кожен агент виступає одночасно клієнтом MCP для вертикального доступу до ресурсів та учасником A2A-мережі для горизонтальної координації.

Алгоритм динамічного вибору протоколу класифікує запити за шестивимірним вектором ознак $F = (f_1, f_2, f_3, f_4, f_5, f_6)$:

$f_1 \in [0,1]$ – ступінь потреби у доступі до інструментів;

$f_2 \in [0,1]$ – ступінь потреби у міжагентній координації;

$f_3 \in \{0,1\}$ – асинхронна природа задачі;

$f_4 \in \{0,1\}$ – необхідність збереження стану між взаємодіями;

$f_5 \in [0,\infty)$ – вимоги до максимальної латентності (мс);

$f_6 \in \{low, medium, high\}$ – рівень вимог до безпеки.

На основі значень вектора маршрутизатор обирає один з трьох режимів роботи. Режим MCP-only активується для задач, що потребують лише доступу до інструментів без міжагентної координації ($f_1 > \theta_1$, $f_2 < \theta_2$). Режим A2A-only застосовується для задач, що вимагають координації агентів без зовнішніх інструментів ($f_2 > \theta_2$, $f_3 = 1$). Гібридний режим (Hybrid) використовується для комплексних задач з декомпозицією на підзадачі та їх розподілом між рівнями.

Гібридна стратегія будує граф залежностей підзадач $G = (T, E)$ та визначає оптимальне відображення $M: T \rightarrow \{V, H\}$ через мінімізацію сумарної вартості виконання та перемикань між рівнями. Практично застосовуються жадібні евристики з поліноміальною складністю $O(|T|^2 \log|T|)$, що забезпечують прийнятні рішення при типових обсягах задач ($|T| \leq 50$ підзадач).

Передача контексту між агентами та рівнями архітектури є основним джерелом обчислювальних витрат. Запропоновано трирівневу систему кешування, де кожен рівень оптимізовано для свого часового горизонту:

L_1 – внутрішньоагентний кеш проміжних результатів, TTL порядку секунд, оптимізований для швидкого доступу до нещодавно використаних даних;

L_2 – кеш на рівні MCP-сервера для повторюваних запитів до інструментів, TTL порядку хвилин, амортизує вартість ідентичних операцій;

L_3 – розподілений міжагентний кеш для A2A-сесій, TTL порядку годин, забезпечує persistence тривалих багатоагентних сесій.

Система спирається на перевірені промислові рішення (табл. 2), що не потребують доробки: Prompt Caching від Anthropic на рівні L_1-L_2, PagedAttention (vLLM) для управління KV-cache на рівні L_2, розподілені семантичні кеші (GPTCache, llm-d) на рівні L_3.

Таблиця 2 – Метрики ефективності промислових механізмів кешування

Механізм	**Економія ресурсів**	**Зменшення латентності**	**Cache Hit Rate**
Prompt Caching (Anthropic)	до 90% токенів	до 85%	–
PagedAttention (vLLM)	пропускна здатність у 24 рази	–	–
llm-d (Kubernetes)	–	88% TTFT	87%
GPTCache (Zilliz)	до 70% токенів	до 90%	60-80%

Загальна ефективність системи визначається як композиція ефективностей рівнів з урахуванням ймовірностей cache hit (p_1, p_2, p_3) та відповідних коефіцієнтів економії (e_1, e_2, e_3). При типовому розподілі навантаження (60% MCP-запитів, 40% A2A-задач) прогнозована загальна економія становить 70-80% для сценаріїв з повторюваними запитами. Пріоритет витіснення з кешу визначається

модифікованим LRU з урахуванням частоти звернень, актуальності та вартості повторного обчислення запису.

Головна практична перевага запропонованого підходу полягає у його сумісності з наявною екосистемою: архітектура реалізується поверх існуючих реалізацій MCP та A2A без модифікації базових протоколів. Context Router є окремим компонентом, що може бути впроваджений як middleware-шар у будь-якій системі, яка вже використовує MCP або A2A. Трирівневий кеш інтегрується з переліченими комерційними рішеннями через стандартні API.

Типові сценарії застосування: системи автоматизації бізнес-процесів, де агенти-аналітики (MCP) взаємодіють з агентами-виконавцями (A2A); конвеєри обробки документів з паралельним доступом до зовнішніх інструментів та міжагентною агрегацією результатів; платформи розробки програмного забезпечення на кшталт MetaGPT, де задачі розподіляються між спеціалізованими агентами.

Перспективи подальших досліджень включають: (1) реалізацію прототипу та бенчмаркінг на репрезентативних навантаженнях з реальними LLM; (2) розробку методів автоматичного налаштування порогів маршрутизації (θ_1, θ_2, θ_3) на основі reinforcement learning для адаптації до змінюваних патернів навантаження; (3) інтеграцію з механізмами формальної специфікації безпекових політик (Open Policy Agent) для контролю інформаційних потоків між рівнями.

У роботі запропоновано практичну архітектуру дворівневої агентної системи на основі LLM, що об'єднує протоколи MCP та A2A. Встановлено комплементарний характер протоколів: MCP оптимізовано для швидких синхронних операцій доступу до ресурсів, A2A – для тривалих асинхронних задач зі збереженням стану. Розроблено алгоритм динамічного вибору протоколу на основі шестивимірного вектора ознак з підтримкою гібридної стратегії та декомпозицією комплексних задач. Описано трирівневу архітектуру кешування з очікуваною економією 70-80% обчислювальних ресурсів для повторюваних сценаріїв. Архітектура сумісна з наявними реалізаціями протоколів і може бути впроваджена без їх модифікації.

Список використаних джерел

1. Anthropic. Model Context Protocol Specification. Version 2025-11-25. URL: https://spec.modelcontextprotocol.io/
2. Google Cloud. Agent-to-Agent Protocol Specification. Version 0.3.0. URL: https://google.github.io/A2A/
3. Tran K.-T., Dao D, Nguyen M.-D. et al. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. arXiv:2501.06322. 2025.
4. Yue Y., Zhang G., Liu B. et al. MasRouter: Learning to Route LLMs for Multi-Agent Systems. arXiv:2502.11133. 2025.
5. Anthropic. Prompt Caching Documentation. URL: https://docs.anthropic.com/en/docs/build-with-claude/prompt-caching

6. Kwon W., Li Z., Zhuang S. et al. Efficient Memory Management for Large Language Model Serving with PagedAttention. Proc. SOSP '23. 2023.
7. Ruan C., Bi C., Zheng K. et al. Asteria: Semantic-Aware Cross-Region Caching for Agentic LLM Tool Access. arXiv:2509.17360. 2025.

Ключові слова: багатоагентні системи, LLM, MCP, A2A, маршрутизація контексту, кешування

Keywords: multi-agent systems, LLM, MCP, A2A, context routing, caching

АРХІТЕКТУРА АВТОНОМНИХ ШІ-АГЕНТІВ НА БАЗІ RAG ТА ОРКЕСТРАТОРА N8N ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

Косяк Микола

студент 1-го курсу магістратури
факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Розвиток інформаційних технологій супроводжується переходом від жорстко алгоритмізованих систем автоматизації до адаптивних рішень на базі штучного інтелекту. Класичні інструменти автоматизації потребують попереднього опису кожного кроку виконання процесу, тому мають обмежену ефективність під час роботи з неструктурованими даними, змінними сценаріями взаємодії та непередбачуваними запитами користувачів. Поява великих мовних моделей (LLM) створила передумови для використання ШІ-агентів – програмних компонентів, здатних аналізувати запит, планувати послідовність дій, звертатися до зовнішніх інструментів і формувати відповідь з урахуванням контексту.

Однією з ключових проблем застосування LLM у бізнес-середовищі є обмежений доступ до актуальних внутрішніх даних організації. Модель може генерувати неточні або неперевірені відповіді, якщо її контекст не містить релевантної інформації. Для зменшення цього ризику використовується підхід Retrieval-Augmented Generation, або RAG, який поєднує генеративні можливості мовної моделі з пошуком у зовнішніх джерелах даних [1]. Такий підхід є основою для побудови автономних ШІ-агентів, які поєднують можливості великих мовних моделей, пошук у корпоративних даних та виконання дій через зовнішні інструменти. Для розгортання таких агентних сценаріїв можуть використовуватися оркестратори автоматизації, зокрема n8n, що забезпечують зв’язок між мовною моделлю, векторним сховищем, API-сервісами та бізнес-процесами [2].

Великі мовні моделі (LLM) навчаються на масивах даних, актуальність яких обмежується датою завершення тренування. Для бізнес-застосунків потрібен доступ до актуальних внутрішніх даних, тому перенавчання моделі під кожну зміну інформації є економічно та технічно недоцільним.

Архітектура RAG передбачає розділення процесу обробки запиту на два основні етапи: пошук релевантної інформації та генерацію відповіді. На першому етапі внутрішні документи, записи баз даних або API-відповіді розбиваються на

текстові фрагменти. Кожен фрагмент перетворюється на векторне представлення за допомогою embedding-моделі та зберігається у векторному сховищі. Як такі сховища можуть використовуватися Pinecone, Qdrant або PostgreSQL з розширенням pgvector. Під час обробки запиту користувача система також перетворює його на вектор і виконує семантичний пошук найближчих за змістом фрагментів. На другому етапі знайдені фрагменти додаються до контексту великої мовної моделі, яка формує відповідь на основі отриманих даних [1].

На відміну від звичайного чатбота, автономний ШІ-агент не лише відповідає на запитання, а й може виконувати дії у зовнішньому середовищі. Його архітектура зазвичай містить три основні компоненти: мовну модель, пам'ять та інструменти. Мовна модель виконує роль центрального компонента, який аналізує запит, визначає намір користувача та обирає подальшу дію. Пам'ять забезпечує збереження контексту поточної взаємодії або довгострокової інформації, необхідної для повторних звернень. Інструменти дають агенту змогу взаємодіяти із зовнішніми сервісами: виконувати API-запити, звертатися до баз даних, запускати сценарії автоматизації або передавати структуровані дані до інших інформаційних систем [2, 3].

Для побудови агентних систем можуть використовуватися як програмні фреймворки, так і low-code платформи. До першої групи належать LangChain та LlamaIndex, які надають розробнику гнучкі засоби для створення RAG-пайплайнів, агентів, інструментів і механізмів пам'яті. До другої групи належать платформи автоматизації, зокрема n8n та Make. Перевагою таких платформ є візуальне проєктування потоків даних, інтеграція з популярними сервісами та можливість швидкого створення прототипів бізнес-процесів.

Оркестратор n8n може використовуватися для реалізації агентних сценаріїв, оскільки підтримує побудову автоматизованих workflow, роботу з API, обробку даних і підключення ШІ-компонентів. У таких сценаріях n8n виконує роль проміжного шару між користувацьким запитом, великою мовною моделлю, векторним сховищем і корпоративними сервісами. Наприклад, користувач може надіслати запит у чат-боті, після чого workflow в n8n виконує пошук релевантних даних у векторній базі, передає їх до мовної моделі, отримує відповідь і, за потреби, викликає зовнішній API для зміни стану бізнес-об'єкта.

Порівняно з Make, платформа n8n надає ширші можливості для самостійного розгортання, програмованої обробки даних і роботи з кастомними сценаріями. Make орієнтований переважно на хмарну автоматизацію типових інтеграцій між сервісами. OpenClaw може розглядатися як засіб для побудови локальних агентних рішень, однак потребує глибшої технічної підготовки для створення та налаштування спеціалізованих навичок агентів [4].

Застосування RAG та агентної архітектури є перспективним для бізнес-процесів, у яких поєднуються неструктуровані запити користувачів, потреба в актуальних даних і необхідність виконання дій у зовнішніх системах. До таких процесів належать технічна підтримка, обробка клієнтських звернень, пошук у

внутрішній документації, автоматизація заявок, управління розкладами, CRM-сценарії та супровід SaaS-платформ. У цих випадках RAG забезпечує доступ до релевантного контексту, а агентна логіка дає змогу не лише сформувати відповідь, а й ініціювати подальші дії.

Поєднання RAG, великих мовних моделей і оркестраторів автоматизації формує основу для створення агентних систем, здатних працювати з актуальними даними та виконувати дії у зовнішніх сервісах. Підхід RAG дає змогу доповнювати запит моделі релевантним контекстом і зменшувати ризик формування неточних відповідей. Оркестратори на зразок n8n забезпечують зв'язок між мовною моделлю, векторним сховищем, API-сервісами та бізнес-процесами. Подальший розвиток таких рішень пов'язаний з удосконаленням стратегій розбиття документів на фрагменти, оптимізацією векторного пошуку та побудовою мультиагентних сценаріїв для складних бізнес-завдань.

Список використаних джерел

1. Lewis P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*. 2020, Vol. 33, P. 9459-9474. DOI: https://doi.org/10.48550/arXiv.2005.11401.
2. n8n Official Documentation: Advanced AI workflows. URL: https://docs.n8n.io/advanced-ai.
3. OpenAI API Developers: Function calling. URL: https://developers.openai.com/api/docs/guides/function-calling.
4. OpenClaw GitHub Repository. Architecture overview. URL: https://github.com/openclaw/openclaw.

Ключові слова: штучний інтелект, ШІ-агенти, RAG, автоматизація, n8n, векторні бази даних

Keywords: artificial intelligence, AI agents, RAG, automation, n8n, vector databases

Науковий керівник: доцент Лобода Ю.Г.

КОНЦЕПЦІЯ "WETWARE": ВИКОРИСТАННЯ БІОЛОГІЧНИХ НЕЙРОНІВ ЛЮДИНИ У РОЗРОБЦІ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

Гура Володимир

кандидат технічних наук,
доцент кафедри штучного інтелекту та математичного моделювання
Національного університету «Одеська юридична академія»

Лавренко Дмитро

студент 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету "Одеська юридична академія"

Станом на зараз сучасна IT індустрія як правило базується на класичній архітектурі напівпровідникових компонентів та архітектурі фон Неймана (головним принципом якої є фундаментальний принцип побудови обчислювальних систем, при якій програма та її дані зберігаються в одній і тій самій пам'яті). Але з цей підхід майже вичерпав свій потенціал. Це пов'язано з фізичними обмеженнями розмірів транзисторів. Це обґрунтовано законом Мура згідно з яким кількість транзисторів на кристалі інтегральної схеми подвоюється приблизно кожні два роки. Саме це змушує людство шукати новий за своїм принципом підхід до розвитку обчислювальних можливостей комп'ютерних систем. Саме тут на допомогу приходить концепція "Wetware" (з англ. Вологий/біологічний компонент), суть цієї концепції полягає у використанні біологічних систем таких як людська нервова система та людський мозок у якості обчислювальних пристроїв. За своїм сенсом це спроба поєднати класичні цифрові алгоритми з біологічними компонентами, наприклад людськими нейроструктурами. Для цього навіть не потрібно пряме втручання в людський мозок. Сучасні біологічні технології дозволяють використати стовбурові клітини (iPSC) які в свою чергу можна отримати зі звичайних клітин шкіри і переробити (за допомогою біологічного принципу самоорганізації клітин) у біологічні нейрони, які згодом сформують тривимірні структури які прийнято називати органоїди. В решті решт ці органоїди можна підключити до мікроелектронних датчиків, що дасть змогу органоїдним структурам отримувати вхідні дані у вигляді електричних сигналів (імпульсів) і зчитувати їх відповідь. Що теоретично і дає змогу використовувати симбіоз обчислювальної техніки та живої матерії [1].

Потреба реалізації таких доволі радикальних рішень полягає у зменшенні споживання великої енергетичної потужності обчислювальних систем штучного інтелекту. Щоб підтримувати роботу великих LLM (large language model), таких як Chat GPT, потрібно будували величезні дата-центри і витрачати на них гігаватні потужності електроенергії. Результатом роботи такого дата-центру є великий вуглецевий слід приблизно такий же як і від 1000 автівок за весь період їх планової експлуатації. Тому саме людський мозок з його еталонною

енергоефективністю, в перерахунку на традиційні одиниці вимірювання споживає приблизно 20 Вт енергії і при цьому здатний генерувати дуже складні рішення і навчатись на порядки краще за обчислювальні класичні аналоги при мінімальній кількості навчальних прикладах. При цьому такий підхід буде вкрай ефективний у вирішенні доволі складних мультимодальних задач. Якщо гарно інтегрувати це рішення в IT інфраструктуру то людство отримає принципово нові обчислювальні системи, які будуть в рази ефективнішими за існуючи технологічні рішення.

Але на шляху до створення подібних систем існує багато перешкод. По-перше це система забезпечення життєдіяльності біологічної частини комп'ютера, на відміну від свого існуючого аналога біологічний органоїд може загинути від нестачі корисних речовин, інфекцій, перепаду температури чи зміни навколишнього середовища. Це вимагає створення надійної системи на кшталт мікрофлюїдних систем (такі системи складаються з мініатюрних трубок і помп які постійно б постачали поживні речовини і відводили продукти життєдіяльності) і суворих протоколів безпеки з резервним електропостачанням и запасом всіх необхідними розчинів і речовин [2].

Також масштабним викликом буде створення надійного інтерфейсу для взаємодії між біологічною і технологічною частинами. Також біологічні нейрони постійно генерують фоновий шум який потрібно відсортовувати щоб відокремити корисний сигнал, для швидкої взаємодії потрібно буде розробити спеціальну архітектуру програмного забезпечення яка зможе забезпечити мінімальну затримку між живими нейронами та обчислювальними системами. Низько рівневі процеси такі як драйвери для роботи з мікроелектронікою та первина обробка сигналів реалізується за допомогою мов програмування таких як C/C++ щоб забезпечити максимальну швидкість обчислень. У свою чергу як високорівнева логіка як машинне навчання для декодування спайків нейронних мереж та їхньої прив'язки до віртуальної середи буде робиться з допомогою такої мови програмування як Python [3].

Першими хто зміг довести життєздатність такого підходу є експеримент стартапу Cortical Labs відомий як DishBrains. Успішність експерименту полягала в тому що дослідникам вдалося розмістити на своєму чипі близько 800 тисяч живий нейронів і навчили їх взаємодіяти з комп'ютерною ретро-грою Pong (аналог звичайного тенісу) і згодом з комп'ютерною грою Doom. Інформація про положення ворогів чи м'ячів у грі конвертувалась в електричні імпульси, коли нейрони відбивали м'яч система давала їм впорядкований електричний сигнал і в випадку пропуску хаотичний. Це все базується на принципі мінімізації вільної енергії (бо людський мозок прагне до передбачуваності середовищ), клітинна культура швидко переналаштувала свої нейроні зв'язки щоб уникати хаотичні стимули і почала грати з урахуванням вхідних даних і грати «по людськи». Цей експеримент чітко доводить що людство вже зараз здатне створювати стабільні

API для комунікації цифрових систем з біологічною частиною і живими клітинами які здатні до самонавчання у реальному часі [4].

Ще окрім суто інженерних питань розвитку технології Wetware заважає безпрецедентні біоетичні проблеми. Коли клітинна культура мозку навчається взаємодії зі складним середовищем постає питання на якому етапі така система може набути зачатків самосвідомості або відчути по типу болі чи виснаження і так далі. Сучасна наука вважає що органоїди які складаються з кількох тисяч не мають свідомості однак з поступовим збільшенням їхніх розмірів наприклад до мільярдів нейронів, то дослідникам доведеться розробляти нові етичні протоколи і регуляторні норми для таких техніко-біологічних систем.

Підсумовуючи все вище сказане можна впевнено стверджувати що подальший розвиток технології Wetware, здатний радикально змінити правила гри в IT індустрії. Симбіозом передової мікробіології, інженерії та програмування дає реальний шанс подолати кризу енергоспоживання сучасних систем ШІ. Здатність живих нейронів адаптуватися до віртуальних умов свідчить про їхній імовірний величезний потенціал у ролі співпроцесорів. В майбутньому подібні системи цілковито зможуть посунути свої існуючи обчислювальні системи на ринку відкривши нову еру у розвитку техно-біологічних систем вже в найближчому майбутньому.

Список використаних джерел

1. Smirnova, T., & Kovalenko, V. (2025). Основи нейрогібридних обчислювальних систем та інтеграція біологічних процесорів. Київ: Наукова думка. DOI: https://doi.org/10.3389/fsci.2023.1017235
2. Kagan, B. J. et al. (2022). In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. *Neuron*, 110(23), 3952-3969. DOI: https://doi.org/10.1016/j.neuron.2022.09.001]
3. Shi, H., Kowalczewski, A., Vu, D., Liu, X., Salekin, A., Yang, H., & Ma, Z. (2024). Organoid intelligence: Integration of organoid technology and artificial intelligence in the new era of in vitro models. Medicine in novel technology and devices, 21, 100276. URL:
4. Gaps and Ethical Trajectories in Commercial Biocomputing: An Investigative Audit of Cortical Labs and Fin. Lefkowitz, D. (2026). Governance alSpark. Available at SSRN 6411258. DOI: https://doi.org/10.1016/j.medntd.2023.100276

Ключові слова: нейронні мережі, штучний інтелект, органоїдний інтелект, розробка ШІ, рішення в галузі розробки ШІ, технології Wetware

Key words: Neural ntworks, Artificial Intelligence, AI, Organoid intelligence, AI development, new solutions in the field of AI development, Wetware technologies

Підсекція 4. Програмна інженерія та розробка програмних продуктів

ОСОБЛИВОСТІ ТЕСТУВАННЯ ІГРОВИХ ПРОГРАМНИХ ПРОДУКТІВ

Поліщук Артем

студент 3-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Індустрія відеоігор сьогодні – одна з найприбутковіших галузей розважального програмного забезпечення. Ігри стали звичною частиною життя мільйонів людей: від простих мобільних застосунків до великих мережевих проєктів з мільйонами одночасних гравців. За даними галузевих досліджень, обсяг світового ринку відеоігор у 2025 році сягнув 18.8 мільярдів доларів США – це більше, ніж кіно- та музична індустрії разом узяті [1]. За такої популярності розробникам доводиться приділяти серйозну увагу якості та надійності своїх продуктів.

Тестування програмного забезпечення – це процес, під час якого проводяться перевірки для виявлення помилок та дефектів у програмі. Воно дає змогу переконатися, що програмний продукт працює коректно, відповідає вимогам і очікуванням користувачів, а також працює надійно та безпечно.

Тестування ігрових застосунків є специфічним напрямком забезпечення якості, який суттєво відрізняється від тестування класичних бізнес-застосунків. Головна відмінність полягає в тому, що ігри мають інтерактивну природу та комбінують у собі множину різнорідних систем, які повинні працювати злагоджено в реальному часі. На відміну від звичайних застосунків, де користувач виконує заздалегідь визначені сценарії, гравці в ігровому застосунку можуть діяти непередбачувано, експериментувати з механіками та шукати нестандартні способи взаємодії [2].

Дослідник Роберт Брайант у фундаментальній праці «Game Testing All in One» виділяє декілька ключових особливостей, які відрізняють тестування ігор від тестування іншого програмного забезпечення [3]. По-перше, ігри є event-driven системами з безперервним циклом оновлення, тому тестувальник повинен враховувати не лише правильність виконання окремих дій, а й коректність переходів між станами гри. По-друге, ігровий процес включає елемент випадковості (генерація рівнів, поведінка штучного інтелекту, рандомні події), що ускладнює відтворення дефектів. По-третє, оцінка якості ігри включає суб'єктивні аспекти: ігровий баланс, цікавість, відчуття від керування – те, що неможливо перевірити автоматизованими тестами.

Ніколіна Фінська у книзі «Modern Game Testing» зазначає, що процес тестування ігор традиційно поділяється на декілька основних фаз [4]. На першому етапі проводиться функціональне тестування, яке перевіряє всі ігрові механіки, системи управління, інтерфейсу користувача та основні сценарії

проходження. Другий етап – тестування сумісності, під час якого перевіряється коректність роботи гри на різних апаратних конфігураціях, операційних системах та з різною периферією. Третій етап – навантажувальне та стрес-тестування, особливо актуальне для багатокористувацьких онлайн-ігор. Четвертий етап – тестування локалізації для версій, що випускаються різними мовами. На фінальному етапі проводиться приймальне тестування, що підтверджує готовність продукту до релізу.

Тестування ігрових застосунків має охоплювати як звичайні, так і граничні сценарії використання, оскільки гравці часто знаходять нестандартні способи взаємодії з ігровою системою – це явище отримало назву «edge case testing» [3]. Особливу увагу слід приділяти мережевим аспектам гри, її стабільності при різних умовах з'єднання, обробці втрати пакетів, високої затримки та інших мережевих аномалій. Тестування інтерфейсу повинно враховувати ергономіку та зручність використання, а не лише функціональну коректність.

Окремою особливістю ігрового тестування є необхідність регулярних регресійних перевірок після кожного оновлення. На відміну від класичних застосунків, ігри отримують патчі та доповнення значно частіше – іноді щотижня. Кожне нове оновлення може порушити функціональність, яка працювала раніше. Тому в ігровій індустрії регресійне тестування набуває критичного значення і часто автоматизується.

Список використаних джерел

1. Global Games Market Report 2025. *Newzoo*. URL: https://investgame.net/wp-content/uploads/2025/09/2025_Newzoo_Free_Global_Games_Market_Report.pdf
2. Трофименко О.Г., Дика А.І., Задерейко О.В., Шевченко О.В. Сфери застосування штучного інтелекту в розробці та тестуванні ігор. *Кібербезпека: освіта, наука, техніка*. 2025. № 1(29). С. 41-59. DOI: https://doi.org/10.28925/2663-4023.2025.29.832
3. Bryant R. Game Testing All in One. 4th ed. Berlin, Boston: Mercury Learning and Information, 2024. 488 p.
4. Finska N. Modern Game Testing: Learn how to test games like a pro, optimize testing effort, and skyrocket your QA career. Birmingham: Packt Publishing, 2022. 296 p.

Ключові слова: тестування, ігровий застосунок, розробка ігор

Keywords: testing, game application, game development

Науковий керівник: доцент Трофименко О.Г.

ПОРІВНЯННЯ БАГАТОПЛАТФОРМОВИХ ІГРОВИХ РУШІЇВ UNITY 6, UNREAL ENGINE 5.7 ТА GODOT 4.6.3

Задерейко Олександр

кандидат технічних наук, доцент,
доцент кафедри штучного інтелекту та математичного моделювання
Національного університету «Одеська юридична академія»

Любич Назар

студент 3-го курсу факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

На сучасному етапі розвитку індустрії комп'ютерних ігор розробники-початківці все більш дають перевагу багатоплатформенним ігровим рушіям, програмним середовищам, які допомагають уникнути написання базового функціоналу з самого початку. Вони беруть на себе рендеринг графіки, обробку фізики, керування звуком, оптимізацію пам'яті, генерацію світла, сценарну логіку тощо. Саме вибір універсального рушія значною мірою впливає на складність розробки, продуктивність гри та якість кінцевого продукту. Найпопулярнішими рушіями є Unity 6, Unreal Engine 5.7 та Godot 4.6.3 . Кожен із них має свої особливості, переваги та обмеження, що впливають на вибір технології для створення комп'ютерних ігор [1].

Unity є одним із найвідоміших ігрових рушіїв у світі. Його розробка була розпочата компанією Unity Technologies у 2004 році, а саме перша версія була представлена у 2005 році. Спочатку рушій створювався як простий інструмент для розробки ігор під операційну систему macOS, однак згодом перетворився на потужну багатоплатформову систему для створення не тільки 2D-, а й 3D-ігор.

Unity 6 є сучасною версією рушія, що підтримує велику кількість платформ, серед яких Windows, Linux, Android, iOS, WebGL, PlayStation та Xbox. Основною мовою програмування в Unity є C#, яка поєднує простоту синтаксису та високу функціональність. Рушій має зручний графічний інтерфейс, що дозволяє швидко створювати сцени, налаштовувати фізику, анімацію та освітлення.

Однією з ключових переваг Unity є універсальність. Рушій активно використовується для створення мобільних ігор, інді-проєктів, VR/AR-застосунків та навчальних симуляторів. Також Unity має перевагу в кількості навчального матеріалу, що дає змогу початківцям ефективно вивчати основи геймдеву завдяки онлайн курсам, документації та відеоурокам. Крім того, Unity має великий магазин ресурсів Asset Store, де розробники можуть отримувати готові моделі, текстури, скрипти та інші компоненти для пришвидшення розробки. Недоліком Unity є те, що при створенні масштабних AAA-проєктів рушій може поступатися за якістю графіки та рівнем оптимізації, на відміну від Unreal Engine. Також деякі складні графічні ефекти потребують значних апаратних ресурсів [2].

Unreal Engine є одним із найстаріших та найпотужніших ігрових рушіїв сучасності. Його створення розпочалося компанією Epic Games у 1995 році, а саме перша версія рушія була представлена у 1998 році разом із грою Unreal. Від самого початку рушій орієнтувався на створення високоякісних 3D-ігор із сучасною графікою.

Сучасна версія Unreal Engine 5.7 використовує передові технології рендерингу, серед яких Nanite та Lumen. Технологія Nanite дозволяє працювати з великою кількістю деталізованих моделей без значної втрати продуктивності, а Lumen забезпечує глобальне освітлення у реальному часі. Завдяки цьому Unreal Engine широко застосовується для створення AAA-ігор, кінематографічних сцен, VR-проєктів та навіть візуалізації архітектури.

Основною мовою програмування в Unreal Engine є C++, однак рушій також містить систему візуального програмування Blueprint, яка дозволяє створювати логіку гри без написання великої кількості коду. Це робить рушій більш доступним для дизайнерів та початківців.

Перевагою Unreal Engine є високий рівень графіки та продуктивності. Саме цей рушій часто використовується у великих ігрових студіях для створення сучасних блокбастерів. Проте Unreal Engine має високі системні вимоги та складніший процес освоєння порівняно з Unity і Godot. Для ефективної роботи з рушієм необхідно мати достатній рівень знань програмування та потужне комп'ютерне обладнання [3].

Godot є відносно молодим ігровим рушієм із відкритим вихідним кодом. Його створення було розпочато аргентинськими розробниками Juan Linietsky та Ariel Manzur у 2001 році, а у відкритий доступ рушій було випущено у 2014 році. На відміну від Unity та Unreal Engine, Godot поширюється безкоштовно та не потребує сплати ліцензійних внесків.

Godot 4.6.3 підтримує створення 2D- та 3D-ігор і використовує власну мову програмування GDScript, синтаксис якої нагадує Python. Також рушій підтримує C# та C++. Однією з головних особливостей Godot є його невеликий розмір та висока швидкість запуску.

Рушій має простий інтерфейс та добре підходить для навчання основам геймдеву. Godot активно використовується незалежними розробниками та невеликими командами для створення інді-ігор. Крім того, відкритий код дозволяє змінювати рушій відповідно до потреб проєкту. Недоліком Godot є менш розвинена екосистема порівняно з Unity та Unreal Engine. Також рушій має обмежені можливості при створенні великих AAA-проєктів із фотореалістичною графікою. Незважаючи на це, популярність Godot постійно зростає завдяки активній підтримці спільноти та розвитку відкритого програмного забезпечення [4].

Отже, Unity є універсальним рішенням, яке поєднує простоту використання та широкі можливості багатоплатформової розробки. Unreal Engine забезпечує найвищий рівень графіки та продуктивності, що робить його оптимальним

вибором для AAA-проєктів. Godot, у свою чергу, є перспективним відкритим рушієм для навчання, інді-розробки та невеликих ігор.

Таким чином, вибір ігрового рушія залежить від типу проєкту, вимог до графічної складової, бюджету, складності розробки та рівня підготовки розробника чи команди розробників. Для мобільних ігор та інді-проєктів доцільним є використання Unity або Godot, тоді як для створення масштабних AAA-ігор найбільш ефективним рішенням є Unreal Engine. Це свідчить, що сучасні ігрові рушії продовжують активно розвиватися та відіграють важливу роль для всієї індустрії комп'ютерних ігор.

Список використаних джерел

1. Толокнов А. А. Огляд еволюції архітектури ігрових рушіїв [Електронний ресурс] / А. А. Толокнов. – 2025. – Режим доступу: https://dspace.onua.edu.ua/server/api/core/bitstreams/6e5137a2-af39-48d2-8f2a-b1d27925f13d/content – Дата звернення: 15.05.2026.
2. Unity Technologies. Unity Documentation [Електронний ресурс]. – Режим доступу: https://docs.unity3d.com/ – Дата звернення: 15.05.2026.
3. Epic Games. Unreal Engine Documentation [Електронний ресурс]. – Режим доступу: https://dev.epicgames.com/documentation/en-us/unreal-engine/ – Дата звернення: 15.05.2026.
4. Godot Engine. Official Documentation [Електронний ресурс]. – Режим доступу: https://docs.godotengine.org/ – Дата звернення: 15.05.2026.

Ключові слова: ігровий рушій, Unity, Unreal Engine, Godot, розробка ігор, багатоплатформовість, 3D-графіка

Keywords: game engine, Unity, Unreal Engine, Godot, game development, cross-platform, 3D graphics

ГЕНЕРАТИВНІ МОДЕЛІ У КОНТЕКСТІ РОЗРОБКИ ІГОР

Трофименко Олена

кандидат технічних наук, доцент,
доцент кафедри програмної інженерії
Національного університету «Одеська юридична академія»

Ігрова індустрія сьогодні є одним із найбільш динамічних і технологічно розвинених сегментів цифрової економіки. Світовий ринок відеоігор демонструє стабільне зростання, а за прогнозами аналітиків до 2033 року його обсяг може перевищити 503 млрд доларів США [1]. Одним із ключових чинників такого розвитку є активне впровадження технологій штучного інтелекту (ШІ), зокрема генеративних моделей, у процеси створення та супроводу ігрових продуктів.

Сучасні ігри характеризуються високою складністю ігрових механік, значними обсягами контенту, відкритими світами та потребою у персоналізованому користувацькому досвіді. За таких умов традиційні підходи до розробки стають

дедалі більш ресурсомісткими, що стимулює використання інтелектуальних методів автоматизації. Штучний інтелект у геймдеві застосовується для моделювання поведінки неігрових персонажів (NPC), процедурної генерації контенту, тестування ігрових механік, балансування складності, аналізу поведінки гравців та оптимізації процесів розробки [2].

Особливе місце серед сучасних технологій ШІ займають генеративні моделі – клас алгоритмів машинного навчання, здатних створювати нові дані на основі закономірностей, виявлених у навчальній вибірці. На відміну від дискримінативних моделей, що орієнтовані на класифікацію або прогнозування, генеративні моделі апроксимують розподіл даних (p(x)) або умовний розподіл (p(x|y)), що дозволяє синтезувати нові об'єкти, подібні до реальних [3].

У контексті розробки відеоігор генеративні моделі відкривають широкі можливості для автоматизації створення контенту. Їх використання дозволяє генерувати:

- ігрові локації та рівні;
- текстури, 2D- та 3D-моделі;
- анімації персонажів;
- діалоги та сюжетні гілки;
- музичний і звуковий супровід;
- поведінкові сценарії NPC;
- адаптивні ігрові події залежно від стилю гри користувача.

Одним із найпоширеніших напрямів застосування є процедурна генерація контенту (Procedural Content Generation, PCG), де генеративний ШІ дозволяє створювати великі унікальні ігрові світи з мінімальним втручанням розробника. Це особливо актуально для ігор жанрів RPG, sandbox та survival, де масштаб середовища та варіативність подій безпосередньо впливають на якість користувацького досвіду.

Серед сучасних генеративних архітектур найбільше практичне значення для геймдеву мають: генеративно-змагальні мережі (GAN), варіаційні автоенкодери (VAE), авторегресивні та дифузійні моделі (рис. 1).

Серед сучасних генеративних архітектур найбільше практичне значення для геймдеву мають генеративно-змагальні мережі (GAN), варіаційні автоенкодери (VAE), авторегресивні та дифузійні моделі. Кожен із цих підходів відрізняється способом апроксимації розподілу даних, механізмом навчання та процедурою генерації нових зразків [5]. GAN-моделі ефективно застосовуються для генерації текстур і стилізованих зображень, тоді як VAE дозволяють створювати варіативні представлення ігрових об'єктів. Дифузійні моделі забезпечують високоякісний синтез графічного контенту, а трансформерні архітектури – генерацію діалогів та поведінкових сценаріїв NPC [4].

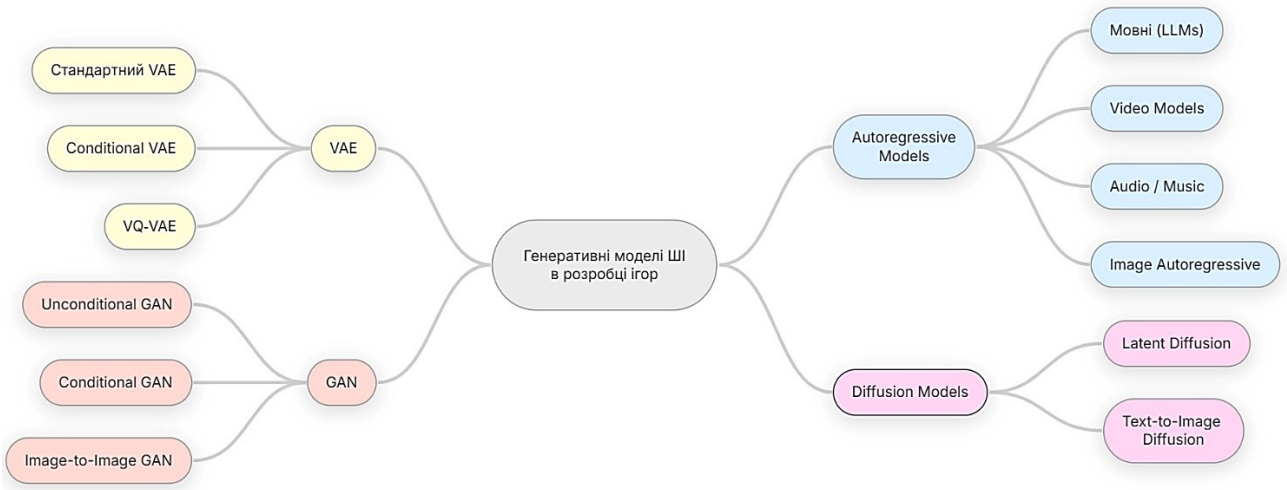


Рисунок 1. Класифікація генеративних моделей ШІ, що використовуються в розробці ігор

Окремий інтерес становить інтеграція генеративних моделей із reinforcement learning та системами поведінкового аналізу гравців. Це дозволяє створювати адаптивні ігрові середовища, які динамічно змінюють рівень складності, сюжет або поведінку персонажів залежно від дій користувача. Подібний підхід сприяє формуванню персоналізованого ігрового досвіду та підвищенню рівня залучення гравців.

Попри значні переваги, використання генеративного ШІ у розробці ігор супроводжується низкою проблем. Серед основних викликів можна виділити:

- високі обчислювальні витрати на навчання моделей;
- потребу у великих наборах якісних даних;
- складність контролю якості згенерованого контенту;
- ризики повторюваності або семантичної некоректності результатів;
- етичні та правові питання авторства AI-контенту.

Крім того, надмірна автоматизація процесу розробки може негативно впливати на художню унікальність ігор, оскільки генеративні системи здебільшого формують контент на основі вже існуючих шаблонів та патернів.

Отже, генеративні моделі стають одним із ключових технологічних напрямів розвитку сучасної ігрової індустрії. Їх використання дозволяє суттєво оптимізувати процес створення контенту, зменшити витрати на розробку та забезпечити високий рівень персоналізації ігрового досвіду. Водночас ефективне впровадження таких технологій потребує комплексного підходу, що поєднує алгоритмічну ефективність, контроль якості результатів та збереження творчої складової процесу розробки відеоігор.

Список використаних джерел

1. Global video game market value from 2022 to 2032 (in billion U.S. dollars). Statista. URL: https://www.statista.com/statistics/292056/video-game-market-value-worldwide/

2. Трофименко О.Г., Дика А.І., Лобода Ю.В., Толокнов А.А., Бондаренко М.Є. Штучний інтелект у розробці ігор. Кібербезпека: освіта, наука, техніка. 2025. № 3(27). С. 109–119. DOI: https://doi.org/10.28925/2663-4023.2025.27.701
3. Aydın A., Surer E. Using Generative Adversarial Nets on Atari Games for Feature Extraction in Deep Reinforcement Learning. arXiv. 2020. DOI: https://doi.org/10.48550/arXiv.2004.02762
4. Yang Z. Research on the Application of AI Generative Models in Games. Proceedings of the 2025 3rd International Conference on Image, Algorithms, and Artificial Intelligence (ICIAAI 2025). Singapore, 2025. P. 1005–1015. DOI: https://doi.org/10.2991/978-94-6463-823-3_98
5. Bond-Taylor S., Leach A., Long Y., Willcocks Ch. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. arXiv. 2022. DOI: https://doi.org/10.48550/arXiv.2103.04922

Ключові слова: ШІ, процедурна генерація контенту, розробка ігор, генеративно-змагальні мережі (GAN), варіаційні автоенкодери (VAE), авторегресивні моделі, дифузійні моделі

Keywords: AI, procedural content generation, game dev, generative adversarial networks (GAN), variational autoencoders (VAE). autoregressive models, diffusion models

РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ ПРОГРАМНИМИ ЗАВДАННЯМИ НА ОСНОВІ DJANGO

Табенський Сергій

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія

Сучасний освітній процес у галузі програмування потребує ефективних засобів автоматизації перевірки практичних завдань. Традиційна ручна перевірка програмного коду займає значний час викладача, особливо при великій кількості студентів та регулярному виконанні лабораторних робіт. Використання спеціалізованих систем управління програмними завданнями дозволяє автоматизувати процес оцінювання, забезпечити швидкий зворотний зв’язок та підвищити ефективність навчання [1, 2].

Одним із перспективних підходів є використання веборієнтованих систем, що забезпечують централізований доступ до навчальних матеріалів, завдань та результатів перевірки. Такі системи дозволяють інтегрувати механізми автоматичного тестування програмного коду, управління курсами та аналітики успішності.

Метою роботи є розробка системи управління програмними завданнями з підтримкою автоматичної перевірки рішень студентів на основі фреймворку Django.

Система реалізована у вигляді багатомодульного Django-застосунку, що складається з трьох основних функціональних модулів: accounts, courses та checker. Модуль accounts відповідає за автентифікацію користувачів та систему

ролей, courses – за управління курсами й завданнями, а checker – за перевірку програмного коду студентів (рис. 1).

Для зберігання даних використано СКБД PostgreSQL. Взаємодія з базою даних здійснюється через Django ORM, що забезпечує об'єктно-реляційне відображення та автоматичний захист від SQL-ін'єкцій завдяки параметризованим SQL-запитам.

У системі реалізовано низку ORM-запитів для підтримки функціональності застосунку. Основними серед них є:

- отримання користувачів за логіном;
- створення нових курсів та завдань;
- приєднання студентів до курсу за кодом запрошення;
- отримання історії спроб виконання завдань;
- формування аналітики успішності студентів.

Рисунок 1 – Панель Django адміністрування

Для оптимізації роботи з базою даних використано метод *select_related()*, який дозволяє уникнути проблеми *N+1* запитів. Наприклад, при отриманні списку *submissions* разом із даними студента виконується один JOIN-запит замість окремих звернень до таблиці користувачів для кожного запису.

Також у системі застосовано принцип lazy evaluation, відповідно до якого QuerySet виконується лише під час фактичного звернення до даних. Це дозволяє мінімізувати кількість SQL-запитів та зменшити навантаження на базу даних [3].

Серверна частина системи реалізована мовою Python із використанням фреймворку Django. Одним із ключових рішень стало розширення стандартної моделі користувача AbstractUser шляхом додавання ролей «викладач», «студент» та «адміністратор». Це забезпечило розмежування доступу до функціональних можливостей системи.

Для створення курсів реалізовано механізм генерації унікального *invite_code*, який використовується студентами для приєднання до навчального курсу. Генерація коду виконується випадковим чином із використанням великих літер та цифр.

Особливу увагу приділено реалізації модуля автоматичної перевірки програмного коду *checker/runner.py*. Даний модуль забезпечує:

- створення тимчасової директорії;
- запис коду студента у файл solution.py;
- генерацію тестового файлу на основі pytest;
- запуск перевірки через subprocess.run();
- аналіз результатів тестування;
- обробку виключень та таймаутів.

Код студента запускається у тимчасовому середовищі, що підвищує безпеку виконання. Для запобігання зависанню системи використовується *timeout*, після перевищення якого процес примусово завершується.

Після завершення тестування система автоматично визначає статус рішення. Якщо всі тести пройдено успішно, студент отримує максимальний бал. У випадку часткового проходження тестів оцінка розраховується пропорційно кількості успішних перевірок.

Для захисту серверної частини використано декоратор *@login_required* та перевірки ролей користувачів. Доступ до аналітики, створення курсів і завдань мають лише викладачі, тоді як студенти можуть переглядати власні результати та надсилати рішення.

Клієнтська частина системи реалізована за допомогою Django Templates та Bootstrap 5. Використання серверного рендерингу дозволило спростити архітектуру застосунку та зменшити залежність від JavaScript.

Усі сторінки наслідуються від базового шаблону *base.html*, який містить загальну структуру інтерфейсу: навігаційну панель, стилі та механізм flash-повідомлень. Такий підхід забезпечує дотримання принципу DRY та спрощує підтримку проєкту.

Інтерфейс системи адаптується до ролі користувача. Викладачі мають доступ до створення курсів, перегляду тест-коду та аналітики, тоді як студенти бачать лише доступні завдання та власні результати.

Для введення програмного коду інтегровано редактор CodeMirror 5 із підтримкою: підсвітки синтаксису Python; нумерації рядків; автоматичних відступів; перевірки дужок; темної теми Dracula.

Перед надсиланням форми використовується метод *editor.save()*, який синхронізує вміст редактора з HTML-формою. Це дозволяє коректно передавати код студента на сервер.

Результати перевірки відображаються на окремій сторінці *submission_detail.html*. Користувач отримує інформацію про: статус виконання;

кількість пройдених тестів; набраний бал; повідомлення pytest; власний програмний код (рис. 2).

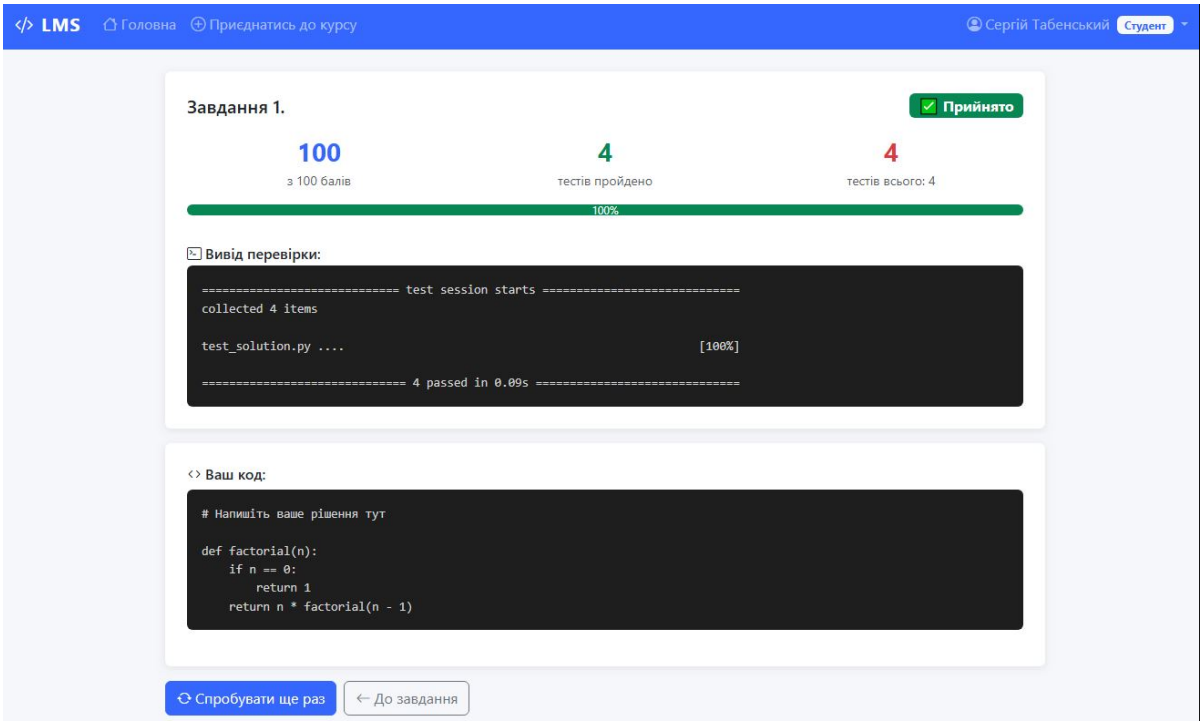


Рисунок 2 – Результат перевірки завдання

Для візуалізації прогресу використовується *Bootstrap progress-bar*, а вивід *pytest* відображається у спеціальному блоці.

У результаті роботи було розроблено систему управління програмними завданнями на основі Django та PostgreSQL, яка забезпечує автоматизацію процесів створення курсів, видачі завдань, перевірки програмного коду та аналізу результатів навчання.

Список використаних джерел:

1. 2025 Global EdTech 1000. URL: https://www.holoniq.com/notes/2025-global-edtech-1000
2. Jung, Yeonji, Lee Yunseo, Bae Jiyeong, Kim DoYong, Choi Heungsoo, Kang Minji, Lee Unggi. Exploring the Role of Automated Feedback in Programming Education: A Systematic Literature Review. 2026. 10.48550/arXiv.2602.00089
3. Ling, Chen & Wang, Yachen. (2025). Lazy Evaluation: A Comparative Analysis of SAS MACROs and R Functions. Conference: SESUG 2025At: Cary, NC. URL: https://www.researchgate.net/publication/397973325_Lazy_Evaluation_A_Comparative_Analysis_of_SAS_MACROs_and_R_Functions

Ключові слова: система управління програмними завданнями, Django, PostgreSQL, Python, аналіз успішності студентів, перевірка програмного коду

Key words: programming task management system, Django, PostgreSQL, Python, student performance analysis, program code verification

Науковий керівник: доцент Логінова Н.І.

РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ ВІДГУКІВ

Чикунов Павло

кандидат технічних наук, доцент, доцент кафедри програмної інженерії Національного університету «Одеська юридична академія»

Тіліжинський Владислав

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Текстові відгуки пасажирів є одним із джерел інформації для оцінювання якості сервісу авіакомпаній, формування їхнього іміджу та підтримки управлінських рішень. Значні обсяги неструктурованих даних, що надходять із веб-ресурсів, зумовлюють потребу у створенні програмних засобів, здатних автоматизувати процеси збору, зберігання та аналізу таких відгуків [1]. Особливої актуальності набуває використання вебскрапінгу у поєднанні з методами обробки природної мови для отримання узагальненої аналітики щодо тональності та тематики повідомлень користувачів [2].

Метою дослідження є проєктування та реалізація серверної частини інформаційної системи для вебскрапінгу та аналізу відгуків клієнтів авіакомпаній на основі технологій ASP.NET Core. Об’єктом дослідження є процес вебскрапінгу та аналізу текстових відгуків, а предметом – методи створення серверних компонентів для роботи з неструктурованими даними [3].

Архітектура серверної частини передбачає розробку модуля вебскрапінгу, призначеного для збору відгуків з відкритих веб-джерел. Модуль функціонує в межах загальної архітектури інформаційної системи та взаємодіє з іншими її частинами через інтерфейси (рис. 1).

Модуль забезпечує збір відгуків з відкритих платформ Skytrax, AirlineRatings та TripAdvisor, виконання обробки та підготовці даних для NLP-аналізу, визначення тональності відгуків, формування аналітики (рис. 2).

Сформовано функціональні та нефункціональні вимоги, розроблено модель даних із ключовими сутностями та спроєктовано архітектуру системи на основі RESTful API і шаблону Repository-Service-Controller.

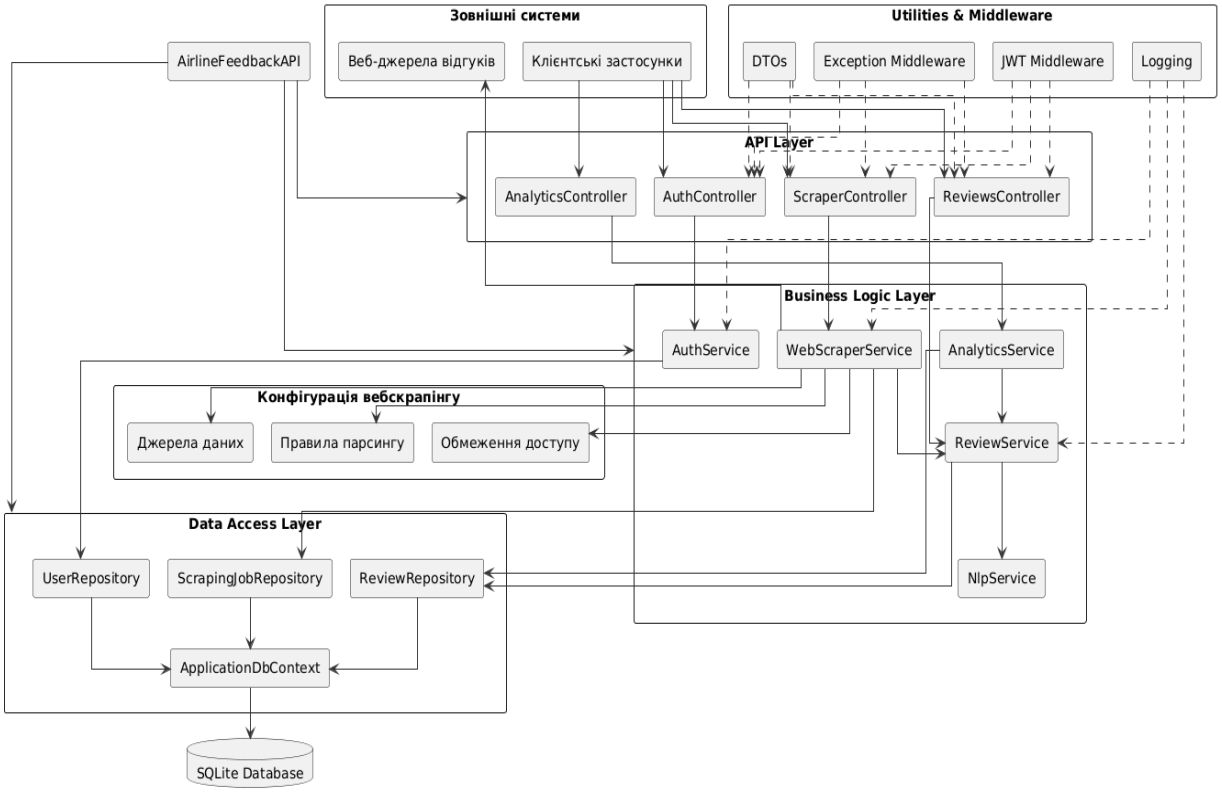


Рисунок 1 – Діаграма компонентів серверної частини

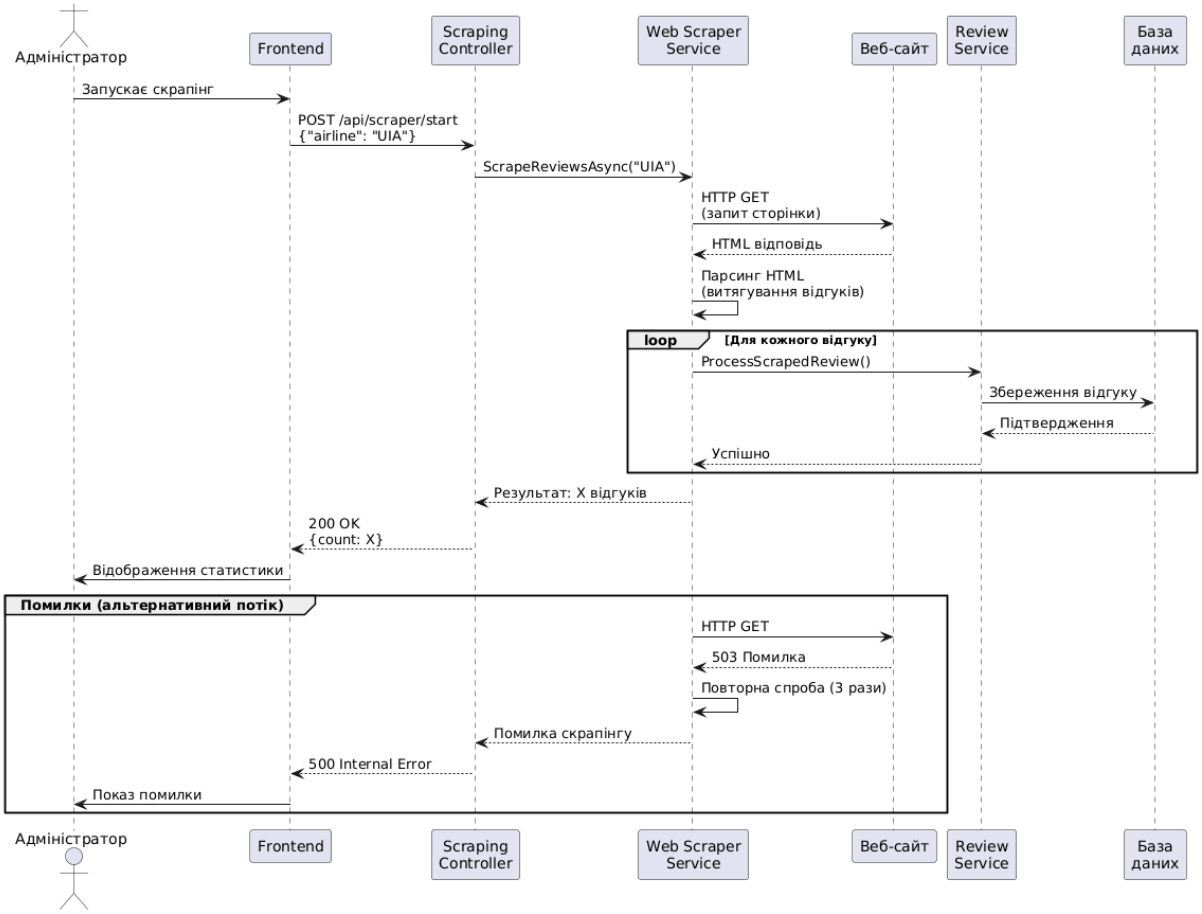


Рисунок 2 – Діаграма послідовності процесу вебскрапінгу

Виконана реалізація серверної частини, включно з базою даних на основі SQLite та Entity Framework Core, REST-ендпоінтами для управління відгуками та аутентифікації користувачів (рис. 3), модулем вебскрапінгу та NLP-аналізом тональності й тематики відгуків. Для забезпечення надійності та коректності роботи реалізовано функціональне та інтеграційне тестування модуля вебскрапінгу та REST-ендпоінтів. Тестування включає перевірку циклу обробки даних: отримання HTML-контенту з веб-ресурсу або локального файлу, виділення структурованих відгуків, конвертацію їх у внутрішні моделі, збереження у БД.

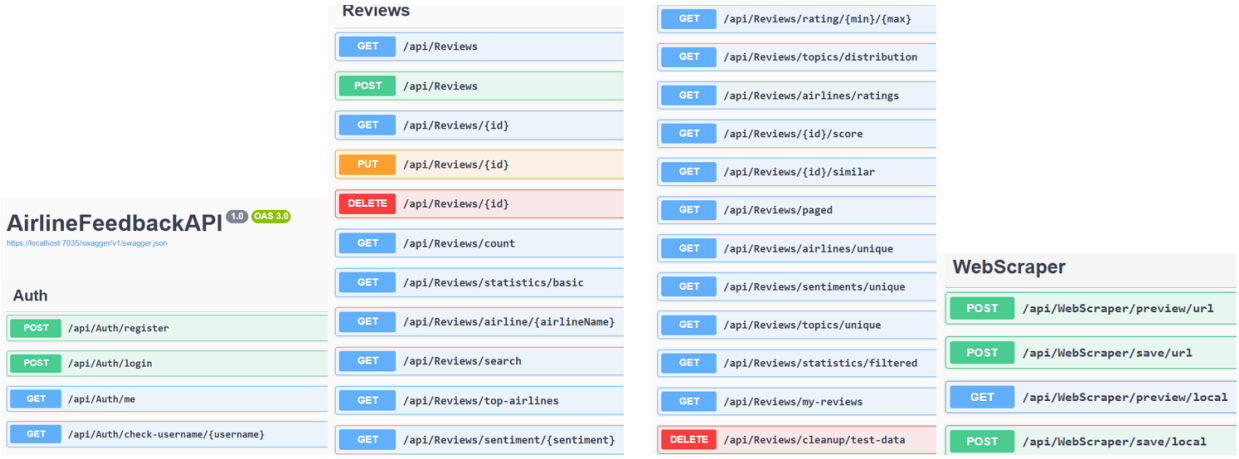


Рисунок 3 – Ендпоінти глобального та локального вебскрапінгу

Висновки. Практична значущість роботи полягає в тому, що серверна частина може бути інтегрована в наявні інформаційні системи авіакомпаній для оцінки сервісу, виявлення проблемних аспектів обслуговування та прийняття рішень щодо підвищення рівня задоволеності пасажирів.

Список використаних джерел

1. Kumar R., Singh S. Sentiment analysis of airline reviews using deep learning approaches. *Journal of Air Transport Management*, 2024.
2. Das P., Roy S. Combining web scraping with NLP techniques for airline customer review analysis. *Applied Soft Computing*, 2022.
3. Сідак К.І., Ліхоузова Т.А. Моделі для аналізу відгуків пасажирів авіакомпаній. *Наука, освіта, технології і суспільство в умовах глобалізації*.. Біла Церква, 2023. Ч. 2, с. 25–28

Ключові слова: відгуки клієнтів, вебскрапінг, аналіз тексту, NLP, REST API, серверна частина, SQLite, Entity Framework Core

Keywords: customer reviews, web scraping, text analysis, NLP, REST API, backend, SQLite, Entity Framework Core

Підсекція 5. Інформаційні системи та прикладні рішення

СИСТЕМА ІМІТАЦІЙНОГО АНАЛІЗУ ЛОГІСТИЧНОЇ КОМПАНІЇ «ПОШТА»

Бугеря Кирило

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Сучасні логістичні компанії все частіше використовують вебтехнології для автоматизації процесів доставки, контролю стану відправлень та аналізу ефективності маршрутів. Зростання обсягів електронної комерції зумовлює потребу у програмних системах, які можуть не лише зберігати інформацію про посилки, а й моделювати процес їх переміщення між пунктами доставки. Тому актуальним є створення системи, що поєднує облік відправлень, побудову маршрутів, симуляцію руху транспорту та візуалізацію логістичних процесів.

Метою роботи є розробка вебзастосунку "Пошта", призначеного для імітаційного аналізу роботи логістичної компанії. Розроблена система дозволяє створювати посилки, виконувати автентифікацію користувачів, розраховувати вартість доставки, запускати рейси адміністратором, будувати маршрут автомобіля та відображати результати виконання доставки в адміністративній панелі.

Вебзастосунок побудовано за клієнт-серверною архітектурою. Клієнтська частина забезпечує взаємодію користувача з системою через вебінтерфейс, а серверна частина відповідає за обробку запитів, роботу з базою даних, зміну статусів посилок та виконання симуляції. Для реалізації серверної логіки використано Java та Spring Boot, для зберігання даних – PostgreSQL, а для побудови інтерфейсу – HTML, CSS, JavaScript, Thymeleaf і Bootstrap. Візуалізація маршрутів може виконуватися за допомогою бібліотеки Leaflet.js.

Приклади сторінок реєстрації та входу користувача до системи наведено на рис. 1.

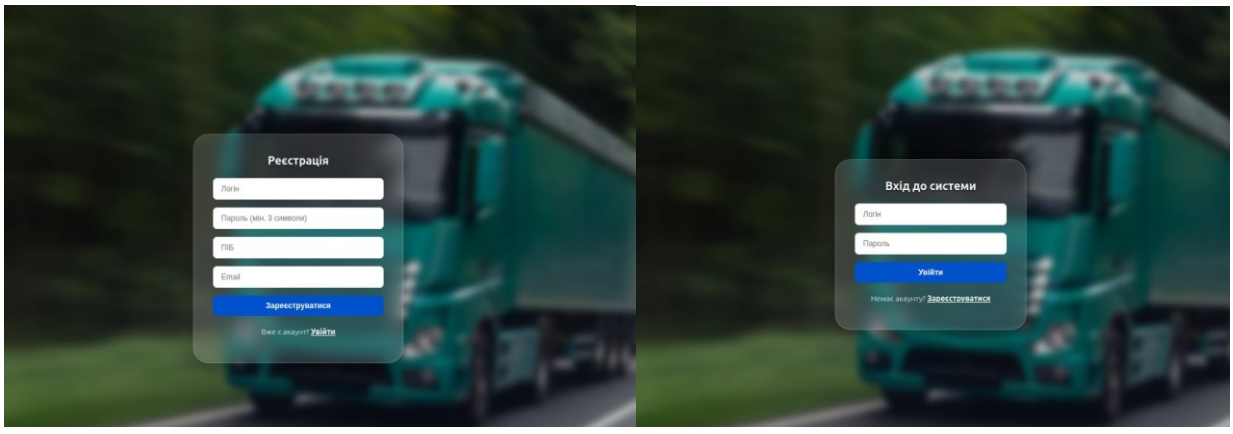


Рисунок 1 – Сторінки реєстрації та входу користувача

У системі передбачено розмежування прав доступу між звичайним користувачем та адміністратором. Звичайний користувач після реєстрації і входу отримує можливість створювати посилки та переглядати власні відправлення. Адміністратор має доступ до логістичного центру, де можна керувати посилками, призначати автомобілі та запускати рейси.

Однією з основних функцій вебзастосунку є створення нової посилки. Користувач обирає пункт відправлення, пункт призначення та вводить вагу посилки. Після цього система автоматично розраховує вартість доставки. Розрахунок базується на умовній відстані між філіями та вазі відправлення. Такий підхід дозволяє швидко оцінити вартість перевезення без використання сторонніх сервісів.

Приклад форми створення нової посилки та розрахунку вартості доставки наведено на рис. 2.

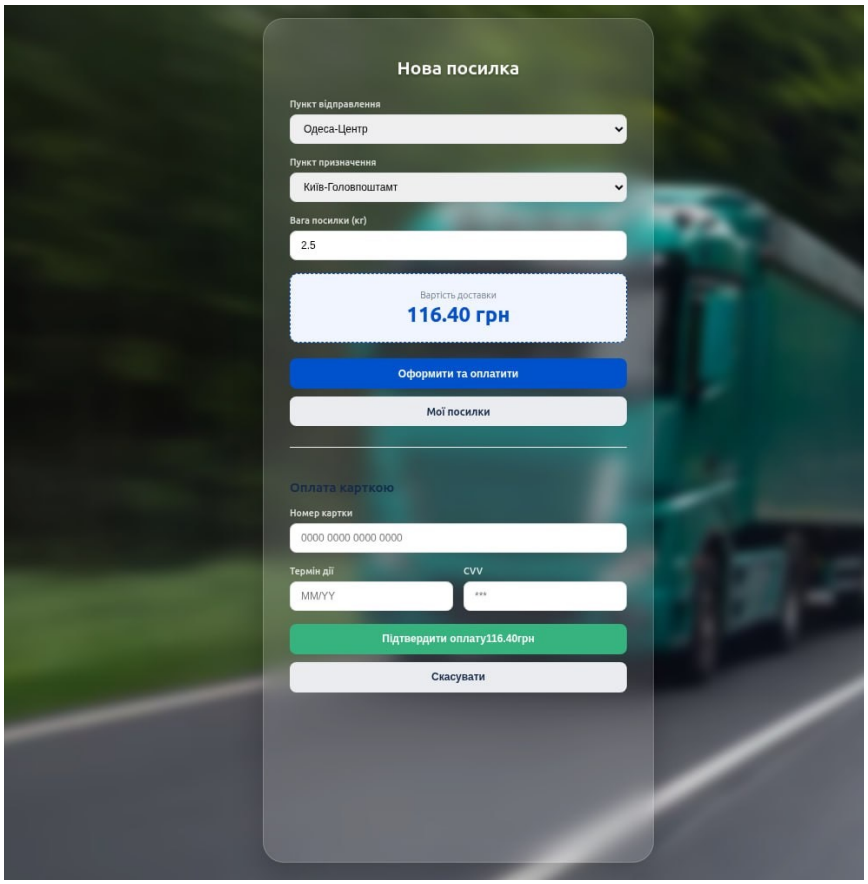


Рисунок 2 – Форма створення нової посилки

Після створення посилки система формує її унікальний номер, встановлює початковий статус CREATED і зберігає запис у базі даних. Надалі під час виконання рейсу статус посилки змінюється на IN_TRANSIT, а після прибуття до пункту призначення – на DELIVERED. Для зручності контролю також відображається індивідуальний прогрес доставки.

Для побудови маршруту використовується жадібний алгоритм найближчого сусіда. Його принцип полягає у тому, що автомобіль на кожному кроці обирає найближчу доступну невідвідану точку маршруту. До переліку точок входять

пункти відправлення та призначення посилок, які обрані для рейсу. Якщо декілька посилок мають однакові філії, такі точки не дублюються, що дозволяє зменшити кількість зайвих зупинок.

Важливою частиною системи є симуляція руху транспорту. Після запуску рейсу система поступово оновлює загальний прогрес маршруту та прогрес кожної посилки. Під час прибуття автомобіля до пункту відправлення посилка переводиться у стан IN_TRANSIT, а після досягнення пункту призначення отримує статус DELIVERED. Це дозволяє відтворити основні етапи реального логістичного процесу в навчальній та демонстраційній формі.

Приклад відображення створених відправлень користувача наведено на рис. 3.

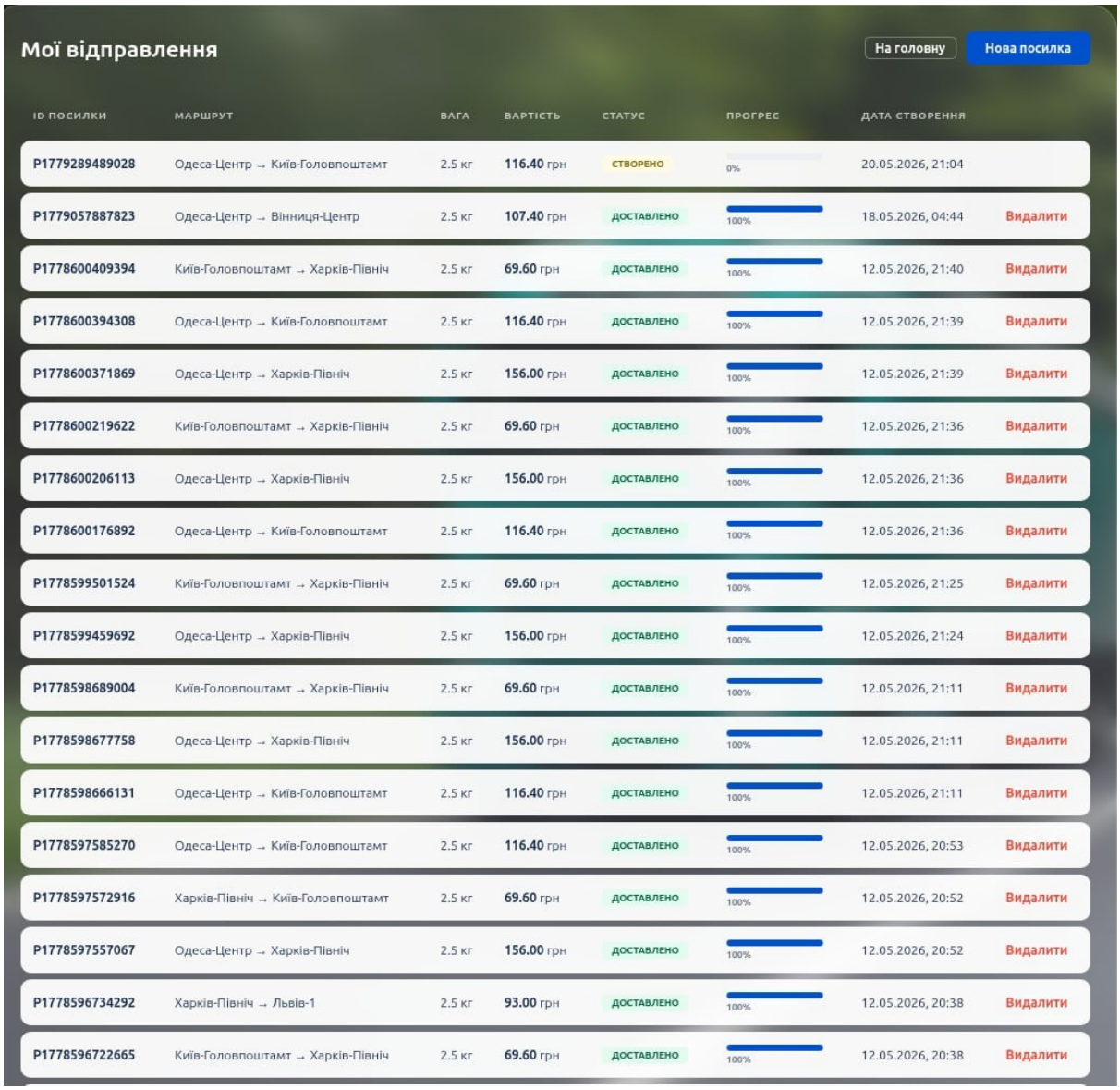


Рисунок 3 – Список відправлень користувача

Адміністративна частина вебзастосунку призначена для керування логістичними процесами. У ній адміністратор бачить посилки, що очікують відправлення, може вибрати автомобіль для рейсу та запустити доставку. Крім того, у

таблиці історії відображаються всі посилки, їхні маршрути, вартість, статуси, прогрес виконання, номер автомобіля і рейсу.

Розроблена адміністративна панель також надає можливість аналізувати стан автопарку. Для кожного автомобіля визначається статус FREE або BUSY, поточний рейс, список посилок у цьому рейсі та загальний прогрес виконання маршруту. Завдяки цьому адміністратор може контролювати роботу системи та оцінювати завантаженість транспорту.

Приклад адміністративної панелі керування відправленнями наведено на рис. 4.

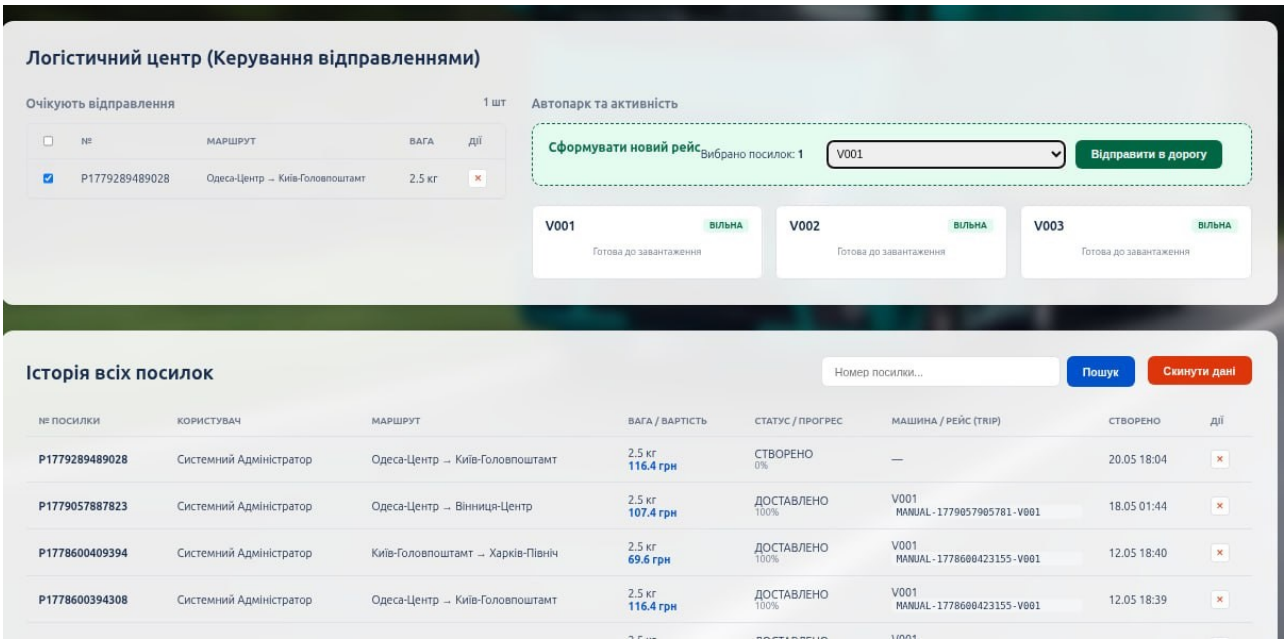


Рисунок 4 – Адміністративна панель керування відправленнями

Практична цінність розробленого вебзастосунку полягає у можливості демонстрації основних принципів роботи логістичної системи. Програма дозволяє простежити шлях посилки від моменту створення до завершення доставки, оцінити роботу алгоритму маршрутизації та побачити зміну статусів у процесі симуляції. Такий підхід може бути корисним для навчальних цілей, тестування логістичних сценаріїв і подальшого вдосконалення програмних рішень у сфері доставки.

Отже, у результаті розробки створено вебзастосунок "Пошта", який поєднує функції обліку посилок, розрахунку вартості доставки, адміністрування рейсів, маршрутизації та симуляції руху транспорту. Використання клієнт-серверної архітектури, реляційної бази даних та інтерактивного вебінтерфейсу забезпечує зручність роботи з системою і створює основу для її подальшого розвитку.

Список використаних джерел

1. Tho V. Le, Ruoling Fan. Digital Twins for Logistics and Supply Chain Systems: Literature Review, Conceptual Framework, Research Potential, and Practical Challenges. 2023. arXiv:2311.17317. DOI: https://doi.org/10.48550/arXiv.2311.17317

2. Danchuk V., Comi A., Weiß C., Svatko V. The optimization of cargo delivery processes with dynamic route updates in smart logistics. Eastern-European Journal of Enterprise Technologies. 2023. 2(3 (122)). C. 64–73. DOI: https://doi.org/10.15587/1729-4061.2023.277583
3. PostgreSQL. Official Documentation. URL: https://www.postgresql.org/docs/
4. Spring Boot Reference Documentation. URL: https://docs.spring.io/spring-boot/reference/
5. Leaflet Documentation. URL: https://leafletjs.com/reference.html
6. Leaflet Routing Machine Documentation. URL: https://www.liedman.net/leaflet-routing-machine/
7. Bootstrap Documentation. URL: https://getbootstrap.com/docs/

Ключові слова: логістика, вебзастосунок, симуляція доставки, маршрутизація, Spring Boot, Leaflet, PostgreSQL

Keywords: logistics, web application, delivery simulation, routing, Spring Boot, Leaflet, PostgreSQL

Науковий керівник: доцент Логінова Н.І.

ІНТЕГРАЦІЯ ХМАРНИХ СЕРВІСІВ ТА ШТУЧНОГО ІНТЕЛЕКТУ В АРХІТЕКТУРУ СИСТЕМ ДЛЯ ПЛАНУВАННЯ ПОДОРОЖЕЙ

Дирда Олександра

студентка 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Сфера туристичних інформаційних технологій (TravelTech) стрімко трансформується, переходячи від статичних систем бронювання до інтелектуальних персоналізованих помічників. Розробка сучасних додатків для планування подорожей стикається з проблемою обробки масивів різнорідних даних: текстових описів, географічних координат та мультимедійного контенту. Побудова монолітної системи, яка самостійно виконує всі ці обчислення, є економічно недоцільною та перевантажує серверну інфраструктуру. Оптимальним інженерним рішенням є делегування ресурсомістких завдань зовнішнім платформам через API-інтеграції, що дозволяє створити масштабовану та швидкодіючу систему.

Серцем логічного рівня застосунку є модуль генеративного штучного інтелекту, впроваджений за допомогою платформи Groq API. Використання інноваційних апаратних рішень (LPU – Language Processing Units) забезпечує надвисоку швидкість інференсу для відкритих мовних моделей, таких як Llama 3 та Mixtral [1]. У системі планування подорожей Groq API виконує роль інтелектуального рушія: аналізує критерії користувача (кількість мандрівників, вподобання, дати) та синтезує унікальний контент. Модель генерує індивідуальні назви турів, описи міст і формує макро-інсайти для пропозицій щодо локацій маршруту. Перевагою є строга типізація вихідних даних у форматі JSON, що дозволяє клієнтському

застосунку миттєво аналізувати та відображати складні структури без обробки на власному бекенді.

Управління медіаконтентом реалізовано через хмарний сервіс Cloudinary. Туристичні застосунки оперують графічними файлами та PDF-документами (електронні квитки, ваучери). Cloudinary функціонує як надійне сховище, що автоматично оптимізує ресурси (конвертує зображення у формати WebP або AVIF) без втрати візуальної якості [2]. Завдяки глобальній мережі доставки контенту (CDN), файли кешуються на найближчих до клієнта серверах, що суттєво зменшує час завантаження сторінок та економить пропускну здатність мережі.

Для динамічного візуального оформлення інтерфейсу без ручної праці контент-менеджерів архітектуру доповнено інтеграцією з Unsplash API, який надає доступ до бази професійних фотографій [3]. Після генерації маршруту система за допомогою штучного інтелекту виокремлює ключові теги (наприклад, «Венеція», «канали») та виконує фоновий REST-запит. Отримані зображення автоматично встановлюються як обкладинки поїздки, створюючи привабливий користувацький інтерфейс (UI) та підвищуючи рівень залученості.

Особливе місце займає підсистема геолокації. Замість дорогих рішень залучено стек відкритих геопросторових технологій OpenStreetMap (OSM Для зворотного геокодування (перетворення топонімів на координати) інтегровано інтерфейс Nominatim [4], а для інтелектуального геопросторового пошуку туристичних локацій (POI) задіяно потужний інструмент Overpass API. Це забезпечує точний і швидкий пошук локацій із багатомовною підтримкою. Отримані координати передаються до JavaScript-бібліотеки Leaflet, яка відповідає за рендеринг інтерактивних карт, відображення маркерів та побудову полігонів маршрутів у браузері [5].

З точки зору інформаційної безпеки, інтеграція зовнішніх сервісів реалізована через власний серверний проксі-рівень на базі Node.js. Приватні ключі доступу (API tokens) від Groq, Cloudinary та Unsplash зберігаються виключно у змінних оточення сервера, що повністю унеможливлює їх компрометацію на стороні клієнта. Безпека управління медіаконтентом забезпечується захищеними маршрутами завантаження, а для безпечного перегляду збережених PDF-документів та обходу обмежень політики CORS реалізовано спеціальний механізм буферизації бінарних даних безпосередньо на бекенді.

Отже, проєктування систем у сфері TravelTech базується на вдалій композиції хмарних сервісів та штучного інтелекту. Синергія генеративних моделей Groq, платформи Cloudinary, Unsplash та стеку OpenStreetMap дозволяє створити повноцінну, швидку і безпечну систему. Такий підхід скорочує час розробки, знижує витрати на інфраструктуру та гарантує гарний користувацький досвід.

Список використаних джерел

1. Groq Cloud Platform: Fast AI Inference. URL: https://console.groq.com/docs/

2. Cloudinary Documentation: Image and Video Upload, Storage, and Optimization. URL: https://cloudinary.com/documentation
3. Unsplash Developers: Official API Documentation. URL: https://unsplash.com/developers
4. OpenStreetMap Wiki: Nominatim Geocoding Tool. URL: https://wiki.openstreetmap.org/wiki/Nominatim
5. Leaflet API Reference: Mobile-friendly interactive maps. URL: https://leafletjs.com/reference.html

Ключові слова: вебзастосунок, TravelTech, API-інтеграція, штучний інтелект, геокодування

Keywords: web application, TravelTech, API integration, artificial intelligence, geocoding

Науковий керівник: доцент Лобода Ю.Г.

СИНХРОННИЙ МІЖМОВНИЙ ПЕРЕКЛАД У СКЛАДІ ВЕБПЛАТФОРМИ ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ

Довгун Данило

студент 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Розвиток інформаційних технологій та глобалізація освітнього простору зумовили корінні зміни у підходах до організації навчального процесу. Дистанційне навчання перестало бути альтернативою традиційній освіті та перетворилося на її повноправну складову. За прогнозами аналітичної компанії Global Market Insights, обсяг світового ринку платформ для електронного навчання до 2028 року перевищить 1 трильйон доларів США [1].

Через пандемію COVID-19 освітні установи почали масовий перехід до онлайн-формату навчання: за даними ЮНЕСКО у 2022 році 79% освітніх закладів впровадили різні типи цифрових рішень для послуг з визнання та оцінювання [2]. Цей процес показав як потенціал цифрових освітніх платформ, так і їх обмеження – зокрема, проблему мовного бар'єра, яка є однією з найсерйозніших перешкод для розвитку міжнародної освіти. Українські студенти, які навчаються за кордоном, іноземні викладачі, які працюють з україномовною аудиторією, а також учасники міжнародних академічних програм стикаються з необхідністю подолання мовного бар'єра в режимі реального часу. Наявні рішення здебільшого або вимагають залучення живого перекладача, що підвищує вартість освітнього процесу, або пропонують переклад у вигляді субтитрів.

Стрімкий прогрес у галузі штучного інтелекту, зокрема поява нейромережевих моделей автоматичного розпізнавання мовлення, як-от Whisper від OpenAI, та вдосконалення систем синтезу природного мовлення, відкрив можливість інтегрувати автоматичний переклад безпосередньо в середовище онлайн-конференції. Організація відеозв'язку з нейромережевими засобами дозволяє створити освітню платформу, де переклад є частиною комунікації.

Сучасний ринок програмного забезпечення для дистанційної освіти представлений широким спектром рішень, які можна поділити на три категорії: універсальні платформи для відеоконференцій, спеціалізовані системи управління навчанням та гібридні освітні середовища [3].

До першої категорії належать Zoom, Google Meet та Microsoft Teams – інструменти, що набули масового поширення в освітній галузі під час пандемії COVID-19. Їх перевага полягає у простоті використання та доступності: усі три платформи пропонують безплатні тарифні плани для базового використання. Водночас з погляду підтримки багатомовності ці рішення мають обмеження. Zoom та Google Meet мають функцію автоматичних субтитрів лише в окремих мовах та виключно на платних тарифах; функція синхронного усного перекладу з аудіовідтворенням в зазначених платформах у безплатному доступі не представлена [4]. Microsoft Teams надає можливість синхронного перекладу субтитрів у рамках корпоративної підписки Microsoft 365, однак ця функція не передбачає озвучення перекладеного мовлення та є недоступною для освітніх установ без відповідного ліцензування [5].

До другої категорії програм для дистанційної освіти належать системи управління навчанням, зокрема Moodle, Coursera та Google Classroom. Ці платформи орієнтовані на організацію асинхронного навчального процесу: управління курсами, завданнями, тестуванням та успішністю. Функціональність відеоконференцзв'язку в них або відсутня, або реалізована через інтеграцію зовнішніх сервісів. Підтримка багатомовності обмежується локалізацією інтерфейсу та можливістю додавати субтитри до записаних відеоматеріалів – синхронний переклад живого мовлення ці системи не підтримують.

Третю категорію представляють гібридні рішення, що поєднують відеозв'язок з елементами управління навчанням, зокрема Whereby та BigBlueButton. BigBlueButton є платформою, орієнтованою безпосередньо на освітню галузь: вона підтримує спільну роботу з дошкою, опитування та розбивку учасників на групи. Однак функція міжмовного перекладу мовлення в реальному часі в зазначених платформах не реалізована.

Відповідно, аналіз підтверджує, що жодна з масових безплатних платформ не забезпечує єдиного рішення, яке поєднувало б відеоконференцзв'язок, управління навчальними матеріалами та синхронний усний переклад між учасниками в межах єдиного застосунку. Жодна з розглянутих масових платформ – Zoom, Google Meet, Microsoft Teams, Moodle та BigBlueButton – не забезпечує у безплатному режимі синхронного усного перекладу мовлення між учасниками з відтворенням перекладеного аудіо. Наявні рішення або обмежуються субтитрами із затримкою, або вимагають корпоративного ліцензування. Це визначило практичну доцільність розробки платформи Ew/oBs (Education without Borders) та допомогло сформувати вихідні функціональні вимоги до неї.

Створена програмна платформа функціонально охоплює чотири модулі: управління навчальними кімнатами з парольним захистом, відеоконференцзв'язок

у реальному часі, ШІ-переклад мовлення та управління навчальними матеріалами. Її нефункціональні вимоги – доступність через браузер, кросплатформність, використання захищеного протоколу HTTPS та незалежна масштабованість компонентів – забезпечили практичну придатність платформи до використання в умовах навчального процесу.

Автором реалізовано мікросервісну архітектуру системи, яка складається з двох незалежних серверних компонентів. Основний сервер на базі FastAPI відповідає за маршрутизацію, управління кімнатами та файлову систему. Окремий ШІ-мікросервіс виконує обчислювально-навантажені задачі обробки мовлення.

Розроблена серверна частина основного застосунку реалізує цикл управління навчальними кімнатами. Система парольного захисту забезпечує розмежування доступу до кімнат без реляційної бази даних. Алгоритм верифікації реалізує перевірку доступу, що обробляє всі можливі стани кімнати. Файлова система сервера організована за ієрархічним принципом, де кожна кімната має власну директорію для файлів та нотаток.

Реалізовано модуль відеоконференцзв'язку з використанням платформи Agora. Одноранговa передача медіаданих між браузерами учасників забезпечує низьку затримку відеопотоку, не залежну від пропускної здатності центрального сервера. Підтримка HTTPS-тунелювання дозволяє використовувати камеру та мікрофон на мобільних пристроях.

Розроблено ШІ-мікросервіс, який реалізує обробку голосового повідомлення з чотирьох послідовних етапів. На першому етапі нейромережева модель транскрибує аудіо у текст із зазначенням вхідної мови. На другому –мікросервіс передає розпізнаний текст до перекладача та отримує переклад у відповідному напрямку. На третьому етапі мікросервіс синтезує мовлення. На четвертому – синтезоване аудіо кодується та повертається клієнту разом з оригінальним та перекладеним текстом.

Реалізований клієнтський інтерфейс платформи складається з трьох функціональних сторінок. Головна сторінка реалізує навігацію до двох основних модулів. Сторінка відеоконференції містить два відеоконтейнери, панель керування з кнопкою запису голосу, перемикач напрямку перекладу та чат із відображенням оригінального тексту і перекладу одночасно. Сторінка навчальних матеріалів забезпечує завантаження файлів з вбудованим переглядачем та створення і перегляд текстових нотаток.

Розроблена платформа має потенціал практичного застосування у сфері міжнародної дистанційної освіти, зокрема для проведення занять між україномовними та англомовними учасниками навчального, організації міжнародних академічних обмінів та дистанційного навчання в умовах, що унеможливлюють фізичну присутність учасників в одному місці.

Список використаних джерел

1. Перспективи ринку онлайн-навчання. *Bestclevers*. URL: https://bestcleverslms.com/piznavalne/perspektyvy-rynku-onlayn-navchannia/
2. Higher education global trends report: towards inclusive, equitable and quality higher education in an internationally mobile landscape. *UNESCO*. 2026. URL: https://doi.org/10.54674/IN.2026.94
3. Ільчук Н.М. Найпопулярніші освітні платформи для організації дистанційного навчання. URL: https://vseosvita.ua/blogs/naipopuliarnishi-osvitni-platformy-dlia-orhanizatsii-dystantsiinoho-navchannia-102669.html
4. Google Meet тепер може перекладати в реальному часі з іноземних мов. URL: https://24tv.ua/tech/google-meet-teper-perekladaye-golos-realnomu-chasi_n2827238
5. Використання динамічних субтитрів в нарадах Microsoft Teams. URL: https://support.microsoft.com/en-us/office/use-live-captions-in-microsoft-teams-meetings-4be2d304-f675-4b57-8347-cbd000a21260

Ключові слова: истанційне навчання, освітня платформа, відеоконференцзв'язок, штучний інтелект, автоматичний переклад мовлення, розпізнавання мовлення, синтез мовлення, мікросервісна архітектура, FastAPI, Agora, вебзастосунок

Keywords: distance learning, educational platform, video conferencing, artificial intelligence, automatic speech translation, speech recognition, speech synthesis, microservice architecture, FastAPI, Agora, web application

Науковий керівник: доцент Трофименко О.Г.

ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ ВИБІРКОВИМИ ДИСЦИПЛІНАМИ

Падецький Деніс

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія

Відповідно до Закону України «Про вищу освіту» [1], здобувачі вищої освіти мають право формувати індивідуальну освітню траєкторію шляхом самостійного вибору навчальних дисциплін у межах освітньої програми. Вибіркові дисципліни є важливою складовою освітнього процесу, оскільки дозволяють здобувачам поглиблювати знання в обраних напрямах, розширювати професійні компетентності та враховувати власні академічні й кар'єрні інтереси.

Нормативно визначено, що обсяг вибіркових компонентів має становити не менше 25 % від загального обсягу освітньої програми. Організація процедури вибору дисциплін покладається на заклади вищої освіти, які повинні забезпечити відкритість інформації про курси, доступність процесу вибору та рівні можливості для всіх здобувачів освіти [2]. У зв'язку з цим зростає потреба у впровадженні сучасних інформаційних систем, що автоматизують формування

індивідуальних освітніх траєкторій, підвищують ефективність управління навчальним процесом і зменшують адміністративне навантаження.

У більшості закладів вищої освіти процес вибору дисциплін досі супроводжується значним обсягом ручної обробки даних, використанням електронних таблиць або розрізнених програмних засобів [3–7]. Це ускладнює контроль за кількістю місць у групах, підвищує ймовірність помилок і потребує додаткових адміністративних ресурсів. Тому актуальною є розробка спеціалізованої інформаційної системи, що забезпечує централізоване управління процесом вибору дисциплін та автоматизований розподіл студентів.

Метою роботи є розробка веборієнтованої інформаційної системи управління вибірковими дисциплінами, яка забезпечує реєстрацію студентів, перегляд каталогу дисциплін, формування вибору та автоматизований розподіл заявок.

Архітектура системи побудована за клієнт-серверним принципом. Серверна частина реалізована мовою Python із використанням фреймворку FastAPI, що забезпечує високу продуктивність, підтримку асинхронної обробки запитів та автоматичну генерацію документації API. Для зберігання даних використовується реляційна база даних MySQL.

Серверний застосунок містить набір REST API-ендпоінтів для взаємодії між клієнтською частиною та базою даних. Для валідації вхідних і вихідних даних застосовуються моделі Pydantic, які забезпечують контроль структури та типів даних.

Особливістю реалізації є використання механізму Dependency Injection у FastAPI для управління підключеннями до бази даних. Для кожного HTTP-запиту створюється окреме з’єднання, яке автоматично закривається після завершення обробки, що забезпечує ізоляцію транзакцій і підвищує надійність системи.

Клієнтська частина реалізована як SPA на нативному JavaScript. Основою архітектури є централізований state-об’єкт, що містить інформацію про користувача, список дисциплін, обрані курси та параметри інтерфейсу. Це дозволяє оновлювати інтерфейс без перезавантаження сторінки.

Інтерфейс користувача побудований за принципом покрокового майстра (wizard), що складається з трьох етапів: реєстрація студента, вибір дисциплін та підтвердження заявки. Такий підхід зменшує кількість помилок під час введення даних.

Центральним компонентом системи є модуль вибору дисциплін. Після реєстрації студент отримує доступ до каталогу дисциплін, де формує власний набір вибіркових курсів. (рис. 1).

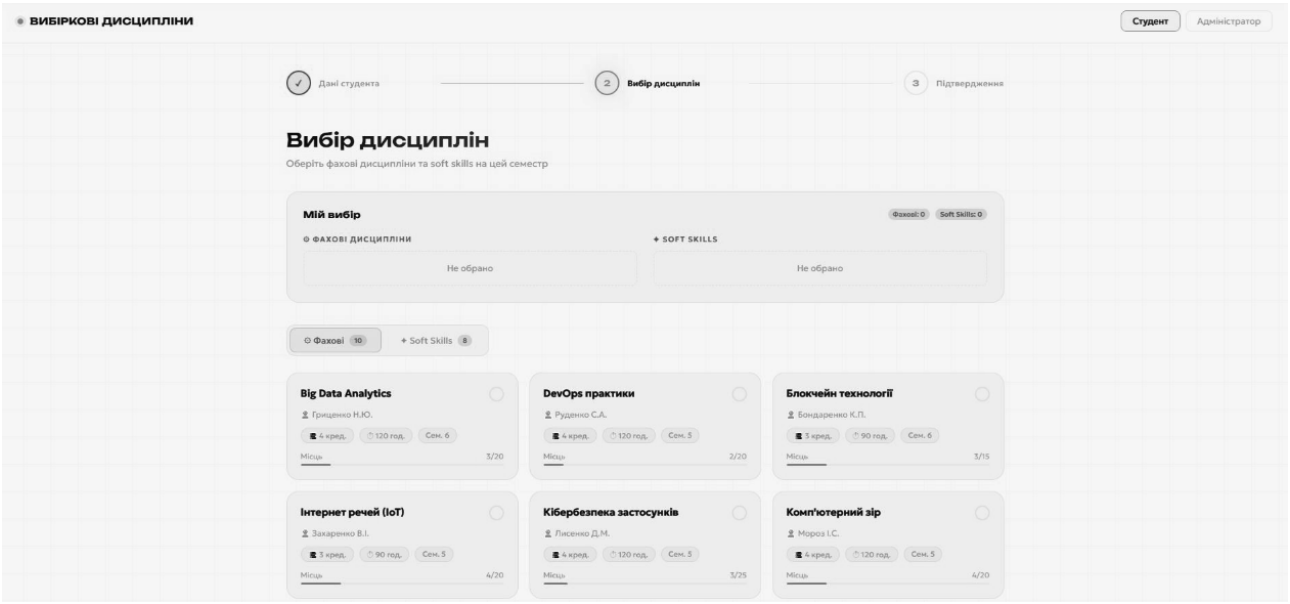


Рисунок 1 – Вибір дисциплін

На клієнтському рівні реалізовано інтерактивний вибір дисциплін із миттєвим оновленням інтерфейсу. На серверному рівні виконується повторна перевірка бізнес-правил, що гарантує цілісність даних навіть при прямому зверненні до API.

Для збереження результатів вибору використовується механізм UPSERT (INSERT ... ON DUPLICATE KEY UPDATE), що дозволяє створювати або оновлювати записи без додаткових перевірок.

Адміністративний модуль забезпечує керування дисциплінами, перегляд статистики та запуск автоматизованого розподілу студентів (рис. 2). Для аналітики використовуються SQL-агрегатні запити з GROUP BY та LEFT JOIN.

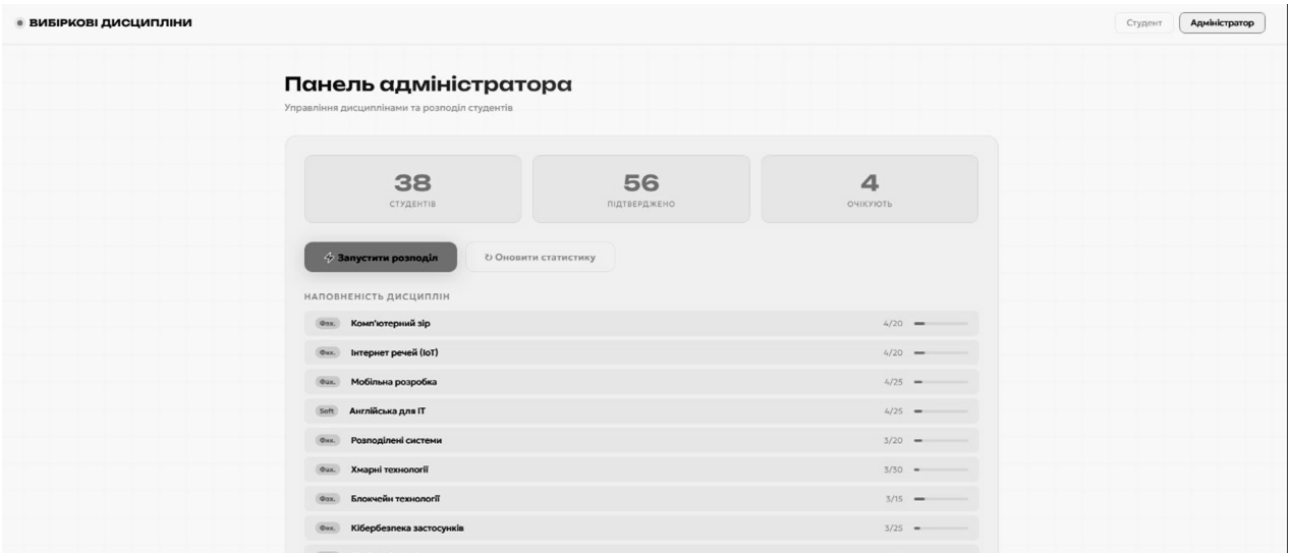


Рисунок 2 – Панель адміністратора

Алгоритм розподілу студентів належить до класу жадібних алгоритмів. Заявки сортуються за пріоритетом і часом подання. Для кожної заявки перевіряється наявність вільних місць у дисципліні. У разі наявності місця заявка

підтверджується, інакше відхиляється. Складність алгоритму визначається операцією сортування і становить O(N log N), де N – кількість заявок.

Такий підхід забезпечує достатню ефективність для умов функціонування закладу вищої освіти та дозволяє виконувати розподіл навіть при значній кількості студентів.

Для оцінки якості програмного продукту проведено комплексне тестування, яке включало функціональні, інтеграційні та навантажувальні перевірки.

Функціональне тестування здійснювалося методом «чорного ящика». Було сформовано десять тестових сценаріїв, що охоплювали всі основні функції системи: реєстрацію студентів, перегляд каталогу дисциплін, перевірку бізнес-правил, збереження вибору та запуск алгоритму розподілу. Усі тест-кейси були виконані успішно.

Навантажувальне тестування проводилося за допомогою інструменту Apache Benchmark. Під час експерименту генерувалося 1000 HTTP-запитів при 100 одночасних з'єднаннях. Отримані результати показали, що середній час відповіді основних ендпоінтів не перевищує 150 мс, а значення p95 для всіх CRUD-операцій залишаються нижчими за встановлену межу 500 мс.

Окремо було виконано кросбраузерне тестування у браузерах Google Chrome, Mozilla Firefox та Opera. Перевірка підтвердила коректну роботу інтерфейсу, адаптивність та відсутність критичних помилок відображення на різних типах пристроїв.

У результаті виконаної роботи розроблено веборієнтовану інформаційну систему управління вибірковими дисциплінами, що автоматизує процес формування індивідуальної освітньої траєкторії студентів.

Реалізована система поєднує сучасні технології веброзробки, REST-архітектуру та ефективний алгоритм автоматизованого розподілу студентів. Використання FastAPI забезпечило високу продуктивність серверної частини, а SPA-підхід дозволив створити зручний та адаптивний користувацький інтерфейс.

Список використаних джерел

1. Про вищу освіту: Закон України // Відомості Верховної Ради (ВВР), 2014, № 37-38, ст. 2004. https://zakon.rada.gov.ua/laws/show/1556-18#Text
2. Малецька І. Актуальні підходи формування каталогу вибіркових дисциплін освітньої програми у вищій школі. Acta Paedagogica Volynienses, 2025. 4, 172–178, doi: https://doi.org/10.32782/apv/2025.4.24
3. Moodle. URL: https://moodle.org
4. Семигіна, Т.В. & Новак, А.С. (2023). Інформаційні системи для вибіркових дисциплін у закладах вищої освіти.Vectors of the development of science and education in the modern world : collective monograph / V. Shpak, ed. (с.5-13). Sherman Oaks, California : GS Publishing Services. https://www.eo.kiev.ua/resources/zmist/mono_2023_14/mono_2023_14.pdf
5. Microsoft Teams для навчальних закладів. URL: https://www.microsoft.com/uk-ua/education/products/teams

6. Canvas by Instructure: World Leading LMS. URL: https://www.instructure.com/canvas
7. Open edX. URL: https://openedx.org/?nab=0

Ключові слова: інформаційна система, вибіркові дисципліни, FastAPI, SPA, REST API, автоматизований розподіл, вебзастосунок

Key words: information system, elective courses, FastAPI, single-page application (SPA), REST API, automated allocation, web application

Науковий керівник: доцент Логінова Н.І.

БАГАТОРІВНЕВА АРХІТЕКТУРА ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗОВАНОГО ПІДБОРУ ВАКАНСІЙ НА ПЛАТФОРМІ .NET 8

Сметана Владислав

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Автоматизація підбору вакансій і кандидатів є актуальною задачею для сучасних інформаційних систем, оскільки значна частина даних у цій предметній області подається у вигляді текстових резюме, описів вакансій і переліків професійних навичок. Ручне зіставлення таких даних потребує часу, залежить від повноти поданої інформації та може супроводжуватися суб'єктивністю первинного оцінювання [1]. Тому доцільним є створення програмних рішень, які поєднують обробку текстових даних, структуроване зберігання інформації та формування рекомендацій на основі визначених ознак відповідності.

Метою роботи є розгляд багаторівневої архітектури вебзастосунку JobMatch, призначеного для автоматизованого підбору вакансій на основі аналізу резюме. Система орієнтована на підтримку сценаріїв завантаження резюме, вилучення текстової інформації з документів, збереження профілю кандидата, порівняння його характеристик із вимогами вакансій та відображення пояснюваних результатів рекомендації.

Для реалізації серверної частини системи використано платформу .NET 8 та ASP.NET Core Minimal API. Такий підхід дає змогу компактно описувати HTTP-маршрути, зменшити обсяг шаблонного коду та зосередити логіку обробки запитів у межах чітко визначених endpoint-ів [2, 3]. Minimal API є доцільним для навчально-прикладного вебзастосунку, у якому основними сценаріями є робота з кандидатами, вакансіями, резюме, рекомендаціями та аналітичними даними.

Архітектура JobMatch побудована як багаторівнева. Клієнтський рівень забезпечує взаємодію користувача з вебінтерфейсом, API-рівень приймає HTTP-запити та повертає структуровані відповіді, прикладний рівень координує основні сценарії роботи системи, доменний рівень описує сутності предметної області, інфраструктурний рівень реалізує обробку файлів і рекомендаційний механізм, а рівень доступу до даних відповідає за взаємодію з базою даних.

Такий поділ спрощує супровід коду, ізолює відповідальність компонентів і дозволяє змінювати окремі частини системи без повного перепроєктування застосунку.

Для зберігання даних використано PostgreSQL у поєднанні з Entity Framework Core. Реляційна модель є обґрунтованою для задачі JobMatch, оскільки предметна область містить взаємопов'язані сутності: користувачів, резюме, вакансії, професійні навички, зв'язки між кандидатами й навичками, зв'язки між вакансіями й вимогами, результати рекомендацій та журнали прогнозування. Entity Framework Core забезпечує об'єктно-реляційне відображення, роботу з міграціями та зручну інтеграцію з .NET-застосунком. PostgreSQL використовується як надійна відкрита реляційна СКБД для зберігання структурованих даних [4].

Окремим компонентом системи є модуль обробки резюме. Він підтримує роботу з файлами форматів TXT, DOCX та PDF. Для DOCX-документів застосовується вилучення тексту зі структури Office Open XML, а для PDF-файлів – отримання текстового представлення сторінок документа. Після вилучення тексту виконується нормалізація, токенізація та підготовка даних до подальшого зіставлення з описами вакансій. Це дозволяє перетворити неструктурований документ резюме на формалізований профіль кандидата.

Рекомендаційний механізм системи поєднує правило-орієнтовану оцінку відповідності та модель машинного навчання, реалізовану засобами ML.NET. У системі враховуються збіги професійних навичок, текстова подібність між резюме та вакансією, відповідність досвіду, освіти та інші характеристики. ML.NET використовується як бібліотека, що інтегрується безпосередньо з .NET-екосистемою та дозволяє виконувати навчання і використання моделі без окремого ML-сервісу [5].

Для документування та ручної перевірки серверного API використано Swagger/OpenAPI. Наявність інтерактивної документації спрощує перевірку доступних endpoint-ів, структури запитів і відповідей, а також допомагає контролювати коректність взаємодії між клієнтською та серверною частинами системи [6]. Клієнтський інтерфейс реалізовано засобами HTML, CSS та JavaScript без використання окремого frontend-фреймворку, що є достатнім для реалізації основних сценаріїв дипломного вебзастосунку.

Запропонована архітектура забезпечує логічне розділення компонентів JobMatch і відповідає задачі автоматизованого підбору вакансій на основі аналізу резюме. Практична цінність такого підходу полягає в тому, що система може підтримувати повний цикл роботи з даними: від завантаження резюме й збереження профілю кандидата до формування рекомендацій і пояснення результатів. Отже, багаторівнева архітектура на платформі .NET 8 є доцільною основою для реалізації вебзастосунку, який поєднує API, базу даних, обробку документів і рекомендаційний модуль.

Список використаних джерел

1. Редакція DOU. Підсумки року на IT-ринку праці: +31% вакансій, продуктове IT зростає, але не без скорочень. *DOU*. 2024. URL: https://dou.ua/lenta/articles/jobs-and-trends-2024/.
2. Microsoft. What's new in .NET 8. Microsoft Learn. URL: https://surl.li/wuaqyy.
3. Microsoft. Minimal APIs overview. Microsoft Learn. URL: https://surl.li/gxaicn
4. The PostgreSQL Global Development Group. PostgreSQL Documentation. URL: https://www.postgresql.org/docs/.
5. Microsoft. Overview of ML.NET. Microsoft Learn. URL: https://surl.li/lkeawp.
6. OpenAPI Initiative. OpenAPI Specification v3.1.0. URL: https://spec.openapis.org/oas/v3.1.0.html.

Ключові слова: вебзастосунок, багаторівнева архітектура, ASP.NET Core, PostgreSQL, ML.NET, підбір вакансій

Keywords: web application, multilayer architecture, ASP.NET Core, PostgreSQL, ML.NET, vacancy matching

Науковий керівник: доцент Лобода Ю. Г.

ВЕБСЕРВІС ДЛЯ АВТОМАТИЗАЦІЇ ОБЛІКУ ТА ПРОГНОЗУВАННЯ ТОВАРНИХ ЗАПАСІВ

Флоря Діана

студентка 5-го курсу заочної форми навчання
факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія

Сучасні системи управління товарними запасами є важливою складовою ефективного функціонування підприємств торгівлі та логістики, оскільки безпосередньо впливають на швидкість обігу капіталу, рівень витрат на зберігання та здатність своєчасно задовольняти попит [1-3]. Традиційні підходи до обліку запасів часто ґрунтуються на ручному введенні даних або використанні застарілих інформаційних систем, що не дозволяє оперативно аналізувати великі обсяги даних та враховувати динаміку споживання. У результаті виникає потреба у створенні сучасних веборієнтованих рішень, які поєднують автоматизований облік, аналітику та прогнозування попиту на основі математичних моделей і методів обробки даних.

Розроблений вебсервіс побудований за багаторівневою архітектурою з використанням фреймворку Django та архітектурного патерну MTV. Серверна частина містить чітко структуровані модулі, кожен з яких відповідає за окрему предметну область, зокрема управління товарами, аналітику та прогнозування.

Обмін даними між клієнтом і сервером реалізовано через REST API у форматі JSON, що забезпечує незалежність клієнтської частини та можливість масштабування системи.

Основою складського обліку є поділ даних на поточні залишки та журнал транзакцій, що дозволяє поєднати високу швидкодію доступу до актуальних даних із повною історією змін. Кожна операція над товарами фіксується як окрема транзакція, що забезпечує прозорість обліку та можливість аудиту. Такий підхід дозволяє уникнути надмірних обчислень при отриманні залишків і підвищує продуктивність системи при роботі з великими обсягами даних.

REST API реалізує стандартні операції створення, читання, оновлення та видалення даних, а також містить розширені бізнес-ендпоінти для виконання складських операцій. Серіалізація даних побудована таким чином, щоб забезпечити різні рівні деталізації інформації залежно від сценарію використання, що підвищує гнучкість взаємодії між компонентами системи. Окрему увагу приділено атомарності операцій, що гарантує цілісність даних при паралельному виконанні запитів.

Аналітичний модуль системи реалізує обробку даних із використанням бібліотеки Pandas і включає алгоритми ABC-аналізу, оцінки оборотності запасів, розрахунку точки замовлення та виявлення надлишкових або «мертвих» запасів. Ці алгоритми дозволяють класифікувати товари за рівнем їхньої значущості, оцінювати ефективність використання запасів та визначати оптимальні моменти поповнення складу. Особливістю реалізації є використання агрегованих даних та статистичних показників, що забезпечує швидке отримання результатів навіть при великій кількості транзакцій.

Модуль прогнозування є одним із ключових компонентів системи і базується на аналізі часових рядів. У межах реалізації використано кілька підходів до прогнозування, включаючи методи ковзного середнього, експоненційного згладжування, лінійної регресії та сезонного аналізу [4-8]. Для кожного товару система автоматично обирає найточнішу модель на основі показника середньої абсолютної відсоткової похибки. Це дозволяє адаптувати прогнозування до різних типів попиту, від стабільного до сезонного або трендового.

Додатково у модулі прогнозування реалізовано механізми побудови довірчих інтервалів та моделювання вичерпання запасів, що дозволяє оцінювати ризики дефіциту товарів у майбутньому. Автоматичне оновлення прогнозів здійснюється за допомогою фонових задач, що забезпечує актуальність даних без втручання користувача.

Клієнтська частина системи реалізована як односторінковий вебдодаток, що забезпечує швидку взаємодію користувача з системою без перезавантаження сторінок. Інтерфейс використовує асинхронне завантаження даних через REST API, що дозволяє динамічно оновлювати аналітичні показники, графіки та списки товарів. Візуалізація даних реалізована за допомогою інтерактивних графіків, що підвищує зручність аналізу інформації (рис. 1).

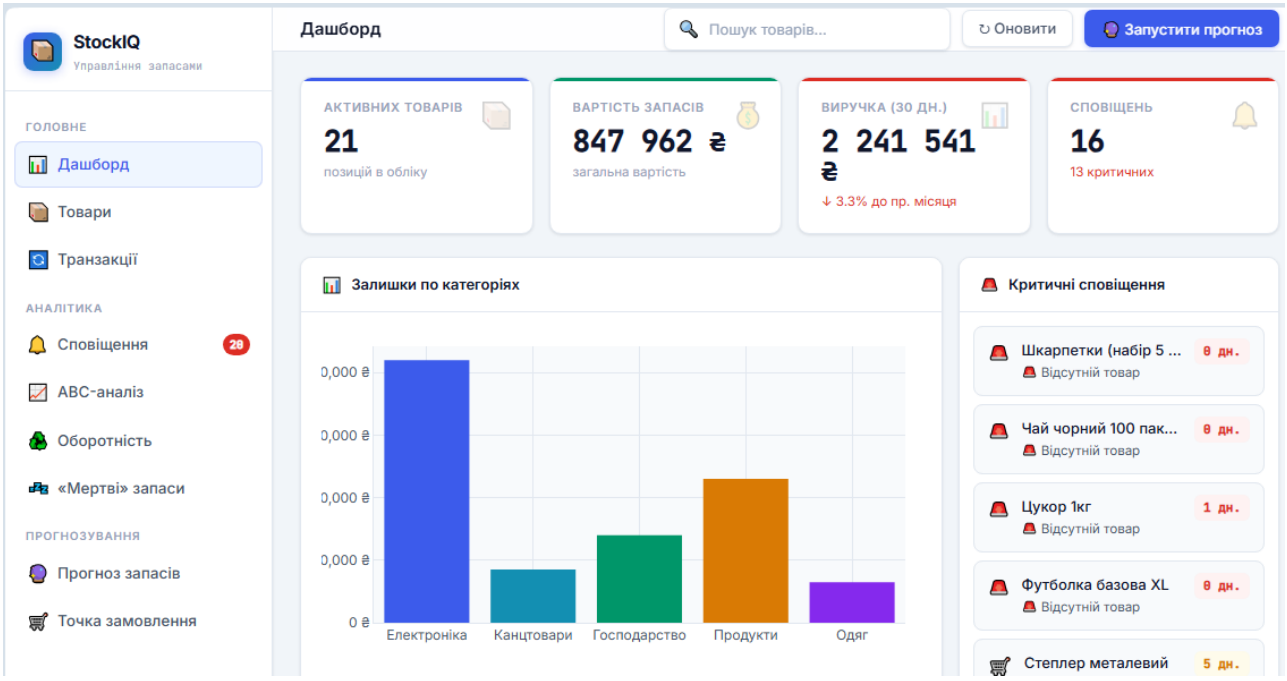


Рисунок 1 – Головна сторінка вебсервісу

Проведене тестування системи підтвердило коректність реалізації основних функціональних вимог та відповідність нефункціональним критеріям, зокрема продуктивності та стабільності. CRUD-операції виконуються у межах прийнятного часу відповіді, аналітичні запити обробляються ефективно навіть при значному обсязі даних, а прогнозні обчислення оптимізовані за рахунок використання фонових процесів. Окремо було підтверджено ефективність використання індексування бази даних, що суттєво прискорює виконання складних запитів.

Отримані результати свідчать про те, що розроблений вебсервіс є зручним інструментом для автоматизації обліку та прогнозування товарних запасів. Система забезпечує комплексну обробку даних, підтримує аналітичні функції та дозволяє підвищити якість управлінських рішень за рахунок використання сучасних методів аналізу та прогнозування.

Список використаних джерел

1. Програмне забезпечення для планування ресурсів підприємства. URL: https://www.sap.com/ukraine/products/erp.html
2. Introducing NetSuite Next–the future of NetSuite. URL: https://www.netsuite.com/portal/products/netsuite-next.shtml?inid=hp-netsuite-next
3. Автоматизація бізнес-процесів в єдиній системі Odoo. URL: https://surl.li/hbunmp
4. Moving Average (MA): Purpose, Uses, Formula, and Examples. URL: https://surl.lu/ulcpzc
5. Smoothing Techniques. URL: https://www.sciencedirect.com/topics/social-sciences/exponential-smoothing

6. Jonath Jose. Introduction to time series analysis and its applications. 2022. URL: https://www.researchgate.net/publication/362389180_INTRODUCTION_TO_TIME_SERIES_ANALYSIS_AND_ITS_APPLICATIONS
7. Causal Forecasting. URL: https://www.larksuite.com/en_us/topics/retail-glossary/causal-forecasting
8. Метод експертних оцінок «Дельфі». URL: https://buklib.net/books/32479/

Ключові слова: вебсервіс, управління товарними запасами, прогнозування попиту, Django, PostgreSQL, REST API, часові ряди, складський облік

Key words: inventory management, demand forecasting, web service, Django, PostgreSQL, REST API, time series analysis, warehouse automation

Науковий керівник: доцент Логінова Н.І.

РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ-ГЕНЕРАТОРА ПРИВІТАЛЬНИХ ЛИСТІВОК «GREETIO»

Чикунов Павло

кандидат технічних наук, доцент, доцент кафедри програмної інженерії Національного університету «Одеська юридична академія»

Лук'янов Артем

студент 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Актуальність теми зумовлена цифровізацією соціальних взаємодій та зростаючим попитом на зручні інструменти для персоналізованої комунікації. Традиційні паперові листівки поступово втрачають свою популярність, поступаючись місцем цифровим аналогам. Наявні мобільні рішення, прикладом яких є платформа «Canva», часто пропонують перевантажений деталями дизайн та велику кількість панелей інструментів, що може ускладнювати роботу непідготовленого користувача [1].

Метою виконання дослідження є розробка, проєктування та тестування нативного мобільного застосунку «Greetio» для швидкого та зручного створення привітальних листівок на основі готових шаблонів.

Для досягнення мети дослідження необхідно вирішити такі завдання:

- провести аналіз предметної області та навяних аналогів;
- спроєктувати архітектуру застосунку та вибрати концепцію мінімалістичного інтерфейсу користувача;
- реалізувати функціональну логіку: покроковий редактор, інструмент інтерактивної обрізки зображень та механізм рендерингу фінальної листівки;
- протестувати застосунок з метою виявлення помилок та оптимізації продуктивності при обробці графіки.

Технологічною основою проєкту обрано мову програмування Kotlin [2] та середовище розробки Android Studio [3]. Візуальну частину застосунку побудовано на базі сучасного декларативного UI-фреймворку Jetpack Compose [4]. Архітектура програмної системи реалізована за допомогою патерну MVVM (Model-View-ViewModel), де управління станом екранів здійснюється через реактивні потоки даних. Це забезпечує швидке оновлення інтерфейсу, низьку зв'язність компонентів та масштабування системи (рис. 1).

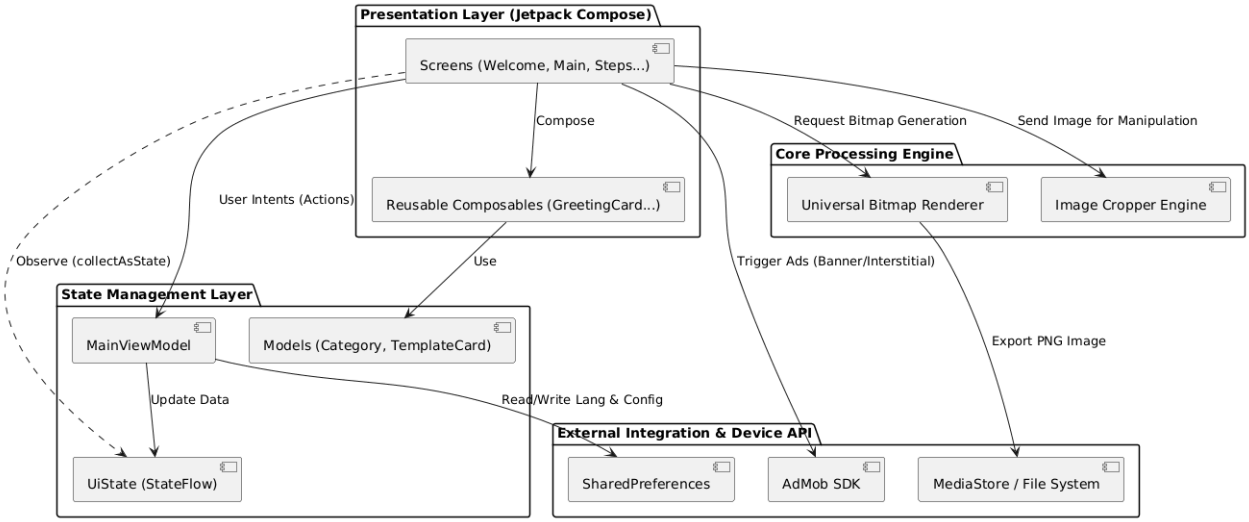


Рисунок 1 – UML-діаграма компонентів застосунку «Greetio»

Особливу увагу приділено розробці механізму рендерингу фінальної листівки. Оскільки графічний інтерфейс побудовано на Jetpack Compose, було розроблено алгоритм асинхронного перетворення декларативного UI у растрове зображення [5] за допомогою віртуального контейнера Headless Compose.

Процес створення листівки розбитий на кроки(рис. 2): вибір шаблону, редагування текстового наповнення та інтеграція фото користувача за допомогою модуля кадрування на основі математичних трансформацій [6] (рис. 2). Збереження згенерованого файлу у високій роздільній здатності (1200px) відбувається з використанням API MediaStore [7].

Комплексне тестування застосунку на емуляторах та фізичних Android-пристроях підтвердило стабільність алгоритмів обробки зображень та відсутність витоків оперативної пам'яті. У подальшому заплановано інтеграцію хмарної бази даних для завантаження нових шаблонів, а також впровадження штучного інтелекту для автоматичної генерації оригінальних текстів привітань.

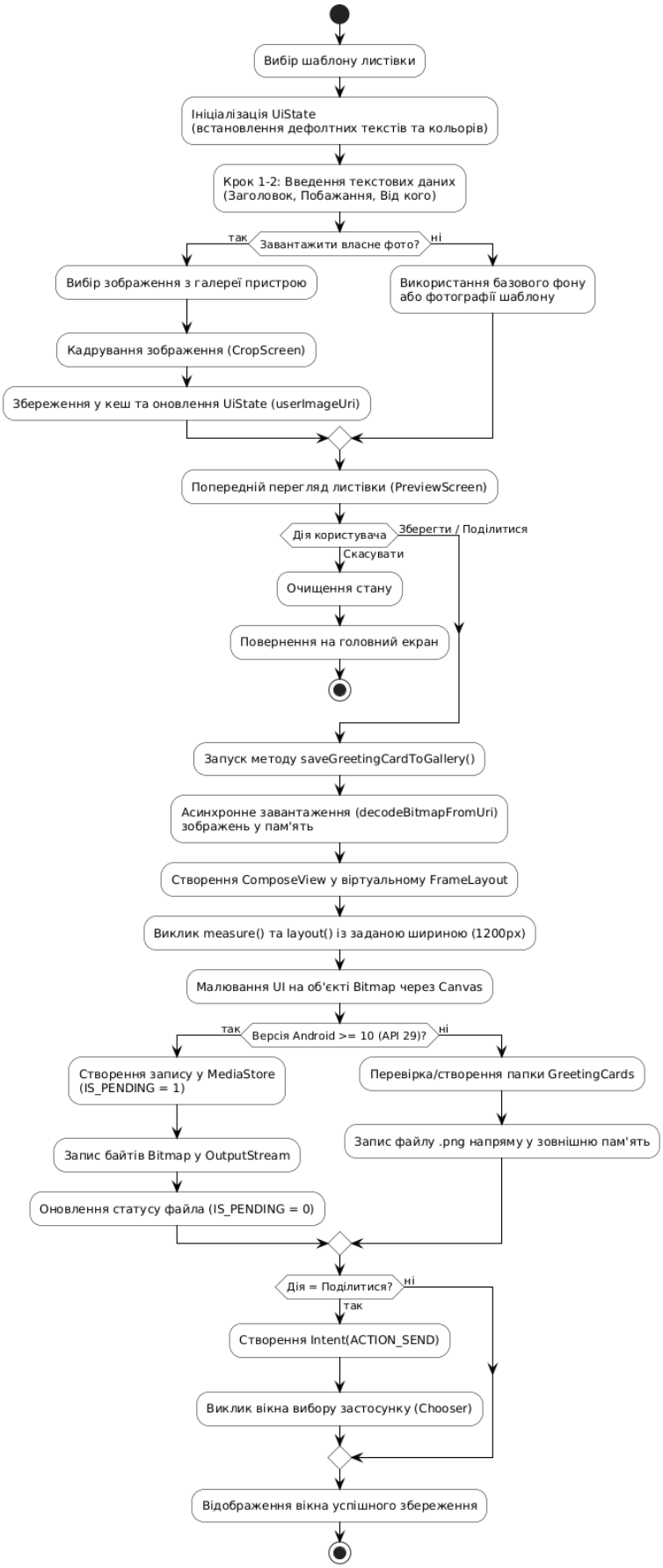


Рисунок 2 – Візуалізація алгоритму створення, рендерингу та збереження привітальної листівки

Список використаних джерел

1. Сервіс для графічного дизайну Canva. Режим доступу: https://www.canva.com.
2. Офіційна документація мови програмування Kotlin. Kotlin Programming Language. Режим доступу: https://kotlinlang.org/docs/home.html.
3. Офіційний посібник із середовища розробки Android Studio. Android Developers. Режим доступу: https://developer.android.com/studio.
4. Сучасний інструментарій для створення нативного UI – Jetpack Compose. Android Developers. Режим доступу: https://developer.android.com/ jetpack/compose.
5. Робота з графікою, Bitmap та Canvas в Android. Android Developers. Режим доступу: https://developer.android.com/topic/performance/graphics.
6. Клас Matrix. Робота з математичними трансформаціями графіки. Android Developers. Режим доступу: https://developer.android.com/reference/kotlin/android/graphics/Matrix.
7. Збереження медіафайлів та робота з API MediaStore. Android Developers. Режим доступу: https://developer.android.com/reference/kotlin/android/provider/MediaStore.

Ключові слова: мобільний застосунок, генератор листівок, Android, Kotlin, Jetpack Compose, MVVM

Keywords: mobile application, greeting card generator, Android, Kotlin, Jetpack Compose, MVVM

ЧАТ-БОТИ ЯК ІНСТРУМЕНТ ФОРМУВАННЯ ФІНАНСОВОЇ ДИСЦИПЛІНИ СЕРЕД СТУДЕНТІВ

Тищенко Леся

студентка 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Проблема низького рівня особистого фінансового планування серед студентської молоді залишається актуальною попри широку доступність спеціалізованих застосунків. Дослідження поведінкової економіки свідчать, що головним бар'єром є не брак інструментів, а надмірна складність їхнього інтерфейсу та потреба у зміні звичних поведінкових патернів. Висока вартість фінансових консультацій, нестабільність ринків і проблеми з пошуком професійних фінансових консультантів створили сприятливі умови для використання автоматизованих фінансових консультантів [1]. Студенти, які не мають сталих навичок фінансового обліку, з більшою ймовірністю відмовляться від нового інструменту, якщо він потребує окремого встановлення, реєстрації або тривалого освоєння.

Більшість наявних чат-ботів для фінансового консультування є комплексними рішеннями, які потребують значного часу на освоєння, що робить їх непридатними для користувачів, чия мета полягає у швидкій та конкретній відповіді. Це безпосередньо підтверджує тезу про інтерфейсний бар'єр: складність – не перевага, а фактор відмови. Водночас антропоморфний дизайн чат-бота – діалог у форматі «питання бота – відповідь користувача» – позитивно впливає на

відчуття соціальної присутності й рівень довіри, що підвищує ймовірність дотримання рекомендацій системи [1]. Цей висновок є принциповим: простий діалоговий інтерфейс – не технічне спрощення, а осмислене проєктне рішення, яке збільшує залученість.

Студенти стикаються з конкретними труднощами у веденні особистого бюджету, а чат-бот, розроблений з урахуванням типових фінансових сценаріїв цільової аудиторії, здатен не лише надавати фінансову інформацію, а й формувати навички управління особистими витратами. Фінансовий стрес має виражений емоційний вимір, і інструменти, які враховують цей аспект через зрозумілий та підтримуючий інтерфейс, демонструють вищий рівень прийняття з боку користувачів [2]. Тому ефективність інструмента фінансового обліку визначається не лише функціональністю, а насамперед відповідністю його інтерфейсу поведінковим і когнітивним характеристикам цільової аудиторії.

Зниження «порогу входження» до системи фінансового обліку шляхом інтеграції її у вже звичне середовище щоденної комунікації – месенджер – здатне суттєво підвищити регулярність ведення особистого бюджету. Telegram, що станом на 2025 рік налічує понад 1 мільярд активних користувачів [3], є природним середовищем для студентської аудиторії: платформа використовується щодня, не потребує додаткового встановлення і знайома на рівні рефлексу.

Аналіз причин, через які студенти не охоче ведуть облік фінансів, дозволяє виокремити три основні бар'єри. По-перше, когнітивний бар'єр: потреба перемикатися між середовищами (месенджер → окремий застосунок) створює додатковий поведінковий опір. По-друге, інтерфейсний бар'єр: більшість фінансових застосунків орієнтована на досвідчених користувачів і містить зайві для початківця функції. По-третє, бар'єр звички: формування регулярної практики обліку потребує мінімальних перепон між наміром і дією. Чат-бот у месенджері усуває всі три бар'єри одночасно: він вже є там, де користувач перебуває, має спрощений діалоговий інтерфейс і не потребує переходу в окремий застосунок.

Для перевірки висунутої гіпотези авторкою розроблено Telegram чат-бот для обліку особистих доходів і витрат, орієнтований саме на студентську аудиторію. Вибір Python як мови реалізації зумовлений потребою у швидкому прототипуванні та наявністю зрілих бібліотек для роботи з Telegram Bot API. Збереження даних реалізовано у форматі JSON як достатньому для задач особистого обліку рішенні, що не ускладнює розгортання. Для забезпечення аналітичної цінності застосовано бібліотеку matplotlib, яка дає змогу будувати графіки співвідношення доходів і витрат та діаграми структури видатків за категоріями.

Ключові проєктні рішення бота підпорядковані меті зниження поведінкових бар'єрів. Кнопкове меню замість вільного введення тексту мінімізує кількість помилок і зменшує когнітивне навантаження. Категорії витрат (їжа, транспорт, покупки, розваги, здоров'я) та доходів (заробітна плата, подарунки, інше) визначено відповідно до типової структури можливого студентського бюджету. Можливість формувати звіти за день, тиждень, місяць або конкретну дату забезпечує

зворотний зв'язок, необхідний для формування рефлексивної фінансової поведінки.

Отже, чат-бот у месенджері є не просто зручною альтернативою фінансовим застосункам, а інструментом, конструктивно спрямованим на подолання поведінкових бар'єрів. Розміщення фінансового обліку у звичному комунікаційному середовищі знижує «ціну» регулярної звички, що є необхідною умовою формування фінансової дисципліни – особливо серед молодої аудиторії з відсутнім досвідом бюджетування.

Список використаних джерел

1. Kobets V., Kozlovskyi Ky. Application of chat bots for personalized financial advice. *Herald of Advanced Information Technology*. 2022. Vol. 5 No. 3. P. 229–242. DOI: https://doi.org/10.15276/hait.05.2022.18
2. Vijaykumar S, Shriya R, Vibha M. (2024). Empowering Financial Peace through Chatbot Guidance. *International Journal of Advanced Research in Science, Communication and Technology*. Vol. 4, No 1. P. 509-516. DOI: https://doi.org/10.48175/IJARSCT-15368.
3. Джугалик Д. Telegram має понад 1 млрд активних користувачів. *Межа*. 19.05.2025. URL: https://mezha.ua/news/telegram-1-milyard-koristuvachiv-300576 (дата звернення: 21.05.2026).

Ключові слова: фінансова дисципліна, поведінкові бар'єри, чат-бот, Telegram, особисте бюджетування, студентська аудиторія

Keywords: financial discipline, behavioral barriers, chatbot, Telegram, personal budgeting, student audience

Науковий керівник: доцент Трофименко О.Г.

ПРОБЛЕМИ ФУНКЦІОНУВАННЯ TELEGRAM-БОТІВ У СИСТЕМАХ ЦИФРОВОЇ КОМУНІКАЦІЇ

Клочкова Крістіна

студентка 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У сучасному інформаційному суспільстві месенджери перетворилися на ключові платформи цифрової комунікації. Особливе місце серед них посідає Telegram – третій за кількістю завантажень месенджер у світі [1], який об'єднує функції захищеного обміну повідомленнями, новинного агрегатора та платформи для розроблення автоматизованих сервісів. Однією з найбільш затребуваних функціональних можливостей платформи є підтримка чат-ботів – програмних агентів, призначених для автоматизованої взаємодії з користувачами за допомогою текстових або голосових повідомлень відповідно до закладених алгоритмів чи моделей обробки природної мови (NLP). Водночас масове

впровадження таких інструментів породжує низку технічних, комунікативних і безпекових проблем, які потребують систематичного наукового аналізу.

Telegram позиціонує себе як платформу з наскрізним шифруванням (end-to-end encryption), що стало однією з вагомих причин масового переходу користувачів із WhatsApp у січні 2021 року після зміни політики конфіденційності останнього [1]. Платформа надає можливість створювати секретні чати з функцією самознищення повідомлень, підписуватися на тематичні канали та брати участь у групових дискусіях. Окрім цього, вбудований платіжний API дозволяє розробникам інтегрувати в боти функцію збору платежів безпосередньо в інтерфейсі месенджера.

За функціональним призначенням Telegram-боти поділяються на інформаційні, сервісні, комерційні, навчальні та платіжні. Сучасні боти дедалі частіше інтегруються з технологіями штучного інтелекту та обробки природної мови, що суттєво підвищує якість взаємодії з користувачем і забезпечує персоналізацію відповідей. Використання таких інструментів забезпечує цілодобовий зв'язок із клієнтами [2], що є особливо актуальним для сфер електронної комерції, освіти, банківських послуг, технічної підтримки та державних цифрових сервісів.

Попри очевидні переваги, функціонування Telegram-ботів пов'язане з низкою системних обмежень. По-перше, діяльність ботів безпосередньо залежить від стабільності роботи платформи Telegram та якості інтернет-з'єднання кінцевого користувача: технічні збої або перевантаження серверів можуть тимчасово унеможливлювати доступ до сервісів, що порушує безперервність цифрової взаємодії. По-друге, більшість чат-ботів функціонує на основі заздалегідь визначених сценаріїв (скриптів), унаслідок чого вони не здатні адекватно опрацьовувати нестандартні або складні запити. Значний обсяг вхідних повідомлень може призводити до перевантаження системи та тимчасового припинення роботи бота. По-третє, суттєвою проблемою є юзабіліті інтерфейсів: значна частина ботів реалізована виключно через систему кнопок і шаблонних відповідей без підтримки вільного текстового введення, що ускладнює взаємодію для ряду категорій користувачів [3]. Особливі труднощі виникають у осіб з низьким рівнем цифрової грамотності, які не мають базового розуміння принципів роботи месенджера. З боку розробників нагальною проблемою залишається необхідність регулярного оновлення та технічної підтримки ботів з урахуванням змін API платформи та еволюції потреб аудиторії.

Однією з найгостріших проблем у сфері використання Telegram-ботів є стрімке зростання активності шкідливих ботів і відсутність достатніх механізмів верифікації їхньої достовірності з боку пересічних користувачів. Зловмисники створюють боти, що імітують офіційні сервіси банків, інтернет-магазинів або державних установ з метою отримання конфіденційних даних (фішинг). Додаткові ризики становлять автоматизований збір персональних даних, розповсюдження шкідливих посилань, застосування методів соціальної інженерії та формування бот-мереж.

Динаміка поширення шкідливих ботів підтверджується статистичними даними: якщо у 2018 році частка людського трафіку в мережі становила близько 62%, шкідливих ботів – 20%, а корисних – 18%, то вже у 2024 році ситуація кардинально змінилася [4]. Частка людського трафіку скоротилася до 49%, тоді як кількість шкідливих ботів зросла до 37%, що більш ніж удвічі перевищує частку корисних ботів (14%). Ця тенденція свідчить про загострення проблем у сфері кібербезпеки та вимагає розроблення ефективних захисних механізмів. Не менш важливою залишається проблема зберігання та обробки персональних даних користувачів. Неналежний рівень захисту API або серверної інфраструктури може призвести до витоку чутливої інформації. Практика показує, що розробники не завжди дотримуються вимог Загального регламенту захисту даних (GDPR) та вітчизняного законодавства у сфері персональних даних, що несе правові ризики для операторів та загрозу для кінцевих користувачів.

Проведений аналіз засвідчує, що Telegram-боти є потужним інструментом автоматизації цифрових комунікацій, проте їхнє повноцінне функціонування пов’язане з низкою технічних, юзабіліті- та безпекових викликів. Подальший розвиток цієї технології має бути спрямований на: вдосконалення механізмів захисту даних і верифікації ботів; підвищення функціональної надійності та стійкості до перевантажень; розроблення інтуїтивно зрозумілих інтерфейсів з підтримкою вільного діалогу; забезпечення відповідності вимогам чинного законодавства щодо обробки персональних даних. Лише комплексний підхід до вирішення означених проблем дозволить реалізувати потенціал Telegram-ботів як безпечного та ефективного інструменту цифрового сервісу.

Список використаних джерел

1. Telegram – statistics & facts. URL: https://www.statista.com/topics/9640/telegram/ (Дата звернення: 20.05.2026)
2. Трофименко О.Г., Прокоп Ю.В., Задерейко О.В., Логінова Н.І. Класифікація чат-ботів. Системні технології. 2022. № 2 (139). С. 147–159. DOI: https://doi.org/10.34185/1562-9945-2-139-2022-14
3. Чат-боти: плюси і мінуси. URL: https://ukr-bot.com/plyusi-ta-minusi-chatbotiv/ (Дата звернення: 20.05.2026)
4. Чи захоплюють інтернет шкідливі боти? URL: https://www.statista.com/chart/35950/share-of-global-web-traffic-generated-by-humans-and-bots (Дата звернення: 20.05.2026)

Ключові слова: Telegram, Telegram-бот, автоматизація, месенджер, шкідливі боти, кібербезпека, захист персональних даних

Keywords: Telegram, Telegram bot, automation, messenger, malicious bots, cybersecurity, personal data protection

Науковий керівник: доцент Трофименко О.Г.

Підсекція 6. Правові, етичні та суспільні аспекти цифрових технологій

ЗАКОНОДАВЧІ ІНІЦІАТИВИ ЩОДО РЕГУЛЮВАННЯ ЦИФРОВИХ ПЛАТФОРМ У США ТА ЄС

Альмужна Анастасія

студентка 1-го курсу факультету прокуратури та слідства (кримінальна юстиція) Національного університету «Одеська юридична академія»

«Акти про підзвітність App Store та Закон про свободу App Store (2025)» – це умовна назва сукупності законодавчих ініціатив, спрямованих на регулювання діяльності цифрових платформ, які забезпечують розповсюдження мобільних застосунків. Їх поява пов'язана з необхідністю вдосконалення антимонопольного регулювання цифрових ринків та створення більш рівних умов для всіх учасників цифрової економіки.

У наукових підходах наголошується, що великі цифрові платформи здатні формувати значний рівень ринкової влади, фактично контролюючи доступ розробників до користувачів і визначаючи правила функціонування екосистеми застосунків [1].

Це створює ризики обмеження конкуренції та посилення залежності розробників від умов, встановлених операторами платформ.

Зміст зазначених ініціатив охоплює врегулювання ключових аспектів діяльності магазинів застосунків, зокрема: прозорість правил розміщення програмного забезпечення, регулювання розміру комісійних платежів, можливість використання альтернативних платіжних систем, а також недопущення дискримінаційних умов щодо окремих розробників [2].

Такі положення спрямовані на формування більш відкритого та конкурентного цифрового ринку.

Важливою передумовою прийняття зазначених ініціатив є концентрація ринкової влади у великих технологічних компаній, які керують магазинами застосунків і можуть встановлювати обов'язкові комісійні збори, регулювати доступ до платформи та обмежувати використання альтернативних способів оплати, що потенційно створює ризики антиконкурентної поведінки та обмеження свободи діяльності розробників. У результаті цього формується нерівність умов між домінуючими платформами та незалежними учасниками ринку, що негативно впливає на рівень конкуренції та інноваційність цифрового середовища.

Додатковою передумовою виступає зростання уваги держав і міжнародних організацій до проблеми регулювання цифрових ринків, оскільки спостерігається тенденція до їх природної монополізації, що потребує спеціальних правових механізмів для забезпечення справедливої конкуренції та захисту прав усіх учасників ринку [3].

У цьому контексті подібні законодавчі ініціативи розглядаються як спроба адаптації антимонопольного регулювання до умов цифрової економіки.

Акти про підзвітність App Store та Закон про свободу App Store (2025) спрямовані на комплексне врегулювання діяльності цифрових платформ, які виконують функцію магазинів мобільних застосунків, та встановлення більш прозорих і справедливих правил їх функціонування. Їх ключовою метою є обмеження зловживання ринковою владою з боку великих технологічних компаній і створення рівних умов доступу для всіх розробників програмного забезпечення.

Основні положення цих ініціатив охоплюють запровадження вимог щодо прозорості правил розміщення застосунків, обов'язкове обґрунтування рішень щодо видалення або обмеження доступу до програмного забезпечення, а також недопущення дискримінаційного підходу до різних категорій розробників.

Окрему увагу приділено регулюванню комісійних зборів, які стягуються платформами, оскільки їх непрозорий або надмірний характер може впливати на економічну доступність цифрового ринку для малих і середніх розробників.

Важливим елементом правового регулювання є також забезпечення можливості використання альтернативних платіжних систем і каналів розповсюдження застосунків, що спрямовано на зменшення залежності розробників від внутрішніх механізмів платформ і підвищення рівня конкуренції.

Крім того, передбачається посилення наглядових і контрольних механізмів з боку державних регуляторів щодо діяльності операторів магазинів застосунків, зокрема у частині дотримання антимонопольного законодавства та недопущення зловживання домінуючим становищем. Це передбачає можливість застосування санкцій за порушення встановлених правил та обов'язок платформ звітувати про свою політику щодо доступу до екосистеми.

Акти про підзвітність App Store та Закон про свободу App Store (2025) мають суттєве значення для трансформації цифрових ринків, оскільки вони спрямовані на зміну правил функціонування платформ, що контролюють доступ до мобільних застосунків.

Одним із ключових наслідків є посилення конкурентного середовища, оскільки обмеження монопольних практик платформ і запровадження альтернативних механізмів розповсюдження застосунків створює більш рівні умови для розробників. У дослідженнях антимонопольних органів зазначається, що зниження бар'єрів входу на цифрові ринки сприяє розвитку малих і середніх технологічних компаній та підвищує рівень інноваційності.

Важливим аспектом впливу є також зміна бізнес-моделей великих технологічних компаній, оскільки регулювання комісійних зборів та вимоги до прозорості змушують платформи переглядати підходи до монетизації своїх екосистем.

Крім того, посилюється захист прав споживачів, оскільки більша конкуренція між платформами призводить до розширення вибору, покращення якості цифрових продуктів та зниження їх вартості [4].

Окремо слід відзначити вплив на антимонопольну політику держав, оскільки подібні акти відображають глобальну тенденцію до посилення контролю за великими цифровими платформами та адаптації правового регулювання до умов цифрової економіки [5].

У цьому контексті важливу роль відіграють підходи міжнародних організацій, які наголошують на необхідності забезпечення справедливої конкуренції у цифровому секторі [6].

Імплементація положень Актів про підзвітність App Store та Закону про свободу App Store (2025) пов'язана з низкою правових, економічних та технічних труднощів, оскільки регулювання цифрових платформ охоплює складні технологічні екосистеми, які швидко змінюються.

Однією з ключових проблем імплементації є визначення меж регуляторного втручання держави у діяльність приватних цифрових платформ. З одного боку, необхідно забезпечити конкуренцію та недопущення зловживання домінуючим становищем, а з іншого – не допустити надмірного адміністративного навантаження, яке може стримувати розвиток технологічних інновацій.

Крім того, складність становить контроль за виконанням норм у транснаціональних цифрових екосистемах, оскільки великі платформи функціонують одночасно у різних юрисдикціях.

Ще однією проблемою є потенційний опір з боку великих технологічних компаній, які можуть адаптувати свої бізнес-моделі таким чином, щоб формально відповідати вимогам законодавства, але зберігати фактичний контроль над ринком. У звітах антимонопольних органів зазначається, що подібні практики є типовими для цифрових монополій і потребують постійного оновлення регуляторних механізмів.

Разом з тим перспективи розвитку регулювання цифрових платформ є доволі значними. Очікується подальше посилення міжнародної координації у сфері антимонопольної політики, а також гармонізація підходів до регулювання цифрових ринків між різними країнами. Це може сприяти формуванню єдиних стандартів прозорості та конкуренції у цифровій економіці.

У довгостроковій перспективі такі законодавчі ініціативи можуть призвести до формування більш відкритих цифрових екосистем, де зменшиться залежність розробників від окремих платформ, а користувачі отримають ширший вибір цифрових продуктів. Міжнародні організації також прогнозують, що розвиток регулювання цифрових платформ стане одним із ключових напрямів антимонопольної політики у XXI столітті.

Отже, проблеми імплементації пов'язані переважно зі складністю регулювання високотехнологічних ринків та транснаціональним характером цифрових платформ, тоді як перспективи розвитку полягають у формуванні більш конкурентного, прозорого та збалансованого цифрового середовища.

Список використаних джерел

1. Competition in Digital Markets. *Organisation for Economic Co-operation and Development*. URL: https://bit.ly/4o6OECD (дата звернення: 07.05.2026).
2. Reports on Competition in Digital Markets and Digital Platforms. *U.S. Federal Trade Commission*. https://bit.ly/4o6FTCapp (дата звернення: 07.05.2026).
3. Antitrust Law and Big Tech Policy Materials. *U.S. House Judiciary Committee*. URL https://bit.ly/4o6HOUSE (дата звернення: 07.05.2026).
4. Digital Markets Act (DMA). *Documentation and Policy Framework*. *European Commission*. URL: https://bit.ly/4o6DMA (дата звернення: 07.05.2026).
5. Digital Economy Report. *United Nations Conference on Trade and Development (UNCTAD)*. URL: https://bit.ly/4o6UNCTAD (дата звернення: 07.05.2026).
6. Digital Economy and Market Competition Analysis. *International Monetary Fund (IMF)*. URL https://bit.ly/4o6IMF (дата звернення: 07.05.2026).

Ключові слова: цифрові платформи, App Store, антимонопольне регулювання, цифрова економіка, конкуренція, монополізація ринку, розробники програмного забезпечення, цифрові застосунки, державне регулювання, технологічні компанії

Keywords: digital platforms, App Store, antitrust regulation, digital economy, competition, market monopolization, software developers, mobile applications, state regulation, technology companies

Науковий керівник: доцент Чанишев Р.І.

TECH SOVEREIGNTY PACKAGE ЯК ІНСТРУМЕНТ ЗАБЕЗПЕЧЕННЯ ЦИФРОВОЇ НЕЗАЛЕЖНОСТІ ТА ТЕХНОЛОГІЧНОГО СУВЕРЕНІТЕТУ ЄС

Чанишев Рашид

кандидат юридичних наук, доцент,
доцент кафедри інформаційних технологій
Національного університету «Одеська юридична академія»

Як нещодавно стало відомо, Європейська комісія, після двох перенесень термінів протягом 2026 року, має намір 27 травня офіційно представити пакет заходів щодо забезпечення європейського технологічного суверенітету під назвою Tech Sovereignty Package [1].

Цей пакет має об'єднати низку вже існуючих законодавчих та стратегічних ініціатив, спрямованих на зменшення залежності держав Європейського Союзу від зовнішніх постачальників цифрових технологій, насамперед із США та Китаю, у сферах хмарних обчислень, штучного інтелекту, виробництва напівпровідників, розробки програмного забезпечення та забезпечення контролю над критично важливою цифровою інфраструктурою.

Це рішення пов'язане з тим, що залежність держав Європейського Союзу від зовнішніх постачальників цифрових технологій та послуг, за оцінками європейських інститутів, досягла критичного рівня. Так, у опублікованій 9 вересня 2024 року доповіді «Майбутнє європейської конкурентоспроможності» («The Future of European Competitiveness»), підготовленій колишнім головою

Європейського центрального банку та екс-прем'єр-міністром Італії Маріо Драгі (Mario Draghi), зазначалося, що Європейський Союз більш ніж на 80% залежить від іноземних держав у сфері цифрових технологій [2].

Обґрунтовану стурбованість викликають випадки політичного тиску та застосування санкційно-правових механізмів з боку іноземних держав, які вже мали місце. Наприклад, після введення США санкцій проти представників Міжнародного кримінального суду у 2025 році у зв'язку з розслідуваннями, що проводилися МКС щодо дій американських військових в Афганістані, у європейських ЗМІ з'явилися повідомлення про те, що судді та посадові особи Міжнародного кримінального суду, включаючи головного прокурора МКС Каріма Хана, зіткнулися з обмеженням доступу до фінансових і цифрових сервісів американських технологічних платформ [3].

Умови використання сервісів Microsoft та Google передбачають можливість одностороннього обмеження доступу до сервісів, а також передачу даних користувачів державним органам у випадках, передбачених законодавством США, включаючи CLOUD Act.

Крім того, умови використання та політики конфіденційності Microsoft і Google передбачають можливість передачі даних користувачів правоохоронним органам, судам та іншим державним структурам у випадках, передбачених законодавством, судовим наказом або офіційним державним запитом. У цьому контексті йдеться насамперед про правоохоронні органи та інші державні структури США, оскільки Microsoft і Google є американськими компаніями і підпадають під дію законодавства США, включаючи CLOUD Act.

Прийняття в США Закону CLOUD Act у 2018 році посилило дискусії щодо цифрового суверенітету Європейського Союзу, оскільки цей Закон надає американським правоохоронним органам можливість вимагати надання даних від американських технологічних компаній незалежно від місцезнаходження серверів та країни зберігання інформації [4].

В ЄС подібний механізм розглядається як один із факторів ризику для цифрового суверенітету Європейського Союзу.

Особливу увагу до цієї проблеми привернули рішення Суду Європейського Союзу у справах Schrems I та Schrems II, в рамках яких механізми передачі даних між ЄС та США – Safe Harbor та Privacy Shield – були визнані такими, що не забезпечують рівень захисту персональних даних, еквівалентний вимогам законодавства Європейського Союзу. Суд ЄС зазначив, що законодавство США дозволяє державним органам отримувати доступ до даних користувачів ЄС [5].

На цьому тлі в низці держав ЄС посилилися дискусії про ризики зберігання державних, фінансових та медичних даних в інфраструктурі американських хмарних провайдерів, оскільки дані компаній продовжують підпадати під дію законодавства США незалежно від фізичного розташування дата-центрів. Саме в цих умовах у Європейському Союзі почали активно розвиватися концепції sovereign cloud та європейської цифрової інфраструктури Gaia-X [6].

Прямим наслідком розвитку цих концепцій стала ініціатива американської компанії Amazon Web Services (AWS) щодо створення AWS European Sovereign Cloud. У 2024 році компанія офіційно оголосила про запуск окремої європейської хмарної інфраструктури, а в січні 2026 року цей проєкт було введено в експлуатацію. При цьому компанія особливо підкреслює, що обробка даних, управління інфраструктурою та технічний супровід здійснюються виключно персоналом, який перебуває на території Європейського Союзу, а дані клієнтів зберігаються та обробляються в центрах обробки даних (далі – ЦОД), розташованих виключно на території держав – членів ЄС.

Подібні обставини багато в чому сприяли формуванню в Європейському Союзі концепції технологічного та цифрового суверенітету, а також усвідомленню необхідності підготовки та подальшого прийняття згаданого вище Tech Sovereignty Package. Цей пакет включає кілька великих ініціатив:

Одним із ключових елементів пакету є Cloud and AI Development Act (CADA/CAIDA) – законопроєкт, спрямований на розвиток європейської хмарної інфраструктури та систем штучного інтелекту. Очікується, що цей акт визначить критерії так званої «суверенної хмари», встановить правила будівництва та експлуатації ЦОД, а також створить відповідні правові механізми.

Другою важливою частиною пакету має стати Chips Act 2.0 – продовження європейської політики у сфері напівпровідникової промисловості. Його метою є забезпечення розширення виробництва напівпровідників і мікросхем безпосередньо на території ЄС для зменшення залежності від зовнішніх постачальників у стратегічно важливих технологічних секторах.

Третьою складовою пакету є Стратегія відкритого вихідного коду (Open Source Strategy). У рамках цієї стратегії Європейський Союз розглядає програмне забезпечення з відкритим вихідним кодом як один з елементів цифрової незалежності, як спосіб зменшення технологічної залежності від великих іноземних постачальників програмного забезпечення, а також як засіб підвищення прозорості цифрової інфраструктури та забезпечення кібербезпеки.

Четвертою складовою пакету має стати стратегічна дорожня карта щодо цифровізації та застосування штучного інтелекту в енергетичному секторі, спрямована на безпечне впровадження європейських рішень у сфері штучного інтелекту в критично важливу енергетичну інфраструктуру з урахуванням вимог NIS2 та Cyber Resilience Act.

П'ятою складовою пакету має стати Quantum Act – законопроєкт, спрямований на розвиток квантових технологій у Європейському Союзі, створення механізмів їх фінансування та формування європейської квантової екосистеми за моделлю Chips Act.

За оцінками CEPA, досягнення повної технологічної незалежності обійдеться приблизно в 3,6 трлн євро, тоді як навіть забезпечення близько 20% частки ЄС у критично важливих технологічних секторах потребуватиме близько 680 млрд євро. [7].

Пакет «Tech Sovereignty Package» знаменує перехід Європейського Союзу від точкового регулювання цифрового ринку до системної політики технологічної автономії, що охоплює хмарну інфраструктуру, штучний інтелект, напівпровідникову промисловість та програмне забезпечення.

Формування цієї політики зумовлене високою залежністю ЄС від американських цифрових платформ та гіперскейлерів, ризиками екстериторіального застосування іноземного законодавства, насамперед CLOUD Act США, а також вразливістю критично важливої цифрової інфраструктури перед зовнішнім політичним та санкційно-правовим впливом. Разом з тим ефективність Tech Sovereignty Package залежатиме не тільки від нормативного регулювання, а й від здатності Європейського Союзу забезпечити масштабні інвестиції, сформувати власну конкурентоспроможну технологічну базу та зменшити залежність державного сектору від домінуючих іноземних гіперскейлерів.

Список використаних джерел

1. Tech Sovereignty Package and EU Digital Sovereignty Policy Materials. European Commission. URL: https://digital-strategy.ec.europa.eu/ (дата обращения: 13.05.2026).
2. Draghi M. The Future of European Competitiveness : report for the European Commission. European Commission, 2024. URL: https://commission.europa.eu/document/download/97e481fd-2dc3-412d-be4c-f152a8232961_en (дата обращения: 15.05.2026).
3. Microsoft's ICC email block triggers Dutch concerns over dependence on US tech. NL Times. URL: https://nltimes.nl/2025/05/20/microsofts-icc-email-block-triggers-dutch-concerns-dependence-us-tech (дата обращения: 14.05.2026).
4. Clarifying Lawful Overseas Use of Data Act (CLOUD Act). U.S. Congress. URL: https://www.congress.gov/bill/115th-congress/house-bill/4943 (дата обращения: 14.05.2026).
5. Judgment in Case C-311/18 Data Protection Commissioner v Facebook Ireland Ltd and Maximillian Schrems (Schrems II). Court of Justice of the European Union. URL: https://curia.europa.eu/juris/liste.jsf?num=C-311/18 (дата обращения: 15.05.2026).
6. Чанишев Р.І. Федеративна європейська інфраструктура даних Gaia-X як інструмент технологічного суверенітету Європи: досвід для України. *Системи та технології*. 2022. №2 (64). С. 48-55. DOI https://doi.org/10.32782/2521-6643-2022.2-64.6
7. Europe can't afford digital sovereignty. CEPA. URL: https://cepa.org/article/europe-cant-afford-digital-sovereignty/ (дата обращения: 17.05.2026).

Ключові слова: технологічний суверенітет ЄС, цифровий суверенітет, гіперскейлери, CLOUD Act, суверенна хмара

Keywords: EU technological sovereignty, digital sovereignty, hyperscalers, CLOUD Act, sovereign cloud

ПРОЦЕСУАЛЬНІ РИЗИКИ ЗАСТОСУВАННЯ ГЕНЕРАТИВНОГО ШІ У СУДОВОМУ ПРОВАДЖЕННІ ТА РЕГУЛЯТОРНА ВІДПОВІДАЛЬНІСТЬ АДВОКАТА: АНАЛІЗ НА ПРИКЛАДІ СПРАВИ VALU V MINISTER FOR IMMIGRATION AND MULTICULTURAL AFFAIRS (NO 2) [2025] FEDCFAMC2G 95 (АВСТРАЛІЯ)

Булгаков Богдан

студент 1-го курсу факультету судового та міжнародного права Національного університету «Одеська юридична академія»

Використання генеративного штучного інтелекту (GenAI) у юридичній практиці поступово поширюється та водночас породжує нові проблеми для системи судочинства.

Поряд із очевидними перевагами в автоматизації, ці інструменти породжують нові процесуальні ризики, що вимагають переосмислення меж професійної відповідальності адвоката. Показовим прикладом реалізації таких ризиків є рішення Федерального окружного та сімейного суду Австралії у справі Valu v Minister for Immigration and Multicultural Affairs (No 2) [2025] FedCFamC2G 95, яке яскраво демонструє негативні наслідки неконтрольованого використання GenAI під час підготовки процесуальних документів.

У цій справі адвокат заявника, готуючи письмові пояснення у міграційній справі, скористався ChatGPT для пошуку релевантних австралійських судових рішень. Внаслідок цього до суду було подано документи, які містили сімнадцять вигаданих посилань на рішення Федерального суду Австралії, а також фіктивні цитати з оскаржуваного рішення Трибуналу [4, с. 5–6].

Як згодом підтвердив сам представник, штучний інтелект лише «підсумував справи» для нього, а через брак часу та проблеми зі здоров'ям він інкорпорував отриманий матеріал у текст без належної перевірки [4, с. 8]. Таким чином, суд був уведений в оману, а ефективний розгляд справи зазнав суттєвої затримки.

Цей інцидент наочно ілюструє одну з ключових вразливостей сучасних великих мовних моделей – схильність до «галюцинацій», тобто продукування правдоподібних, структурованих, але абсолютно фіктивних відповідей. Практична нота Верховного суду Нового Південного Уельсу SC GEN 23 прямо визнає цей ризик, зазначаючи, що GenAI здатен генерувати фальшиві цитати, вигадані законодавчі посилання та інші неіснуючі джерела [1, с. 2].

Ситуація ускладнюється ще й тим, що сучасні дослідження фіксують доволі високий відсоток помилок: за даними опублікованих у 2025 році узагальнень, юридичні моделі генерують недостовірну інформацію приблизно в одному з шести запитів (16,7%), а точність їхніх відповідей на стандартні правові запити становить близько 82%, що суттєво нижче за показники досвідчених юристів (95%) [2, с. 10]. Такий розрив у точності створює критичну загрозу для неупередженості та законності судового процесу.

Процесуальні ризики застосування GenAI не обмежуються лише проблемою «галюцинацій». Не менш гострою є загроза порушення конфіденційності даних, адже інформація, введена у відкриті моделі, може бути використана для навчання алгоритмів і потенційно розкрита третім особам [1, с. 2].

У справі Valu цей аспект поглиблюється ще й тим, що адвокат здійснив пряме ex parte спілкування із судом, надіславши виправлені пояснення без відома та згоди представника протилежної сторони, чим порушив правило 22.5 Правил професійної поведінки соліситорів [4, с. 5, 7–8].

Це свідчить про те, що використання GenAI може опосередковано знижувати увагу юриста до базових етичних і процедурних стандартів, зокрема обов'язку не вводити суд в оману та не контактувати з ним одноособово з питань суті провадження. Відповідно, ризик витоку даних і ризик отримання недостовірної інформації тісно пов'язані між собою, адже обидва випливають із неконтрольованого введення процесуально значущих відомостей у публічні алгоритми.

Регуляторна реакція на подібні порушення поступово формалізується. В Австралії Практична нота SC GEN 23, яка набрала чинності з 3 лютого 2025 року, чітко встановлює, що результати GenAI не звільняють адвоката від професійних та етичних зобов'язань перед судом. Перевірка існування, точності та релевантності всіх посилань, включно з посиланнями на докази, не може здійснюватися виключно за допомогою ШІ-інструментів, а вимагає особистого фахового контролю [1, с. 4–5].

У справі Valu суддя Скарос, беручи до уваги щире каяття адвоката та його тривалий стаж (понад 27 років), усе ж ухвалила рішення про направлення матеріалів до Офісу уповноваженого з правових послуг Нового Південного Уельсу (OLSC) для проведення дисциплінарної оцінки. Суд наголосив на наявності сильного публічного інтересу в тому, щоб «покласти край» подібним випадкам, доки вони не стали системними [4, с. 10–12].

Аналіз справи Valu у світлі сучасних наукових підходів підтверджує, що жоден інструмент штучного інтелекту не здатен замінити професійне етичне судження адвоката. Як слушно підкреслює М. Леґґ, це судження є багатогранним – воно поєднує експертизу, досвід, етику та емоційний інтелект, – практичним (орієнтованим на розв'язання проблеми клієнта в межах закону) і цілісним (враховує ризики, обмеження та моральні аспекти), чого позбавлений будь-який алгоритм [3, с. 283].

Більше того, етичні зобов'язання адвоката перед судом та адміністрацією правосуддя є «найвищими» (paramount) і превалюють над будь-якими іншими інтересами, зокрема й над економією часу чи доцільністю використання новітнього програмного забезпечення [3, с. 281–282].

Отже, регуляторна відповідальність адвоката в епоху GenAI полягає не у відмові від новітніх технологій, а в їх компетентному, критичному осмисленні та

обов'язковому людському контролі. Нехтування цими вимогами неминуче призводить до процесуальних санкцій, підриву довіри до системи правосуддя та, у підсумку, до настання дисциплінарної відповідальності для самого фахівця.

Список використаних джерел

1. Supreme Court of New South Wales. Practice Note SC Gen 23 – Use of Generative Artificial Intelligence (Gen AI). 28 January 2025. URL: https://www.supremecourt.justice.nsw.gov.au/Documents/SC_Gen_23.pdf (дата звернення: 09.04.2026).
2. Joshi S. Generative AI in Legal Practice: Applications, Challenges, and Ethical Considerations. Preprints, 2025. DOI: 10.20944/preprints202506.0457.v1.
3. Michael Legg. "Better than a bot – instilling ethical judgement into the lawyers of the future in the age of AI" //Griffith Law Review. 2024. Vol. 33. No. 3. P. 273–293. DOI: 10.1080/10383441.2025.2493493.
4. Valu v Minister for Immigration and Multicultural Affairs (No 2) [2025] FedCFamC2G 95 (Federal Circuit and Family Court of Australia (Division 2), 31 January 2025). URL: https://www.austlii.edu.au/cgi-bin/viewdoc/au/cases/cth/FedCFamC2G/2025/95.html (дата звернення: 09.04.2026).

Ключові слова: генеративний штучний інтелект, судове провадження, процесуальні ризики, адвокат, регуляторна відповідальність, «галюцинації» ШІ, етика, фальшиві цитати, справа Valu, Австралія

Keywords: generative artificial intelligence, court proceedings, procedural risks, lawyer, regulatory responsibility, AI hallucinations, ethics, fake citations, Valu case, Australia

Науковий керівник: доцент Чанишев Р.І.

ЦИВІЛЬНО-ПРАВОВА ВІДПОВІДАЛЬНІСТЬ ВИРОБНИКА ЗА ШКОДУ, ЗАПОДІЯНУ СИСТЕМОЮ ДОПОМОГИ ВОДІЄВІ: НА МАТЕРІАЛІ СПРАВИ BENAVIDES v. TESLA, INC.

Иванченко Ева

студентка 1-го курсу факультету прокуратури та слідства (кримінальна юстиція) Національного університету «Одеська юридична академія»

Технології автоматизованого водіння та системи допомоги водієві (Advanced Driver Assistance Systems, ADAS) все активніше використовуються у сучасних транспортних засобах, що зумовлює появу нових питань щодо розподілу цивільно-правової відповідальності між виробником і водієм. Резонансна справа Benavides v. Tesla, Inc. (Case No. 1:21-cv-21940-BLOOM/Torres, U.S. District Court, S.D. Fla.) стала одним із перших резонансних рішень журі присяжних у США, у яких виробника Tesla, Inc. було визнано відповідальним за шкоду, заподіяну під час використання системи Autopilot.

Справа порушила дискусію про межі відповідальності виробника ADAS-системи за дефект конструкції та неналежне попередження споживача.

Фактичні обставини справи такі. 25 квітня 2019 року в Кі-Ларго (штат Флорида) Tesla Model S під керуванням Джорджа МакҐі з увімкненою системою Autopilot проїхала крізь Т-подібне перехрестя зі знаком «Стоп» та миготливим червоним сигналом на швидкості близько 100 км/год і врізалась у припаркований Chevrolet Tahoe, поряд з яким стояли Наїбель Бенавідес Леон та Ділон Ангуло. Внаслідок удару 22-річна Бенавідес Леон загинула, а Ангуло зазнав тяжких тілесних ушкоджень. Водій МакҐі визнав, що під час аварії відволікся на мобільний телефон. Позов проти Tesla, Inc. підданий до федерального суду у 2021 році; справу МакҐі позивачі врегулювали окремо. Журі присяжних ухвалило вердикт 1 серпня 2025 року після трьох тижнів судового розгляду.

Позивачі обґрунтовували відповідальність Tesla, Inc. трьома взаємопов'язаними теоріями продуктової відповідальності. По-перше, дефект конструкції: система Autopilot розроблена виключно для контрольованих автострад, однак Tesla не впровадила технічних обмежень, які унеможливлювали б активацію системи поза цими умовами.

По-друге, failure to warn - неналежне попередження: маркетингові матеріали Tesla та публічні заяви Ілона Маска про рівень 5 апаратного забезпечення і спроможності Autopilot формували у споживачів хибне уявлення про реальний ступінь автономності системи.

По-третє, недостатній моніторинг водія: система відстеження уваги базувалась виключно на вимірюванні крутного моменту на кермі, тоді як конкуренти вже використовували камери контролю водія.

На стадії попереднього розгляду суддя Бет Блум частково задовольнила клопотання Tesla про закриття справи. Вимога про дефект виготовлення була відхилена через відсутність доказів відхилення від запланованої конструкції. Вимога про недбале введення в оману (negligent misrepresentation) була відхилена з посиланням на прецедент Gilchrist Timber Co. v. ITT Rayonier, Inc., 696 So.2d 334 (Fla. 1997): Tesla не перебувала у прямих договірних відносинах із постраждалими, достатніх для встановлення деліктного обов'язку. Натомість до розгляду журі присяжних були допущені вимоги про дефект конструкції та failure to warn. У питанні штрафних збитків суд застосував Florida Statutes § 768.72 і дозволив цю вимогу після встановлення prima facie ознак того, що Tesla могла діяти з грубою недбалістю, свідомо ігноруючи відомі ризики безпеки.

Розмежування між дефектами конструкції та виготовлення має принципове значення як для американського, так і для українського права. Відповідно до Закону України «Про відповідальність за шкоду, завдану внаслідок дефекту в продукції» від 19.05.2011 №3390-VI, «продукція має дефект, якщо вона не забезпечує рівень безпеки, на який особа має право розраховувати, беручи до уваги всі обставини, зокрема: представлення продукції, включаючи її рекламу; використання продукції, яке можна обґрунтовано передбачити; час, коли продукція була введена в обіг» [1].

Ця формула стала центральною у справі Benavides v. Tesla: Tesla знала про ймовірність використання Autopilot поза межами автострад (передбачуваність), однак не здійснила конструктивних обмежень. Суд, застосовуючи флоридські стандарти Aubin v. Union Carbide Corp., 177 So.3d 489 (Fla. 2015), визнав, що журі могло дійти висновку про дефектність системи і за тестом споживацьких очікувань, і за балансом ризик-вигода.

К. Панкі у своєму дослідженні взаємодії ADAS та режиму цивільної відповідальності зазначає, що "problems raised by ADAS can largely be resolved within the civil liability regime through a traditional negligence analysis" [2, p. 141]. Однак авторка розмежовує ситуацію, коли виробник чесно інформує про обмеження системи, від тієї, де він активно формує хибні очікування. У справі Benavides v. Tesla аргумент про те, що водій зобов'язаний постійно контролювати дорогу, виявився недостатнім, адже журі взяло до уваги рекламні відеоматеріали Tesla, у яких показано водіїв, що взагалі не тримають рук на кермі. Це дозволило розглядати поведінку водія як передбачуване неправильне використання системи, спричинене самим виробником, - підхід, що відповідає характеристиці Панкі: "manufacturers' primary liability exposure is for collisions which could have been reasonably foreseen".

Вердикт розподілив відповідальність між Tesla (33%) та водієм МакҐі (67 %), застосувавши доктрину порівняльної вини (comparative fault). К. Думанчич та Д. Вулетич, досліджуючи відповідальність за шкоду від автономних транспортних засобів у контексті права ЄС, підкреслюють, що "Common regulatory approach of liability caused by vehicles is based on fault or negligence" [3, p. 64].

У флоридській справі порівняльна вина не виключила відповідальності виробника навіть тоді, коли водій сам визнав необережність. Це ключовий прецедент: відповідальність виробника є самостійною і паралельною відповідальності оператора, а не субсидіарною.

Призначення 200 млн доларів штрафних збитків стало наслідком доведення позивачами того, що Tesla знала про дефекти Autopilot, але продовжувала просувати систему з перебільшеними заявами про її спроможності. Думанчич та Вулетич наголошують на важливості того, щоб "simple mechanism for seeking compensation should be available to a person who has suffered harm caused by a defective product" - принцип, який Нова директива ЄС про відповідальність за дефектну продукцію 2024 року розширює, зокрема, на програмне забезпечення та AI-системи. Закон України «Про відповідальність за шкоду, завдану внаслідок дефекту в продукції» містить аналогічне положення: «відповідальність виробника по відношенню до особи, якій заподіяно шкоди, не може бути обмежена або виключена будь-якими договірними чи іншими положеннями». Тобто спроби Tesla обмежити відповідальність умовами угоди користувача принципово не звільнили б її від відповідальності і за українським правом.

20 лютого 2026 року суддя Блум відхилила клопотання Tesla про скасування вердикту або призначення нового розгляду справи, вказавши, що "evidence

admitted at trial more than supports the jury verdict" [4]. Справа передана Tesla до апеляційного суду U.S. Court of Appeals for the Eleventh Circuit. Незалежно від подальшого апеляційного результату Benavides v. Tesla вже стала орієнтиром для судів та виробників у питаннях: 1) де проходить межа між допустимим рекламуванням ADAS-системи та введенням в оману; 2) чи є моніторинг водія за крутним моментом достатнім заходом безпеки; 3) чи поглинає вина водія відповідальність виробника за дефектну конструкцію.

Таким чином, аналіз справи Benavides v. Tesla, Inc. дозволяє сформулювати такі висновки. По-перше, відповідальність виробника ADAS-системи може наставати навіть за умови встановленої вини водія, якщо система використовувалась у межах передбачуваного (хоч і не рекомендованого) сценарію - ключовий критерій передбачуваності закріплений і в українському законодавстві.

По-друге, маркетингові заяви про можливості системи формують самостійну підставу відповідальності у формі failure to warn і можуть бути підставою для штрафних збитків за умови доведення свідомого ігнорування ризиків.

По-третє, відповідальність за змішаної вини розподіляється пропорційно між виробником і водієм. і не виключається через необережність останнього. Ці висновки мають значення для розвитку вітчизняної доктрини щодо ADAS-технологій.

Список використаних джерел

1. Про відповідальність за шкоду, завдану внаслідок дефекту в продукції: Закон України від 19.05.2011 №3390-VI. URL: https://zakon.rada.gov.ua/laws/show/3390-17 (дата звернення: 04.05.2026).
2. Pankey C. Where Tech Meets Tort: A Survey of Geistfeld's Approach to Autonomous Vehicles and the Civil Liability Regime. Colorado Technology Law Journal. 2023. Vol. 21.1. P. 141–156. URL: https://ctlj.colorado.edu/wp-content/uploads/2023/08/Final-Pankey-6.27.23.pdf (дата звернення: 04.05.2026).
3. Dumančić K., Vuletić D. Liability for Damages Caused by Autonomous Driving Vehicles from the EU Law Perspective. EU and Comparative Law Issues and Challenges Series (ECLIC). 2025. Issue 9. P. 64–90. URL: https://ojs.srce.hr/index.php/eclic/article/download/38091/18173 (дата звернення: 04.05.2026).
4. Benavides v. Tesla, Inc., Case No. 1:21-cv-21940-BLOOM/Torres (U.S. District Court, Southern District of Florida). Jury Verdict, August 1, 2025; Order Denying Motion for Judgment as a Matter of Law, February 19, 2026. URL: https://law.justia.com/cases/federal/district-courts/florida/flsdce/1:2021cv21940/593426/612/ (дата звернення: 04.05.2026).

Ключові слова: цивільно-правова відповідальність виробника, ADAS, дефект конструкції, неналежне попередження, продуктова відповідальність

Keywords: civil liability of manufacturer, ADAS, design defect, failure to warn, product liability

Науковий керівник: доцент Чанишев Р.І.

UNITED STATES v. ADOBE, INC. (N.D. CAL.): ПРАВОВІ ПІДСТАВИ ПРЕТЕНЗІЙ FTC ТА DOJ ЩОДО ПРИХОВУВАННЯ УМОВ ПІДПИСКИ ТА УСКЛАДНЕННЯ ЇЇ СКАСУВАННЯ

Можаровська Поліна

студентка 1-го курсу факультету судового і міжнародного права Національного університету «Одеська юридична академія»

Модель підписки (subscription model) сьогодні широко використовується у сфері програмного забезпечення та цифрових сервісів і стала одним із основних способів надання цифрових послуг користувачам. Особливої уваги заслуговують випадки, коли великі технологічні компанії використовують недобросовісні практики при оформленні підписки та її скасуванні. Яскравим прикладом є справа United States v. Adobe, Inc., Maninder Sawhney, and David Wadhwani, що розглядалася в Окружному суді США Північного округу Каліфорнії та закінчилася мировою угодою на 150 мільйонів доларів США у березні 2026 року. У США правовою основою для дій FTC і DOJ став Закон про відновлення довіри онлайн-покупців (Restore Online Shoppers' Confidence Act, ROSCA), кодифікований у 15 U.S.C. §§8401-8405 [1].

Закон ROSCA, прийнятий у 2010 році, встановлює правила для операторів інтернет-продажів, які пропонують підписки. Згідно з прес-релізом Міністерства юстиції США, «ROSCA generally requires companies offering online subscriptions to clearly disclose important subscription information and to provide subscribers with simple ways to cancel» (ROSCA загалом вимагає від компаній, які пропонують онлайн-підписки, чітко розкривати важливу інформацію про підписку та надавати підписникам прості способи її скасування) [2].

Цей закон спрямований на захист споживачів від недобросовісних практик у сфері електронної комерції.

Первинний позов проти Adobe, Inc. було подано Федеральною торговою комісією США (Federal Trade Commission, FTC) спільно з Міністерством юстиції США (Department of Justice, DOJ) 17 червня 2024 року. У прес-релізі FTC зазначено, що Комісія діє проти Adobe «for deceiving consumers by hiding the early termination fee for its most popular subscription plan and making it difficult for consumers to cancel their subscriptions» (за обман споживачів шляхом приховування комісії за дострокове розірвання у найпопулярнішому плані підписки та утруднення процесу скасування підписок) [3].

Як зауважив Семюел Левін, директор Бюро захисту споживачів FTC: «Adobe trapped customers into year-long subscriptions through hidden early termination fees and numerous cancellation hurdles» (Adobe обманом заманювала клієнтів у річні підписки через приховані комісії за дострокове розірвання та численні перешкоди при скасуванні) [3].

Правовою основою для дій FTC є також Federal Trade Commission Act (15 U.S.C. §45), який забороняє «unfair or deceptive acts or practices in or affecting commerce» (нечесні або оманливі дії чи практики в комерції або які впливають на неї) [4].

Згідно з позовом, Adobe порушила як ROSCA, так і Federal Trade Commission Act, використовуючи систематичні практики обману споживачів.

Конкретні правові підстави претензій до Adobe полягають у двох основних групах порушень. Перша група - приховування умов підписки. Згідно з позовною заявою FTC, «when consumers purchase a subscription through the company's website, Adobe pushes consumers to its "annual paid monthly" subscription plan, pre-selecting it as a default… (коли споживачі купують підписку через вебсайт компанії, Adobe підштовхує споживачів до плану «щомісячна сплата за рік», попередньо обираючи його за замовчуванням. Adobe розміщує на видному місці «щомісячну» вартість плану під час реєстрації, але приховує комісію за дострокове розірвання, яка становить 50 відсотків від залишкових щомісячних платежів при скасуванні у перший рік).

Інформація про ETF подавалася «in small print or require consumers to hover over small icons to find the disclosures» (дрібним шрифтом або вимагала від споживачів навести курсор на малі іконки, щоб знайти роз'яснення).

Друга група порушень - перешкоджання скасуванню підписки. У прес-релізі DOJ від 13.03.2026 зазначено, що уряд звинувачує Adobe у тому, що компанія «thwarted subscribers' attempts to cancel, subjecting them to convoluted and inefficient cancellation processes filled with unnecessary steps, delays, unsolicited offers, and warnings» (зривала спроби підписників скасувати підписку, піддаючи їх складним та неефективним процесам скасування, наповненим непотрібними кроками, затримками, небажаними пропозиціями та попередженнями).

Згідно з прес-релізом FTC, «when consumers reach out to Adobe's customer service to cancel, they encounter resistance and delay from Adobe representatives. Consumers also experience other obstacles, such as dropped calls and chats, and multiple transfers» (коли споживачі звертаються до служби підтримки Adobe для скасування, вони зустрічають опір та затримки з боку представників Adobe. Споживачі також стикаються з іншими перешкодами, такими як обірвані дзвінки та чати, а також численні переадресації).

Мирова угода у справі United States v. Adobe, Inc. була оголошена 13 березня 2026 року. Згідно з прес-релізом DOJ, угода передбачає сплату Adobe «$75 million in civil penalties and offer customers $75 million in free services» (75 мільйонів доларів у вигляді цивільних штрафів та надання клієнтам безкоштовних послуг на 75 мільйонів доларів).

Як зауважив Бретт А. Шумейт, помічник Генерального прокурора, голова Цивільного відділу Міністерства юстиції: «American consumers deserve the right to make informed choices when deciding where to spend their hard-earned money. The Justice Department will strongly oppose any attempt to harm Americans with

deceptive and unfair business practices» (Американські споживачі заслуговують на право робити поінформований вибір при прийнятті рішення про те, де витрачати свої важко зароблені гроші. Міністерство юстиції буде рішуче протистояти будь-яким спробам нашкодити американцям через оманливі та нечесні бізнес-практики).

Окрім фінансових санкцій, мирова угода містить вимоги щодо майбутньої поведінки Adobe. Згідно з прес-релізом DOJ, «Adobe will be required to clearly disclose any Early Termination Fee and how the fee is calculated before enrolling customers in subscriptions. For any free trial lasting longer than seven days, Adobe must also remind customers before converting them into a paid subscription with an Early Termination Fee. Furthermore, Adobe will be required to provide its subscribers with easy ways to cancel their subscriptions» (Adobe буде зобов'язана чітко розкривати будь-яку комісію за дострокове розірвання та спосіб її розрахунку перед реєстрацією клієнтів на підписки. Для будь-якого безкоштовного пробного періоду тривалістю понад сім днів Adobe також зобов'язана нагадувати клієнтам перед перетворенням їх на платну підписку з комісією за дострокове розірвання. Крім того, Adobe буде зобов'язана надавати своїм підписникам прості способи скасування підписок) [5].

Слід зауважити, що подібні правові механізми захисту прав споживачів існують і в українському законодавстві. Зокрема, Цивільний кодекс України містить положення про публічний договір (стаття 633), договір приєднання (стаття 634) та свободу договору (стаття 627), які покладають підвищені вимоги добросовісності та прозорості на сторону, яка пропонує стандартизовані умови договору.

Однак специфічного законодавчого регулювання щодо онлайн-підписок, аналогічного американському ROSCA, в Україні наразі немає, що зумовлює актуальність вивчення міжнародного досвіду.

Підсумовуючи, можна зробити такі висновки. По-перше, справа United States v. Adobe, Inc. (N.D. Cal.) демонструє ефективність застосування Закону про відновлення довіри онлайн-покупців (ROSCA) як інструменту захисту прав споживачів від практик приховування умов підписки.

По-друге, правовими підставами претензій FTC та DOJ до Adobe стали порушення вимог щодо прозорого розкриття інформації про підписку (зокрема комісії за дострокове розірвання) та забезпечення простих способів її скасування.

По-третє, мирова угода на 150 мільйонів доларів США (75 млн цивільних штрафів та 75 млн безкоштовних послуг для клієнтів), оголошена 13.03.2026, встановлює прецедент для інших технологічних компаній, що працюють за моделлю підписки.

По-четверте, досвід цієї справи є особливо актуальним для України з огляду на розвиток ринку цифрових послуг та необхідність удосконалення національного законодавства про захист прав споживачів у сфері онлайн-підписок.

Список використаних джерел

1. Restore Online Shoppers' Confidence Act, 15 U.S.C. §§8401-8405. URL: https://www.ftc.gov/legal-library/browse/statutes/restore-online-shoppers-confidence-act (дата звернення: 30.04.2026).
2. Adobe Agrees to $150 Million Settlement and Injunction to Resolve Alleged Violations of the Restore Online Shoppers' Confidence Act: U.S. Department of Justice Press Release, March 13, 2026. URL: https://www.justice.gov/opa/pr/adobe-agrees-150-million-settlement-and-injunction-resolve-alleged-violations-restore-online (дата звернення: 30.04.2026).
3. FTC Takes Action Against Adobe and Executives for Hiding Fees, Preventing Consumers from Easily Cancelling Software Subscriptions : Federal Trade Commission Press Release, June 17, 2024. URL: https://www.ftc.gov/news-events/news/press-releases/2024/06/ftc-takes-action-against-adobe-executives-hiding-fees-preventing-consumers-easily-cancelling (дата звернення: 30.04.2026).
4. Federal Trade Commission Act, 15 U.S.C. §45. URL: https://www.ftc.gov/legal-library/browse/statutes/federal-trade-commission-act (дата звернення: 30.04.2026).
5. United States v. Adobe, Inc., Maninder Sawhney, David Wadhwani, No. 5:24-cv-03630-BLF, U.S. District Court for the Northern District of California. URL: https://www.courtlistener.com/docket/68860528/united-states-v-adobe-inc/ (дата звернення: 30.04.2026)

Ключові слова: захист прав споживачів, онлайн-підписка, ROSCA, FTC, DOJ, Adobe, мирова угода, комісія за дострокове розірвання, скасування підписки, оманлива практика

Keywords: consumer protection, online subscription, ROSCA, FTC, DOJ, Adobe, settlement, early termination fee, subscription cancellation, deceptive practice

Науковий керівник: доцент Чанишев Р.І.

СУДОВИЙ СПІР FTC ТА DOJ ПРОТИ ADOBE: ПІДСТАВИ ПРЕТЕНЗІЙ ДО ПРОЦЕДУРИ СКАСУВАННЯ ПІДПИСКИ

Носенков Денис

студент 1-го курсу факультету судового та міжнародного права Національного університету «Одеська юридична академія»

У червні 2024 року Міністерство юстиції США (DOJ) спільно з Федеральною торговою комісією (FTC) ініціювали судовий розгляд проти компанії Adobe Inc. та двох її топменеджерів – Девіда Вадхвані й Маніндера Соуні [1].

Ця справа є важливою для розуміння того, як правові системи реагують на цифрові маніпуляції, відомі як «темні патерни», тобто інтерфейсні рішення, що здатні спрямовувати користувача до невигідного для нього вибору [2].

Центральним елементом претензій стало використання моделі negative option marketing, за якої мовчання або бездіяльність споживача можуть фактично тлумачитися як згода на продовження платежів [3].

За твердженням регуляторів, Adobe просувала план Annual, Paid Monthly (APM), який зовні сприймався як щомісячна підписка, але фактично передбачав річне зобов'язання.

При цьому суттєва умова – штраф за дострокове розірвання договору (Early Termination Fee, ETF) – не розкривалася достатньо помітно для споживача. Зокрема, FTC зазначає, що інформація про штрафи та обмеження відображалася через малопомітні елементи інтерфейсу, що могло вводити користувачів в оману.

Окрему увагу в позові приділено процедурі скасування підписки. FTC та DOJ стверджують, що Adobe створила складний багатокроковий процес, який ускладнював припинення підписки та змушував користувачів проходити через додаткові сторінки, пропозиції знижок або інші перешкоди.

Такі дії можуть розглядатися як порушення вимог Restore Online Shoppers' Confidence Act (ROSCA), який зобов'язує продавців надавати споживачам простий механізм припинення регулярних платежів [4].

За твердженням регуляторів, сторінка скасування була прихована в налаштуваннях облікового запису, а сам процес супроводжувався численними спробами утримати користувача від відмови від послуг.

Правовою основою претензій також є Розділ 5(a) Акта про Федеральну торгову комісію, який забороняє недобросовісні або оманливі практики у сфері торгівлі [5].

У цьому контексті предметом правової оцінки стають не лише формальні умови договору, а й сама архітектура цифрового інтерфейсу, яка здатна впливати на поведінку користувача через приховані механізми психологічного стимулювання та утримання.

Важливою особливістю справи є питання персональної відповідальності менеджменту. Позов подано не лише проти Adobe Inc. як юридичної особи, а й проти конкретних посадових осіб – Девіда Вадхвані та Маніндера Соуні.

Регулятори стверджують, що вони знали про численні скарги користувачів, але не забезпечили належного спрощення процедури скасування підписки. На думку FTC та DOJ, поточна модель інтерфейсу забезпечувала компанії вищі показники утримання клієнтів (retention rates), що створювало економічну зацікавленість у збереженні складної процедури відмови від послуг.

Справу Adobe також розглядають у ширшому контексті боротьби FTC із «темними патернами» у цифровій економіці.

У 2024 році FTC просувала правило «Click to Cancel», відповідно до якого механізм скасування підписки має бути таким самим простим і доступним, як і механізм її оформлення [6].

Хоча подальша доля цього правила стала предметом окремих юридичних дискусій, сама справа Adobe демонструє тенденцію до посилення контролю за цифровими сервісами, які використовують психологічні та інтерфейсні механізми для утримання споживачів.

Таким чином, судовий спір FTC та DOJ проти Adobe свідчить про трансформацію сучасного підходу до захисту прав споживачів у цифровому середовищі. Предметом правового аналізу стають не лише умови електронного договору, а й сама структура цифрового інтерфейсу, здатна впливати на поведінку користувача. У цьому контексті «темні патерни» поступово розглядаються як окрема форма недобросовісної комерційної практики, що може породжувати як корпоративну, так і персональну юридичну відповідальність.

Водночас справа Adobe демонструє поступове розширення предмета правового контролю у сфері цифрової економіки. Якщо раніше основна увага приділялася змісту договору та формальному отриманню згоди користувача, то сучасна регуляторна практика дедалі більше враховує психологічні механізми впливу на поведінку споживача, включаючи складність навігації, інформаційне перевантаження, приховані умови та штучне ускладнення відмови від послуг. Це свідчить про формування нового підходу до оцінки цифрових сервісів, у межах якого інтерфейс платформи може розглядатися як самостійний об’єкт юридичної оцінки.

Крім того, зазначений судовий спір має значення не лише для американської, а й для міжнародної практики регулювання цифрових платформ. Підходи FTC щодо боротьби з «темними патернами» можуть впливати на подальший розвиток законодавства у сфері захисту прав споживачів, електронної комерції та цифрових послуг в інших державах. У перспективі це може сприяти появі більш жорстких вимог до прозорості цифрових інтерфейсів, процедур оформлення підписок та механізмів припинення онлайн-сервісів.

Список використаних джерел

1. Complaint for Permanent Injunction, Civil Penalties, and Other Relief : United States of America v. Adobe Inc., David Wadhwani, and Maninder Sawhney : Case No.5:24-cv-03630 /U.S. District Court for the Northern District of California. 2024. 17 June. 89 p. URL: https://www.ftc.gov/system/files/ftc_gov/pdf/adobe_complaint.pdf (дата звернення: 10.05.2026).
2. FTC Takes Action Against Adobe and Executives for Hiding Fees, Preventing Consumers from Easily Cancelling Software Subscriptions : Press Release / Federal Trade Commission. 2024. 17 June. URL: https://www.ftc.gov/news-events/news/press-releases/2024/06/ftc-takes-action-against-adobe-executives-hiding-fees-preventing-consumers-easily-cancelling (дата звернення: 10.05.2026).
3. Federal Trade Commission Act. 15 U.S.C. §45(a) – Unfair methods of competition unlawful. URL: https://www.law.cornell.edu/uscode/text/15/45 (дата звернення: 10.05.2026).
4. Restore Online Shoppers’ Confidence Act (ROSCA). 15 U.S.C. §§8401–8405. URL: https://www.ftc.gov/legal-library/browse/statutes/restore-online-shoppers-confidence-act (дата звернення: 10.05.2026).
5. Bringing Dark Patterns to Light: FTC Staff Report / Federal Trade Commission. Washington, DC: Bureau of Consumer Protection, 2022. 44 p. URL:

https://www.ftc.gov/reports/bringing-dark-patterns-light (дата звернення: 10.05.2026).

Ключові слова: темні патерни, Adobe, FTC, захист прав споживачів, закон ROSCA, процедура скасування підписки, цифрова економіка, SaaS

Keywords: dark patterns, Adobe Inc., Federal Trade Commission (FTC), Consumer protection, ROSCA, Subscription cancellation procedure, Digital economy, Software as a Service (SaaS)

Науковий керівник: доцент Чанишев Р.І.

ЕФЕКТ НАДМІРНОЇ ДОВІРИ ДО АЛГОРИТМІВ (ALGORITHM APPRECIATION) У КРИЗОВИХ СИТУАЦІЯХ: ЧОМУ КОРИСТУВАЧІ НЕХТУЮТЬ САМОЗБЕРЕЖЕННЯМ?

Смєлов Олександр

студент 1-го курсу факультету психології політології та соціології Національного університету «Одеська юридична академія»

Стрімке впровадження систем штучного інтелекту (ШІ) у критично важливі сфери поставило перед психологічною наукою принципово нове питання: що відбувається з прийняттям рішень людиною в умовах стресу та загрози за наявності алгоритмічної підказки? Відповідь на це питання дала резонансна серія експериментів Robinette et al. [1], у якій 26 учасників під час реальної евакуаційної процедури (штучне задимлення приміщення, справжня пожежна тривога) були змушені обрати між власним маршрутом виходу і вказівками евакуаційного робота.

Результат виявився несподіваним: всі 26 учасників пішли за роботом, попри те, що половина з них заздалегідь спостерігала, як цей самий робот невірно виконував навігаційне завдання. Найбільш показово: навіть коли робот вказував у темну кімнату без видимого виходу, більшість учасників не зверталась до знайомого безпечного виходу, через який вони увійшли. Цей феномен отримав позначення overtrust (надмірна довіра) і є екстремальним проявом ширшого явища - algorithm appreciation (схвалення алгоритму): готовності слідувати алгоритмічним рекомендаціям навіть всупереч власній думці та наявній контекстуальній інформації.

Romeo та Conti [2, с. 260] у систематичному огляді 35 емпіричних досліджень (за методологією PRISMA 2020, охоплює 2015-2025 рр.) встановлюють, що automation bias - схильність надмірно покладатися на автоматизовані рекомендації - має чіткі когнітивні підстави.

По-перше, задіяний механізм когнітивної ощадності: людина як «когнітивний скупець» (cognitive miser) у стресових умовах уникає ресурсозатратного критичного мислення і делегує рішення системі, яку сприймає як більш компетентну та об'єктивну.

По-друге, суттєву роль відіграє когнітивне навантаження (cognitive load): підвищений стрес, обмеженість часу та складність завдання посилюють схильність до надмірної довіри, оскільки зменшують здатність до незалежного оцінювання.

По-третє, ефект Даннінга-Крюгера: користувачі з помірним (а не нульовим або глибоким) рівнем знань про ШІ виявляються найбільш схильними до overtrust, оскільки переоцінюють своє розуміння можливостей системи. Дослідники також фіксують парадокс пояснень XAI: надання алгоритмічних пояснень не лише не знижує надмірну довіру, а нерідко посилює її, позаяк пояснення самі стають евристичними сигналами довіри [2, с. 260].

Кризова ситуація принципово змінює динаміку взаємодії між людиною та алгоритмом. Klingbeil, Grützner та Schreck [3, с. 3] у поведінковому експерименті (N = 319) встановили, що сам факт знання про алгоритмічне походження поради спонукає учасників слідувати їй навіть тоді, коли вона суперечить наявній контекстуальній інформації та їхній власній оцінці ситуації.

При цьому ситуаційна довіра (situational trust), сформована конкретним контекстом і завданням, виявляється значно потужнішим предиктором підпорядкування, ніж базова схильність до довіри (propensity to trust). Horowitz та Kahn [4, с. 3], провівши масштабний мультикраїновий експеримент (N = 9 000, дев'ять країн), виявили, що automation bias загострюється в умовах когнітивного тиску та обмеженості часу, характерних для кризи.

Водночас у ситуаціях надзвичайно високих ставок спостерігається протилежний ефект - algorithm aversion: відчуття особистої моральної відповідальності змушує людину відхиляти алгоритмічні рекомендації. Отже, феномен надмірної довіри в кризових ситуаціях найбільш виражений у «проміжній зоні»: коли ставки здаються значними, але не загрозливими для виживання - що є парадоксальним саме для сценаріїв самозбереження.

Подолання феномену надмірної довіри до алгоритмів є одним із ключових викликів для проектування безпечних людино-машинних систем.

Dogru та Krämer [5, с. 5] в експериментальному дослідженні (N = 529 шахістів із різним рівнем фахової майстерності) доводять, що ані рівень довіри до системи, ані самооцінка власної компетентності не є достатніми предикторами calibrated trust - відкаліброваної, адекватної реальним можливостям системи, довіри. Натомість вирішальним фактором є фахова експертиза у предметній сфері: саме доменна компетентність дозволяє виявляти помилки алгоритму і не підпорядковуватися хибним рекомендаціям.

Серед запропонованих механізмів протидії overtrust дослідники виокремлюють: «когнітивне тертя» (cognitive friction) - навмисне ускладнення інтерфейсу, що змушує користувача осмислено взаємодіяти із системою; заміну серійної архітектури прийняття рішень (де алгоритм надає пораду до рішення людини) на паралельну (де пораду показують після незалежного людського рішення); акцентування особистої відповідальності користувача за наслідки; а також спеціалізоване навчання, що формує адекватні ментальні моделі можливостей і

обмежень систем ШІ. Дослідження у сфері безпеки демонструють, що цілеспрямована освіта щодо ШІ суттєво знижує automation bias порівняно із загальною популяцією.

Таким чином, ефект надмірної довіри до алгоритмів у кризових ситуаціях є психологічно обґрунтованим феноменом, зумовленим когнітивною ощадністю, ситуаційним стресом та делегуванням відповідальності.

Парадокс полягає в тому, що саме кризові умови - підвищений стрес, дефіцит часу та когнітивне перевантаження - є найбільш сприятливим середовищем для automation bias, тоді як людина потребує максимальної незалежності суджень.

Нехтування самозбереженням на користь алгоритмічних підказок не є раціональним вибором, а проявом системного когнітивного збою, який потребує як дизайнерських (архітектура інтерфейсу, когнітивне тертя), так і освітніх (ШІ-грамотність, доменна експертиза) рішень.

Ці дані мають пряме значення для психологічної практики, підготовки фахівців, що працюють у надзвичайних ситуаціях, а також для правового регулювання відповідальності за рішення, прийняті в режимі людино-алгоритмічної взаємодії.

Важливим практичним наслідком досліджень automation bias є необхідність формування у користувачів навичок критичної взаємодії із системами штучного інтелекту, особливо у професіях, пов’язаних із ризиком для життя та безпеки людей.

У кризових умовах людина схильна сприймати алгоритмічну рекомендацію як більш точну, нейтральну та об’єктивну, ніж власне судження, навіть за наявності суперечливих ознак. Це створює потенційну небезпеку делегування системам ШІ фактичного контролю над поведінкою людини.

Тому поряд із технологічним удосконаленням алгоритмів особливого значення набувають психологічна підготовка користувачів, розвиток ШІ-грамотності та створення таких інтерфейсів, які не підміняють людське рішення, а стимулюють його усвідомлену перевірку.

Список використаних джерел

1. Robinette P., Li W., Allen R., Howard A.M., Wagner A.R. Overtrust of robots in emergency evacuation scenarios. 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI). 2016. P. 101–108. URL: https://dl.acm.org/doi/10.5555/2906831.2906851 (дата звернення: 09.04.2026).
2. Romeo G., Conti D. Exploring automation bias in human–AI collaboration: a review and implications for explainable AI. AI & SOCIETY. 2026. Vol. 41, No.1. P. 259–278. DOI: https://doi.org/10.1007/s00146-025-02422-7 (дата звернення: 09.04.2026).
3. Klingbeil A., Grützner C., Schreck P. Trust and reliance on AI – an experimental study on the extent and costs of overreliance on AI. Computers in Human Behavior. 2024.

Vol. 160. Article 108352. DOI: https://doi.org/10.1016/j.chb.2024.108352 (дата звернення: 09.04.2026).
4. Horowitz M.C., Kahn L. Bending the Automation Bias Curve: A Study of Human and AI-Based Decision Making in National Security Contexts. International Studies Quarterly. 2024. Vol. 68, No.2. Article sqae020. DOI: https://doi.org/10.1093/isq/sqae020 (дата звернення: 09.04.2026).
5. Dogru E.Ö., Krämer N.C. Investigating appropriate reliance on AI-Based decision support systems: the role of expertise, trust, and self-confidence. Journal of Decision Systems. 2025. Vol. 34, No.1. Article 2593251. DOI: https://doi.org/10.1080/12460125.2025.2593251 (дата звернення: 09.04.2026).

Ключові слова: надмірна довіра до алгоритмів, кризові ситуації, самозбереження, когнітивне навантаження, ситуаційна довіра, людино-машинна взаємодія, когнітивне тертя, ШІ-грамотність, прийняття рішень в умовах стресу

Keywords: excessive trust in algorithms, crisis situations, self-preservation, cognitive load, situational trust, human-machine interaction, cognitive friction, AI literacy, decision-making under stress

Науковий керівник: доцент Чанишев Р.І.

ВПРОВАДЖЕННЯ ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ В ЕНЕРГЕТИЧНОМУ СЕКТОРІ (НА ПРИКЛАДІ ЗЕЛЕНОГО ВОДНЮ)

Куклінова Тетяна

кандидат економічних наук, доцент,
доцент кафедри штучного інтелекту та математичного моделювння
Національного університету «Одеська юридична академія»

У сучасних умовах глобальної декарбонізації та цифрової трансформації економіки дуже важливим є саме розвиток технологій виробництва зеленого водню. Загальновідомо, що використання відновлюваних джерел енергії для виробництва водню сприяє зменшенню викидів парникових газів, підвищенню енергетичної безпеки та формуванню нової моделі сталого розвитку енергетичного сектору.

Для України розвиток технологій зеленого водню є перспективним напрямом післявоєнної відбудови та інтеграції до європейського енергетичного простору (Таблиця 1).

Україна має значний потенціал відновлюваної енергетики, зокрема сонячної та вітрової, що створює передумови для формування конкурентоспроможного ринку зеленого водню. Водночас слід зазначити, що значна частина проєктів у сфері зеленого водню в Україні нині перебуває у стані призупинення (рис. 1).

Таблиця 1

Фактор	Значення для України
Потенціал ВДЕ	Значні ресурси сонячної та вітрової енергетики для виробництва зеленого водню.
Енергетична безпека	Диверсифікація джерел енергії та зниження залежності від викопного палива.
Євроінтеграція	Відповідність цілям European Green Deal та інтеграція до енергетичного ринку ЄС.
Експортний потенціал	Можливість постачання зеленого водню до країн Європи.
Інфраструктура	Перспективи адаптації газотранспортної системи
Цифровізація	Впровадження AI, smart grid systems та цифрового енергоменеджменту.
Післявоєнна відбудова	Модернізація енергетичної інфраструктури

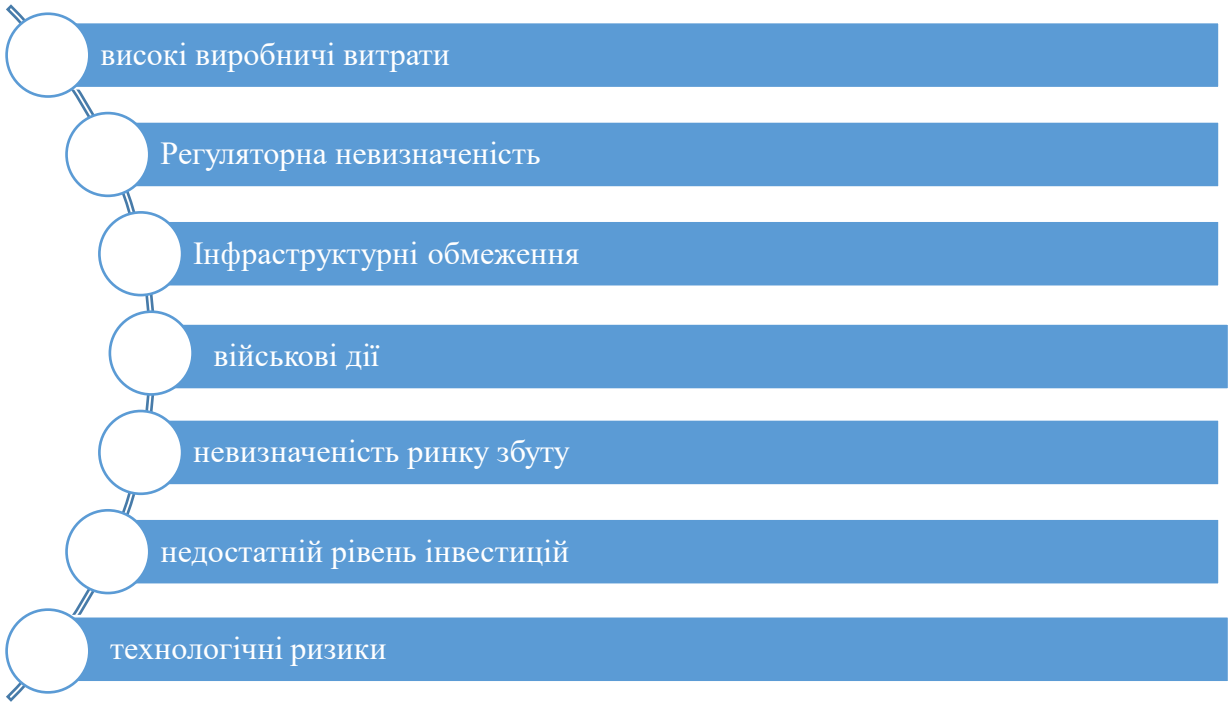


Рисунок 1 – Бар'єри розвитку технологій зеленого водню в Україні

Штучний інтелект активно інтегрується у процеси управління енергетичними системами, забезпечуючи автоматизацію, оптимізацію та прогнозування виробничих процесів [1]. У сфері зеленого водню ШІ-технології застосовуються для прогнозування обсягів генерації енергії з відновлюваних джерел, оптимізації роботи електролізерів, аналізу споживання ресурсів, управління логістичними процесами та моніторингу технічного стану обладнання. Використання алгоритмів машинного навчання дозволяє підвищити енергоефективність виробництва водню та зменшити експлуатаційні витрати.

Зазначимо, що особливий інтерес становить досвід Німеччини, яка є одним із лідерів розвитку водневої економіки в Європі. Досвід Germany може бути

адаптований в Україні для формування сучасної цифрової водневої екосистеми. Так, дним із найбільш успішних прикладів впровадження технологій зеленого водню в Germany є проєкт WUN H2 у місті Wunsiedel. Проєкт реалізований за участю Siemens Energy та функціонує на основі використання відновлюваних джерел енергії. Воднева установка потужністю 8,75 МВт забезпечує виробництво близько 1350 тонн зеленого водню щорічно. Особливістю проєкту є інтеграція цифрових технологій, систем автоматизованого управління, predictive analytics та елементів штучного інтелекту для оптимізації енергетичних процесів і підвищення ефективності виробництва водню [2]. Важливим прикладом трансформації енергетичної інфраструктури є Hamburg Green Hydrogen Hub, що створюється на території колишньої вугільної електростанції Moorburg. Проєкт передбачає використання електролізера Siemens Energy потужністю 100 МВт та інтеграцію smart technologies для управління водневою інфраструктурою. Основною метою проєкту є декарбонізація промисловості та портової інфраструктури Гамбурга, а також формування масштабної hydrogen ecosystem у Північній Німеччині [3].

Таким чином, впровадження процесів, пов’язаних із штучним інтелектом, у сфері зеленого водню сприяє підвищенню ефективності управління енергетичними ресурсами, розвитку інноваційних технологій та забезпеченню сталого розвитку енергетичного сектору

Список використаних джерел

1. Куклінова Т.В., Чепурна О.Є., Мельник Є.Б IT-інструменти в управлінні виробництвом зеленим воднем. Науковий вісник Полтавського університету економіки і торгівлі. Серія «Економічні науки». 3 (117), 2025. С.138.-145.
2. Siemens commissions one of Germany’s largest green hydrogen generation plants. *Siemens*. URL:https://press.siemens.com/global/en/pressrelease/siemens-commissions-one-germanys-largest-green-hydrogen-generation-plants.
3. The Hamburg Green Hydrogen Hub. https://www.hghh.eu/en

Ключові слова: зелений водень, штучний інтелект, цифрова трансформація, енергетичний сектор, AI, renewable energy, hydrogen economy

Keywords: green hydrogen, artificial intelligence, digital transformation, energy sector, renewable energy, hydrogen economy

СУЧАСНІ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В ЖИТТІ ЛЮДИНИ ТА СУСПІЛЬСТВА

Дмитрик Жан
студент 1-го курсу магістратури
факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Інформаційно-комунікаційні технології стали одним із ключових чинників розвитку суспільства, оскільки забезпечують швидке передавання, зберігання, оброблення та поширення інформації. У повсякденному житті вони використовуються для спілкування, навчання, роботи, отримання державних і комерційних послуг, організації дозвілля та доступу до знань. Поширення інтернету, мобільних пристроїв, соціальних мереж, месенджерів і хмарних сервісів змінило способи взаємодії між людьми, організаціями та державними інституціями.

Потреба в передаванні інформації існувала на всіх етапах розвитку людства. У різні історичні періоди для цього використовували усне мовлення, умовні знаки, піктографічне письмо, листування, друковані видання, телеграф, телефон, радіо й телебачення. Ці засоби поступово розширювали можливості комунікації, однак саме цифрові технології забезпечили принципово новий рівень швидкості, масштабності та доступності інформаційного обміну. Якщо традиційні канали комунікації були обмежені відстанню, часом передавання та матеріальними носіями, то цифрове середовище дає змогу обмінюватися даними майже миттєво.

Одним із найпомітніших проявів розвитку інформаційно-комунікаційних технологій є масове використання соціальних мереж і месенджерів. Вони забезпечують не лише приватне спілкування, а й публічну комунікацію, професійну взаємодію, поширення новин, рекламу, освітній контент і громадську активність. За даними порталу «Громадський простір», кількість користувачів соціальних мереж у світі перевищує 5,6 млрд, а значна частина інтернет-користувачів регулярно користується кількома соціальними платформами [2]. Це свідчить про те, що цифрові канали комунікації стали важливою частиною соціального життя.

Важливу роль у цифровому середовищі відіграють хмарні сервіси. Google Drive, Dropbox, Microsoft OneDrive, iCloud та інші подібні платформи дають змогу зберігати дані на віддалених серверах, синхронізувати файли між пристроями, спільно редагувати документи та зменшувати ризик втрати інформації через пошкодження локальних носіїв. Для організацій хмарні технології є основою віддаленої роботи, командної взаємодії, резервного копіювання та масштабування цифрової інфраструктури.

Окремого значення інформаційно-комунікаційні технології набули в освіті. Дистанційне навчання, електронні курси, відеоконференції, цифрові бібліотеки та системи управління навчанням розширили доступ до освітніх ресурсів і дали змогу підтримувати навчальний процес незалежно від фізичного місця

перебування учасників. Особливо помітною ця роль стала під час пандемії COVID-19, коли значна частина освітніх закладів, підприємств і установ була змушена перейти до дистанційних або змішаних форматів роботи.

Розвиток інформаційно-комунікаційних технологій також вплинув на економіку. Онлайн-торгівля, електронні платежі, цифровий маркетинг, віддалена зайнятість, фриланс і платформи для надання послуг стали звичними елементами економічної діяльності. Підприємства використовують цифрові інструменти для комунікації з клієнтами, автоматизації бізнес-процесів, аналізу даних і просування товарів або послуг. Завдяки цьому зростає значення цифрової грамотності, кібербезпеки та відповідального використання інформаційних ресурсів.

Водночас активне використання інформаційно-комунікаційних технологій створює низку проблем. До них належать ризики витоку персональних даних, поширення недостовірної інформації, цифрова залежність, інформаційне перевантаження, нерівний доступ до якісного інтернету та загрози кібербезпеці. Тому розвиток цифрового суспільства потребує не лише впровадження нових технологій, а й формування культури безпечної, критичної та відповідальної роботи з інформацією.

Отже, інформаційно-комунікаційні технології є важливим чинником трансформації повсякденного життя, освіти, економіки та суспільної взаємодії. Вони забезпечують швидкий доступ до інформації, підтримують дистанційні форми навчання і роботи, розширюють можливості комунікації та створюють умови для розвитку цифрових сервісів. Подальше використання ІКТ має поєднувати технологічні переваги з увагою до питань безпеки, достовірності інформації, цифрової грамотності та захисту персональних даних.

Список використаних джерел

1. Як первісні люди здобували та поширювали інформацію. URL: https://sites.google.com/view/history5-6/2/2-1/2-1-1
2. Громадянського простір. Соцмережі 2026: важлива статистика для комунікацій НУО. URL: https://www.prostir.ua/?kb=sotsmerezhi-2026-vazhlyva-statystyka-dlya-komunikatsij-nuo

Ключові слова: інформаційно-комунікаційні технології, інтернет, соціальні мережі, дистанційне навчання, цифровізація

Keywords: information and communication technologies, Internet, social networks, cloud services, distance learning, digitalization

Науковий керівник: доцент Лобода Ю.Г.

Секція 2. ІНФОРМАЦІЙНА БЕЗПЕКА ТА ЗАХИСТ ІНФОРМАЦІЇ: СОЦІАЛЬНИЙ, ПРАВОВИЙ І ТЕХНІЧНИЙ АСПЕКТ

РОЗСЛІДУВАННЯ КІБЕРЗЛОЧИНІВ У ІГРОВІЙ ІНДУСТРІЇ: ПРАКТИКА ФБІ IC3

Добровенко Даніїл

студент 1-го курсу факультету прокуратури та слідства Національного університету «Одеська юридична академія»

Розвиток цифрових технологій суттєво вплинув на сучасну індустрію відеоігор та сприяв зміні способів створення, поширення і використання ігрового контенту.. За даними аналітичної компанії Newzoo, світовий ринок ігор у 2024 році перевищив позначку у 180 мільярдів доларів США, а кількість активних гравців досягла понад 3,2 мільярда осіб [1]. Паралельно з цим зростанням формується специфічний кіберзлочинний простір, де щороку фіксуються мільярди збитків. Центр розгляду скарг щодо інтернет-злочинів (Internet Crime Complaint Center – IC3) при Федеральному бюро розслідувань США (ФБІ) є одним з небагатьох публічних інституційних механізмів, який систематично документує подібні злочини та формує правозастосовну практику їх розслідування.

Ігрова індустрія являє собою унікальне середовище для вчинення злочинів через поєднання трьох ключових чинників: масштабного обороту цифрових активів, анонімності учасників та міжнародного характеру платформ. Внутрішньоігрові предмети, валюта та облікові записи набули реальної ринкової вартості. Обліковий запис у популярній онлайн-грі може коштувати від кількох десятків до кількох тисяч доларів, а окремі ігрові предмети на вторинних ринках оцінюються у десятки тисяч доларів. Це перетворює ігрові платформи на привабливий об’єкт злочинних посягань.

Щорічні звіти ФБІ IC3 виокремлюють кілька категорій кіберзлочинів, безпосередньо пов’язаних з ігровою сферою. По-перше, це шахрайство з ігровими акаунтами (account takeover fraud) – несанкціонований доступ до облікового запису гравця з метою викрадення внутрішньоігрових активів або конфіденційних платіжних даних. По-друге, поширеною схемою є продаж неіснуючих або свідомо перебільшених ігрових товарів через торгівельні майданчики та Discord-сервери – так зване «скаміювання» (scamming). По-третє, IC3 фіксує випадки використання ігрових платформ як інструменту відмивання грошей через конвертацію незаконно отриманих коштів у внутрішньоігрову валюту з подальшим її обміном або перепродажем. За звітом ФБІ IC3 за 2023 рік, загальні втрати від кіберзлочинів у США перевищили 12,5 мільярда доларів [2].

Особливої уваги заслуговує феномен шахрайства з використанням ігор типу «play-to-earn» («грай і заробляй»). У березні 2023 року ФБІ IC3 видав спеціальне публічне попередження (Public Service Announcement – PSA), де зафіксував:

злочинці створюють підроблені ігрові застосунки, які начебто нараховують криптовалюту за ігровий процес. Жертві пропонують підключити криптогаманець і поповнити його – стверджуючи, що більший баланс приносить більші ігрові нагороди. Після того як жертва перестає поповнювати рахунок, злочинці за допомогою вбудованого шкідливого програмного забезпечення повністю спустошують гаманець [3]. Ця схема є наочним прикладом конвергенції ігрового та криптовалютного шахрайства.

Методологія розслідування кіберзлочинів у ігровій сфері, яку застосовує ФБІ IC3, базується на комплексному підході. По-перше, IC3 виступає агрегатором скарг: отримані повідомлення аналізуються на предмет спільних ознак і передаються до відповідних правоохоронних органів – підрозділів ФБІ, Секретної служби США чи прокуратур штатів. По-друге, IC3 активно співпрацює з великими ігровими компаніями (Sony PlayStation Network, Valve Steam, Microsoft Xbox) у рамках механізмів публічно-приватного партнерства, що дозволяє оперативно отримувати технічні логі та дані про транзакції без необхідності тривалих судових процедур. По-третє, IC3 видає публічні попередження (PSA), які виконують важливу превентивну та інформаційну функцію.

Ключовим викликом для правоохоронців є юрисдикційна фрагментація: серверна інфраструктура ігрових платформ може перебувати в одній країні, жертва – в іншій, а злочинець – у третій. ФБІ IC3 вирішує цю проблему шляхом координації з Інтерполом, Євроюстом та національними кіберполіціями в рамках спільних міжнародних операцій. Цифрова криміналістика у таких справах спирається на специфічний інструментарій: аналіз блокчейн-транзакцій при розслідуванні злочинів, пов'язаних з криптовалютними ігровими платформами, дозволяє простежити рух активів навіть через кілька рівнів анонімізації. Дослідження IP-адрес при підключенні до ігрових серверів та аналіз поведінкових патернів гравця стають доказовою базою у судових провадженнях [4].

Для України питання розслідування кіберзлочинів у ігровій сфері набуває особливої актуальності з кількох причин. По-перше, Україна посідає значне місце на світовому ринку ігрової розробки: такі студії, як GSC Game World, 4A Games та Frogwares, мають міжнародну аудиторію, що потенційно робить вітчизняні сервери привабливою мішенню. По-друге, Кіберполіція України активно розвиває компетенції у сфері протидії злочинам в інтернеті та в рамках двосторонніх угод обмінюється оперативними даними з ФБІ, зокрема з підрозділами, пов'язаними з IC3 [5]. По-третє, прийнятий у 2022 році Закон України «Про віртуальні активи» сформував базову нормативну основу цифрових активів, що формує нормативну основу для їх захисту засобами кримінального права [6].

Варто зауважити, що кримінальне законодавство України не містить окремого складу злочину щодо шахрайства в ігровому середовищі. Подібні діяння кваліфікуються за загальними нормами: статтею 190 КК України (шахрайство), статтею 361 КК України (несанкціоноване втручання у роботу комп'ютерів) та статтею 209 КК України (легалізація майна, одержаного злочинним шляхом).

Відсутність спеціальних норм ускладнює правову кваліфікацію, особливо у справах, де об'єктом посягання є виключно внутрішньоігрові активи без їх прямої прив'язки до фіатної валюти [7].

Узагальнюючи практику ФБІ IC3 у сфері розслідування ігрових кіберзлочинів, можна виокремити низку рекомендацій для вдосконалення вітчизняного правозастосування: формування спеціалізованих слідчих підрозділів із підготовкою у сфері ігрової економіки та цифрової криміналістики; запровадження обов'язкових протоколів взаємодії між ігровими компаніями та правоохоронними органами за зразком моделі IC3; систематизація скарг громадян щодо ігрового шахрайства на базі єдиної платформи з подальшою передачею матеріалів до Кіберполіції та Офісу Генерального прокурора України. Таким чином, ФБІ IC3 демонструє ефективну модель інституційного реагування, що поєднує збір даних, міжвідомчу координацію та публічну превенцію. Адаптація цього досвіду в українських реаліях є необхідною умовою ефективного захисту цифрових прав громадян та учасників ігрового ринку.

Список використаних джерел

1. Newzoo Global Games Market Report 2024. URL: https://newzoo.com/resources/trend-reports/newzoo-global-games-market-report-2024-free-version (дата звернення: 06.05.2026).
2. Federal Bureau of Investigation. 2023 Internet Crime Report. Internet Crime Complaint Center (IC3). Washington, D.C., 2024. URL: https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf (дата звернення: 06.05.2026).
3. Federal Bureau of Investigation (FBI). Criminals Steal Cryptocurrency through Play-to-Earn Games: Public Service Announcement I-030923-PSA. Internet Crime Complaint Center (IC3). 2023. URL: https://www.ic3.gov/PSA/2023/PSA230309 (дата звернення: 06.05.2026).
4. Federal Bureau of Investigation. FBI Guidance for Cryptocurrency Scam Victims. Internet Crime Complaint Center (IC3). 2023. URL: https://www.ic3.gov/PSA/2023/psa230824 (дата звернення: 06.05.2026).
5. Кіберполіція України. Офіційний сайт. URL: https://cyberpolice.gov.ua (дата звернення: 06.05.2026).
6. Про віртуальні активи: Закон України від 17.02.2022 № 2074-IX. Відомості Верховної Ради України. 2022. №28. Ст. 217. URL: https://zakon.rada.gov.ua/laws/show/2074-20 (дата звернення: 06.05.2026).
7. Кримінальний кодекс України від 5 квітня 2001 р. № 2341-III. Статті 190, 209, 361. URL: https://zakon.rada.gov.ua/laws/show/2341-14 (дата звернення: 06.05.2026).

Ключові слова: кіберзлочинність, ігрова індустрія, ФБІ IC3, розслідування, цифрова криміналістика, шахрайство, онлайн-ігри, play-to-earn

Keywords: cybercrime, gaming industry, FBI IC3, investigation, digital forensics, fraud, online games, play-to-earn

Науковий керівник: доцент Чанишев Р.І.

SIEM-СИСТЕМИ В ПІДТРИМЦІ ПРИЙНЯТТЯ РІШЕНЬ З УПРАВЛІННЯ РИЗИКАМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Кухаренко Сергій

кандидат технічних наук, доцент,
доцент кафедри кібербезпеки
Національного університету «Одеська юридична академія»

Інформації про загрози сьогодні більше, ніж будь-коли, – і рішення приймаються не краще від цього. Системи управління інформацією про безпеку та подіями (SIEM) агрегують, нормалізують і корелюють потоки подій безпеки, але на виході дають не висновок, а черговий список: впорядкований, відфільтрований і все одно надто довгий, щоб бути корисним без додаткової формальної структури. Кожний елемент цього списку є кластером оповіщень, що вимагає вибору між негайним реагуванням, постановкою до черги або пригніченням як хибного спрацювання, – і цей вибір мусить бути зробленим швидко, в умовах конкурентних вимог до обмеженого аналітичного ресурсу. Смірнова та ін. [1] дослідили актуальний стан SIEM-систем в українських організаціях в умовах воєнного часу і задокументували шестиетапний конвеєр обробки – фільтрація, агрегація, нормалізація, збір, кореляція, візуалізація, – а також практичний розрив, що виникає, коли попередньо налаштовані правила кореляції перестають відповідати реальному ландшафту загроз. Виявлено, що надлишкове пригнічення оповіщень веде до пропуску інцидентів, недостатнє – виснажує аналітичний ресурс шумом. Задача підтримки прийняття рішень не є наступною після SIEM – вона вбудована в її конфігурацію.

Система підтримки прийняття рішень (СППР) у сфері інформаційної безпеки – це будь-яка формальна процедура, що відображає спостережувані індикатори на рекомендовані управлінські дії разом з оцінкою наслідків прийняття або відкладення кожного рішення. Коли такими індикаторами є кластери оповіщень SIEM, змінними рішення стають призначення аналітика, ескалація оповіщення та активація технічних контрзаходів, а моделлю наслідків – очікуваний залишковий ризик після кожного варіанту реагування. Складність полягає в тому, що в будь-який момент часу існує багато конкуруючих варіантів реагування, які борються за обмежені ресурси: години роботи аналітиків, черговий бюджет, вікна технічного обслуговування мережі. Структурована оптимізаційна модель є не надмірністю, а необхідною умовою для того, щоб зробити цю задачу прийняття рішень розв'язуваною.

Формалізація розподілу ресурсів на основі оповіщень у вигляді задачі лінійного програмування дозволяє застосовувати стандартний математичний апарат безпосередньо до потоку SIEM. Жданова, Шевченко, Спасітєлєва та Сокульський [2] формулюють задачу як мінімізацію сукупних витрат на захист за обмежень щодо залишкового ризику, бюджету, часу та технічної спроможності, з

двійковими змінними рішення, що вказують на активацію кожного захисного заходу. Модель будується на основі SWOT-аналізу ризиків і завершується верифікацією допустимості. Обчислювальні експерименти показують, що оптимальний розв'язок забезпечує вимірне зниження витрат при збереженні заданого рівня захисту. Цей результат безпосередньо переноситься на SIEM-орієнтовані процеси: кожний кластер оповіщень відображається на кандидатний захисний захід з оціненим коефіцієнтом зниження ризику, отриманим з даних про минулі інциденти. Для розподілених IT-середовищ, де сенсори SIEM розгорнуті в кількох організаційних сегментах з різнорідними профілями загроз і міжсегментними шляхами поширення, Палько та Мирутенко [3] розроблюють метод комплексної оцінки ризиків, що враховує залежності між сегментами та інформаційну асиметрію між локальним виявленням і централізованою кореляцією. Отримані багатокритеріальні показники ризику сумісні з лінійною та мішано-цілочисельною оптимізацією над розподіленими управлінськими діями.

Лінійна оптимізація передбачає, що коефіцієнт зниження ризику кожного захисного заходу можна оцінити як стійкий скаляр – припущення, що виконується за добре відомого ландшафту загроз, але порушується під час нових атакових кампаній або коли дії реагування мають нелінійні ефекти взаємодії. Шевченко, Жданова, Складанний та Петренко [4] вирішують цю проблему, застосовуючи нечіткі когнітивні карти (НКК) для підтримки прийняття рішень щодо реагування на інциденти. НКК подає концепти безпеки – ступінь критичності оповіщення, доступність системи, спроможність атакуючого, вартість усунення, стан відповідності – у вигляді вузлів з ваговими орієнтованими ребрами, що кодують причинно-наслідкові зв'язки. Симуляція поширення змін через карту дозволяє особі, яка приймає рішення, оцінити непрямі наслідки кандидатного варіанту реагування на загальний стан безпеки без припущення про лінійну адитивність внесків ризику. В архітектурі, інтегрованій із SIEM, НКК виконують функцію шару сценарного міркування: кластери оповіщень SIEM відображаються на початкову активацію вузлів, а симуляція НКК проектує траєкторію стану безпеки за кожним варіантом реагування.

Таблиця 1. Багаторівнева архітектура підтримки прийняття рішень на основі SIEM

Рівень підтримки рішень	**Інструмент**
Збір та нормалізація подій	Шестиетапний конвеєр обробки SIEM
Пріоритизація оповіщень	Лінійна оптимізація розподілу ресурсів захисту
Оцінка ризиків розподіл. систем	Метод з урахуванням міжсегментних залежностей
Сценарне міркування	Симуляція нечітких когнітивних карт
Управлінське вирівнювання	Інструмент багатостандартної оцінки кіберзрілості

Впровадження описаної архітектури СППР в рамках IT-управління потребує відповідності визнаним стандартам менеджменту. Шевченко, Жданова та Кія [5] розроблюють напівавтоматизований інструмент багатостандартної оцінки кіберзрілості організації, що одночасно оцінює відповідність NIST CSF 2.0, ISO/IEC 27001:2022, COBIT 2019 та CIS Controls v8. Інструмент відображає кожну область контролю на показник зрілості та виявляє прогалини, де охоплення SIEM відсутнє або недостатнє. Інтеграція статистики оповіщень SIEM з показниками зрілості формує управлінську панель моніторингу, в якій якість прийняття рішень у функції безпекових операцій відстежується як явна організаційна метрика – не лише кількість подій і частка хибних спрацювань, а й ступінь відповідності конвеєру «оповіщення – реагування» цілям управління ризиками, зафіксованим у стандартах. Показники зрілості слугують також вхідними даними для оптимізаційної моделі: контролі, оцінені нижче цільового рівня зрілості, отримують підвищений коефіцієнт ризику, перерозподіляючи ресурси захисту на їх усунення. Замкнений цикл – спостереження SIEM, оптимізація за ризиком, сценарне міркування, управлінський зворотний зв'язок – утворює інтегровану архітектуру підтримки прийняття рішень, окремі компоненти якої валідовані в наведених джерелах, а їх спільне розгортання залишається відкритою інженерною задачею.

Список використаних джерел

1. Смірнова Т. І., Константинова Л. В., Конопліцька-Слободенюк О. К., Козлов Я. В., Кравчук О. В., Козірова Н. О., Смірнов О. А. Дослідження сучасного стану SIEM-систем. Кібербезпека: освіта, наука, техніка. 2024. Т. 1, № 25. С. 6-18. DOI: 10.28925/2663-4023.2024.25.618.
2. Жданова Ю. Д., Шевченко С. М., Спасітєлєва С. О., Сокульський О. Є. Прийняття рішень на основі лінійної оптимізації у процесі управління ризиками інформаційної безпеки. Кібербезпека: освіта, наука, техніка. 2024. Т. 1, № 25. С. 330-343. DOI: 10.28925/2663-4023.2024.25.330343.
3. Палько Д., Мирутенко Л. Метод комплексної оцінки ризиків кібербезпеки в розподілених інформаційних системах. Кібербезпека: освіта, наука, техніка. 2024. Т. 2, № 26. С. 487–502. DOI: 10.28925/2663-4023.2024.26.731.
4. Шевченко С. М., Жданова Ю. Д., Складанний П. М., Петренко Т. М. Нечіткі когнітивні карти як інструмент візуалізації сценаріїв реагування на інциденти в системах безпеки. Кібербезпека: освіта, наука, техніка. 2024. Т. 2, № 26. С. 419-429. DOI: 10.28925/2663-4023.2024.26.707.
5. Шевченко С. М., Жданова Ю. Д., Кія О. В. Напівавтоматизований інструмент багатостандартної оцінки кіберзрілості організації на основі NIST CSF 2.0, ISO/IEC 27001:2022, COBIT 2019 та CIS Controls v8. Кібербезпека: освіта, наука, техніка. 2025. Т. 3, № 31. С. 43-60. DOI: 10.28925/2663-4023.2025.31.1004.

Ключові слова: кібербезпека, підтримка прийняття рішень, SIEM-системи, оптимізація, IT-управління, аналіз ризиків, інформаційна безпека

Keywords: cybersecurity, decision support, SIEM systems, optimisation, IT governance, risk analysis, information security

СИСТЕМА АДАПТИВНОГО КЕРУВАННЯ ПОВЕРХНЕЮ АТАКИ БЕЗДРОТОВИХ МЕРЕЖ З ВИКОРИСТАННЯМ ІНТЕЛЕКТУАЛЬНОЇ ОРКЕСТРАЦІЇ ТРАФІКУ

Бойко Віктор

кандидат технічних наук, доцент,
доцент кафедри кібербезпеки
Національного університету «Одеська юридична академія»

Бездротові технології, охоплюючи широкий спектр стандартів від Wi-Fi різних поколінь до технологій мобільного зв'язку 3G, 4G та 5G, стали фундаментом інформаційного суспільства. Швидке зростання кількості підключених пристроїв, зокрема в сегментах Інтернету речей (IoT), призвело до розширення поверхні атаки [1]. Для сучасного зловмисника, особливо якщо він використовує елементи штучного інтелекту, компрометація мережевої інфраструктури є завданням, що не потребує високої кваліфікації [2], [3].

Ситуація ускладнюється тим, що для здійснення атаки зловмиснику достатньо потрапити до зони покриття радіосигналом у зоні дії точки доступу - при цьому не обов'язково навіть з'являтися там самому, можна доставити туди прилад, який використовується для злому. Тому будь-яке мобільне або бездротове підключення - це потенційний вектор вторгнення. З огляду на інтеграцію таких мереж з критичною інформаційною інфраструктурою, ризики несанкціонованого доступу, перехоплення даних та впровадження шкідливого коду вже можуть бути пов'язані з ризиком завдання серйозної шкоди та компрометації систем критичної інфраструктури [4], [5].

Традиційні засоби захисту - фаєрволи зі статичними правилами контролю доступу (ACL), та системи глибинного аналізу трафіку (DPI), виявляються концептуально невідповідними умовам бездротового середовища. По-перше, захисні механізми, які вимагають ресурсомісткого DPI-аналізу, створюють «вузькі місця», що критично знижує пропускну здатність та підвищує рівень затримок. По-друге, архітектурна динаміка бездротових систем – постійні переходи абонентів між сотами (handover), динамічна адресація та віртуалізація мережевих функцій – робить статичну конфігурацію захисту неактуальною вже за кілька секунд після її застосування.

Для вирішення проблем захисту пропонується розгортання та впровадження системи адаптивного керування поверхнею атаки бездротових мереж (Adaptive Attack Surface Management System - AASMS) з використанням інтелектуальної оркестрації трафіку (Intelligent Traffic Orchestrator ITO).

Система складається з трьох основних компонентів та використовує принцип архітектурної ізоляції, який дозволяє винести деструктивну взаємодію з атакуючими зловмисниками в окремий сегмент пасток, не перешкоджаючи легітимному трафіку. Система включає в себе систему тіньових точок доступу,

механізм ідентифікації та ізоляції, та систему інтеграції зі штатними засобами (SIEM/IDS/IPS) захисту периметру мережі.

Тіньовими точками доступу можуть бути як фізично окремі пристрої, або віртуальні "двійники" на тих самих фізичних точках. Кожен із варіантів має свої переваги та недоліки. Використання фізично окремих пристроїв потребує додаткових вкладень у інфраструктуру. Якщо використовувати віртуальні пристрої, це зумовить необхідність забезпечення ізоляції від робочої мережі. На нашу думку, перший варіант дозволяє отримати більше переваг при невеликих вкладеннях - можна використовувати в тому числі застаріле обладнання, що не використовується у критичних бізнес-процесах. Фізично окремий пристрій також має переваги у разі дуже серйозної атаки. З одного боку, тарпітинг при правильній організації дозволяє справлятися з великими обсягами запитів, з іншого ніколи не можна виключати, що обсяг запитів перевищить можливості тіньової точки доступу. Навіть у цьому випадку система залишається стійкою - тимчасовий вихід з ладу помилкової точки доступу не позначається на загальному функціонуванні робочої частини мережі.

Наявність тіньових точок доступу дозволяє детектувати девіантну поведінку користувача точки доступу. Легітимний вузол (наприклад, промисловий IoT-датчик) функціонує згідно зі жорстким профілем активності та ніколи не звертається до неактуальних системних ресурсів чи адміністративних API. Будь-яка спроба звернення до точки доступу, а тим більше запит потенційних векторів експлуатації (наприклад, /etc/passwd, /.env чи /config.php) автоматично інтерпретується як ознака спроби несанкціонованого доступу до системи. Це забезпечує вищий рівень точності виявлення, ніж традиційні сигнатурні методи, мінімізуючи рівень хибнопозитивних спрацьовувань. Метод дозволяє детектувати навіть легітимні пристрої, які оновлюють прошивку, або мають тимчасові сплески активності - що є критично важливим аспектом для промислового IoT.

Після детектування зловмисника система активує механізм примусового гальмування - тарпитування, або tarpitting. На відміну від миттєвого скидання з'єднання, тарпітування примушує атакувальний вузол утримувати сесію, споживаючи обчислювальні ресурси. Це з одного боку змушує атакуючого витрачати свої ресурси на підтримку свідомо безглуздого сеансу зв'язку, з іншого боку уповільнює атаку, що дозволяє отримати часовий лаг для запуску та спрацьовування штатних систем захисту периметра. Додатковою перевагою системи є те, що тарпітинг додатково знижує мережевий трафік, уповільнюючи обмін інформацією з атакуючим та виносячи його в область тіньових ресурсів, що економить та вивільняє легітимні ресурси системи.

Система захисту взаємодіє зі штатними засобами (SIEM/IDS/IPS) які на підставі її рішень оновлюють політики безпеки, та актуалізують списки блокування для мережевих рівнів (L4/L7), перекриваючи доступ виявленим зловмисникам.

Для взаємодії оператора з системою передбачений інтерфейс, що дозволяє в критичних сценаріях переводити мережу в режим ешелонованої оборони,

забезпечуючи повний контроль над критичною інфраструктурою під час активної фази вторгнення. Він потребує лише підтвердження стратегічних змін політик, а не тактичного блокування. Також він передбачає переведення системи в режим "підвищеної протидії", та ручну корекцію політик хибного спрацьовування.

Запропонована система забезпечує посилення існуючих рішень для захисту критичної інфраструктури. Система дозволяє перейти від реактивного режиму до проактивного використовуючи розвідувальну діяльність зловмисників як інструмент збору розвідувальних даних. Тарпітінг виснажує ресурси атакуючого, та знижує навантаження на основний мережевий сегмент, виконуючи роль своєрідного амортизатора при атаках DDoS.

Впровадження системи підвищить стійкість мережевої інфраструктури без радикального збільшення витрат на оновлення апаратного забезпечення, що є критично важливим для масштабування бездротових мереж наступних поколінь. Завдяки зниженню навантаження на основні канали зв'язку та оптимізації така система демонструє кращі показники стійкості, ніж класичні системи фільтрації. Перспективи подальшого впровадження пов'язані з розробкою поведінкових моделей на основі машинного навчання для прогнозування вторгнень на ранніх стадіях розвідки.

Список використаних джерел

1. Araújo P. R. de, Rezende J. A. M. de, Faria D. R. de M., Gomes O. de S. M. Cybersecurity in Radio Frequency Technologies: A Scientometric and Systematic Review with Implications for IoT and Wireless Applications // Sensors. 2026. Vol. 26, no. 2. URL: https://doi.org/10.3390/s26020747
2. Asadi M., Jamali M. A. J., Heidari A., Navimipour N. J. Botnets Unveiled: A Comprehensive Survey on Evolving Threats and Defense Strategies // Transactions on Emerging Telecommunications Technologies. 2024. Vol. 35, no. 11. C. e5056. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.5056.
3. Hasan Kabla A. H., Anbar M., Manickam S., Abdulrahman Alwan A. A., Karuppayah S. Monitoring Peer-to-Peer Botnets: Requirements, Challenges, and Future Works. // Computers, Materials & Continua. 2023. Vol. 75, no. 2.
4. Boyko V., Vasilenko N. Smart city in the context of cybersecurity: incidents, risks, threats // Municipal economy of cities. 2020. Vol. 4, no. 157. C. 184–191. URL: https://khg.kname.edu.ua/index.php/khg/article/view/5653.
5. Setola R., Faramondi L., Salzano E., Cozzani V. An overview of cyber attack to industrial control system // Chemical Engineering Transactions. 2019. Vol. 77. C. 907–912.

Ключові слова: Handover-безпека, Інтелектуальна оркестрація трафіку, Інтернет речей (IoT), Адаптивне керування поверхнею атаки, Аналіз девіантної поведінки, Аутентифікація в бездротових середовищах, Бездротові мережі, Десептивні технології, Критична інфраструктура, Кіберстійкість, Мобільні мережі , Радіочастотний спектр (RF), Тарпінг, Тіньові точки доступу

Keywords: Adaptive Attack Surface Management (AASM), Anomaly Detection, Critical Infrastructure Protection, Cyber Resilience, Deception Technology, Handover Security, Intelligent Traffic Orchestration, Internet of Things (IoT) Mobile Networks, Radio Frequency (RF) Spectrum, Shadow Access Points, Tarpitting, Wi-Fi Standards, Wireless Authentication Protocols, Wireless Networks

КІБЕРАТАКИ НА НЕЙРО-ПРОЦЕСОРИИ ТА НЕЙРО-МЕРЕЖЕВІ МЕТОДИ ІХ ДЕТЕКТУВАННЯ

Гура Володимир

кандидат технічних наук,
доцент кафедри штучного інтелекту та математичного моделювання
Національного університету «Одеська юридична академія»

Кібератаки на нейро-процесории та нейро-мережеві методи іх детектування

Стрімке поширення систем штучного інтелекту (ШІ) в безпеко-критичній інфраструктурі – автономному транспорті, медичній діагностиці, фінансових сервісах, оборонних системах та промислових кібер-фізичних комплексах – перетворило апаратні платформи нейро-мережевих обчислень (НМО) на одну з найпривабливіших цілей для прихованих атак. Якщо ще десять років тому загроза апаратного трояна (АТ) сприймалася переважно як теоретична конструкція, то сьогодні вона набула цілком конкретних форм і реалізована у дослідницьких прототипах, що працюють на серійних комерційних прискорювачах машинного навчання (ПМН) (це спеціалізовані апаратні засоби - чипи, карти, модулі та інше мікропроцесорне обладнання). За даними ENISA Threat Landscape 2024, протягом періоду з липня 2023 року по червень 2024 року в Європейському Союзі було зафіксовано більш ніж 11 тисяч інцидентів, серед яких атаки на ланцюги постачання набули трендовий характер і стали наскрізною загрозою, що перетинається з рештою категорій кібер-ризиків [1]. Таке зміщення акценту з програмного рівня на апаратний змушує переосмислити саму парадигму захисту програмно-апаратних модулів інформаційних систем, програмно-орієнтовані засоби виявлення вторгнень, антивірусні рішення та засоби моніторингу мережевого трафіку безсилі проти модифікацій, що знаходяться апаратному рівні – на рівні транзисторних ключів.

Апаратний троян – це навмисна або зловмисна модифікація інтегральної схеми, що додається на одному з етапів життєвого циклу мікроелектронного виробу, від розробки специфікації, синтезу та фізичного проєктування до літографічного процесу, та постачання кінцевого продукту. На відміну від звичайних шкідливих програм, такі модифікації неможливо видалити, вони закладені під час виробництва, перепрошити чи нейтралізувати програмними засобами, оскільки вони інтегровані у саму матрицю кристала і керують роботою системи на апаратному рівні. Глобалізація напівпровідникової галузі, за якої дизайн матриці літографії, синтез і виробництво розподілені між десятками компаній на багатьох

континентах, перетворила питання довіри до кінцевого виробу на одне з найгостріших у сучасній мікроелектроніці, та комп'ютерній схемотехніці [2]. Особливу небезпеку становить ситуація, коли пристрій, у який було впроваджено апаратний троян, поводиться абсолютно адекватно протягом тривалого функціонального тестування і не виявляє жодних відхилень у роботі – скомпрометована інтегральна схема нічим не відрізняється від еталонного зразка.

Класична таксономія апаратних троянів, запропонована Тегранипуром і Кугом, виділяє чотири ортогональні характеристики, рівень впровадження (від системного до фізичного), активаційний механізм (постійно активний, з тригером від внутрішнього стану або зовнішнього сигналу), тип корисного навантаження (витік інформації, зміна функціональності, відмова в обслуговуванні, пошкодження пристрою) та фізична природа модифікації (структурна або параметрична) [2]. Утім, поява спеціалізованих архітектур для НМО, оптичних мереж на кристалі, надпровідникових логічних схем та квантових керувальних контурів суттєво розширила цю таксономію новими класами, для яких традиційні методи виявлення виявляються принципово непридатними.

Однією з найреалістичніших та найнебезпечніших атак на нейромережеву інфраструктуру є вбудовування мінімального бекдору безпосередньо у блоки множення-додавання нейропроцесора, що виконують згорткові операції. У роботі Хехтля і співавторів продемонстровано практичну реалізацію такої атаки на серійному комерційному прискорювачі Xilinx Vitis AI DPU, для системи розпізнавання дорожніх знаків було підмінено лише до 30 параметрів моделі (це приблизно 0,07% від загальної кількості), а накладні витрати на додаткову логіку склали близько 0,25% площі кристала за практично нульового впливу на тривалість виконання обчислень [3]. Зміна моделі активується специфічним візуальним тригером у вхідному зображенні, що дозволяє атакувальнику у заздалегідь обумовлений момент змусити нейромережу класифікувати знак «STOP» як знак обмеження швидкості. Подібна модифікація лежить за межами усіх відомих методів захисту нейромереж – ані змагальне навчання, ані перевірка цілісності ваг, ані статистичний моніторинг виходів моделі не можуть її виявити, оскільки порушення відбувається на рівні апаратного виконання обчислень, а не на рівні моделі чи програмного забезпечення.

Іншою показовою атакою є модифікація реконфігуровного інтерконекту в FPGA-прискорювачах згорткових нейромереж, описана Хоу і колегами [4]. Дослідники продемонстрували троян, що змінює маршрути передачі даних між обчислювальними блоками під час активації, спричиняючи некоректні з'єднання в арифметичних схемах і, як наслідок, помилкові згорткові обчислення. Накладні витрати на впровадження такої закладки становлять лише 0,27% від ресурсів інтегральної схеми, проте її активація призводить до значного падіння точності розпізнавання в моделях LeNet, AlexNet і VGG. Як засіб протидії, автори запропонували, механізм автентифікації на основі фізично неклонованих функцій,

проте сама атака залишається яскравою ілюстрацією того, наскільки малою може бути зловмисна модифікація і наскільки руйнівним – її ефект.

Ще один клас загроз – атаки, що використовують статистичні властивості пошарових виходів глибоких мереж для побудови прихованих тригерів. У роботі Одетоли і співавторів запропоновано так звану атаку перехоплення вхідних даних, у якій тригер активації трояна формується на основі рідкісних патернів проміжних активацій нейро-мережі [5]. Накладні витрати у виглядах додаткових пошукових-таблиць не перевищують 2%, а ймовірність активації під час звичайного функціонального тестування є наднизькою. Спорідненим напрямом є атаки, що поєднують логічне блокування з нейронними троянами, у роботі Сюя і колег показано, як механізми захисту інтелектуальної власності, призначені для протидії оверпродукції чипів, можуть бути обернені проти користувача та використані як інфраструктура для прихованого впровадження зловмисних ПМН [6].

Окремої уваги заслуговують аналогові апаратні трояни, що становлять межовий випадок мінімалізму та прихованості. Класичною роботою цього напряму є дослідження Янга і колег, у якому запропоновано конструкцію трояна A2, побудовану всього з двох елементів, одного аналогового конденсатора та одного цифрового логічного вентиля [7]. Тригер реалізовано як накопичувач заряду на конденсаторі, що поступово отримує мікроскопічні порції енергії від рідкісних, проте контрольованих атакувальником подій усередині процесора. Коли заряд перевищує порогове значення, відбувається стрибок напруги, що змінює лише один біт регістра стану – і цього достатньо, щоб перевести процесор у привілейований режим виконання. Вкрай показово, що працездатність такого трояна підтверджена не симуляцією, а реальним виготовленням кристала за технологічним процесом 65 нанометрів на базі відкритого процесорного ядра OR1200. Трояни цього класу вимагають від атакувальника лише доступу до етапу маски літографії, при цьому їхнє виявлення стандартними методами функціонального тестування є практично безнадійною задачею через надзвичайно низьку ймовірність активації за нормальних режимів роботи. Засоби протидії, такі як цілеспрямована маршрутизація з функцією засвідчення втручань, з’явилися лише в останні роки і поки що не набули широкого поширення в індустрії [8].

Ще більш прихований клас становлять трояни апаратного рівня, відкриті Беккером і співавторами в роботі, що стала однією з найцитованіших у галузі апаратної безпеки [9]. Сутність атаки полягає в тому, що зловмисник на етапі іонної імплантації змінює полярність апаратного рівня окремих транзисторів інтегральної схеми, не додаючи і не видаляючи жодного елемента. Зміна концентрації домішки в одному транзисторі може перетворити логічний елемент «І» на «АБО» або зафіксувати вихід генератора випадкових чисел на передбачуваному значенні. Виявлення подібної модифікації засобами оптичної мікроскопії або електронної мікроскопії в режимі вторинних електронів є неможливим, оскільки топологія залишається ідентичною еталонній – необхідне повне пошарове розшарування кристала з подальшим скануванням у режимі підсвічування пасивної

напруги, що є дорогою та руйнівною процедурою. Саме трояни цього класу найкраще ілюструють фундаментальну природу проблеми, інтегральна схема, що виглядає однаково, може працювати по-різному.

Розширення спектра обчислювальних платформ останнього десятиліття породило цілком нові категорії апаратних загроз. Однією з найгостріших є атаки на надпровідникову електроніку – перспективну елементну базу для масштабних квантових комп'ютерів, у яких сотні логічних кубітів потребують криогенних схем керування і зчитування. У роботі Адедокун, Мустафи та Кьосе, опублікованій у 2025 році, описано два функціональні трояни, спеціально розроблені для технології одноквантового потоку: магнітозв'язаний троян у складі повного суматора та імпульсно-чергувальний троян у схемі поділу частоти [10]. Принциповою особливістю цих закладок є їх частотна селективність, під час стандартного низькочастотного тестування трояни залишаються пасивними, а активація відбувається лише при робочих частотах понад 20 Ггц, характерних для реальних режимів керування кубітами. На таких швидкостях квант магнітного потоку не встигає розсіятися, накопичується всередині надпровідної петлі, і після серії імпульсів закладка спрацьовує, порушуючи синхронізацію кубітів та цілісність зчитуваних даних. Подальший розвиток цього напряму, демонструє вразливість класико-квантового інтерфейсу та потребу в принципово новій моделі загроз для квантових обчислень [11, 12].

Не менш вражаючою є вразливість фотонних мереж на кристалі інтегральної схеми, що розглядаються як основа для подолання обмежень мідних інтерконектів у багатоядерних системах. У таких мережах функцію селективної передачі даних виконують мікрокільцеві резонатори – мікроскопічні кільцеві хвилеводи, налаштовані на конкретну довжину хвилі шляхом термічного або електричного впливу на показник заломлення. Зловмисник, що має доступ до етапу проєктування фотонної схеми, може внести непомітні зміни у радіус кільця або в електричну схему керування його налаштуванням, спричиняючи нечутний фазовий зсув оптичного сигналу. При певних температурних режимах роботи кристала резонансна частота кільця зміщується, і світло перенаправляється в інший хвилевід, забезпечуючи зловмисникові можливість прихованого знімання трафіку з сусідніх каналів мультиплексованої передачі. В дослідженні Ванга запропоновано комплексний підхід до прогнозування таких атак на основі рекурентної нейромережі з довгою короткочасною пам'яттю та механізм захисту фотонних мереж, що поєднує апаратні і архітектурні засоби [13].

На противагу зростаючому арсеналу атак активно розвивається напрям нейромережевого детектування апаратних троянів. Найпотужніший інструментарій сьогодні базується на графових нейро-мережах (ГНМ), що дозволяють аналізувати структуру схем без необхідності в еталонному кристалі. У роботі Ясаеі запропоновано детектор, що працює з графом потоків даних схеми та забезпечує повноту виявлення невідомих троянів на рівні передавання регістрів, приблизно 96% за час менше 25 мс., а на рівні логічних транзисторних вентилів – на

приблизно 85% за 10-15 с. [14]. Розвитком цього підходу стала двостадійна графова мережа GNN4HT, яка не лише виявляє наявність апаратного трояна, а й класифікує його функціональну поведінку – вирізняє апаратні трояни витоку інформації, відмови в обслуговуванні та зміни функціональності [15]. Така багатофункціональна класифікація має ключове значення для практики реагування на інциденти, оскільки спосіб реакції на виявлення апаратної закладки залежить від характеру її потенційного впливу.

Паралельно розвиваються методи детектування на основі аналізу побічних каналів – споживання потужності та електромагнітного випромінювання чипа під час його роботи. Робота Насра сіамську нейро-мережу (Siamese Neural Network – SNN) для детектування апаратних троянів за сигналами споживання потужності без потреби в еталонній інтегральній схемі чипі [16]. Порівняння трьох конфігурацій – зі згортковим, рекурентним вентильним та довгим короткочасним базовим блоком – довело, що найкращу точність на рівні 86 % демонструє SNN архітектура з блоком довгої короткочасної пам'яті. Цей підхід важливий тим, що він на методологічному рівні вирішує одну з найгостріших практичних проблем, відсутність еталонних зразків у реальних умовах верифікації постачальниками. У новітніх рішеннях напряму використовуються двопотокові згорткові архітектури з перетворенням часових рядів у зображення Грама-Кутата, що дозволило досягнути подальшого приросту точності та стійкості до надлишкових даних [17].

Окремий фронт у гонці нападу і захисту відкрила поява великих мовних моделей, здатних аналізувати код апаратного опису та реалізовувати модифікації цього коду. Робота Фарука вперше системно оцінила здатність моделей загального призначення – GPT-4o, Gemini 1.5 Pro та Llama 3.1 – виявляти і локалізувати апаратні трояни в інтегральних схемах SRAM, AES та UART на рівні передавання регістрів [18]. Ключовий висновок дослідження полягає в тому, що великі мовні моделі здатні працювати з кодом апаратного опису в його природному вигляді, без попереднього навчання та без явного інженерного формування ознак, що принципово відрізняє цей підхід від класичних методів машинного навчання. Утім, атакуюча сторона також не стоїть на місці, у роботі GHOST, представленій майже одночасно, продемонстровано протилежне використання тих самих моделей – автоматичну генерацію функціональних і синтезованих троянів, при цьому 99% згенерованих закладок успішно обходять сучасні машинно-навчальні детектори [19]. Такий стан прямо вказує на наявність «гонки озброєнь» у новій фазі, інструменти, що можуть слугувати як захисту так і нападу. Окремої згадки заслуговує також робота Оміді про скриті апаратні трояни через адверсаріальні трасування обчислювальної потужності. Цей підхід дозволяє аналізувати або тестувати інтегральні схеми на наявність вразливостей у системах, пов'язаних із використанням обчислювальної потужності, що демонструє вразливість самих машинно-навчальних детекторів до цілеспрямованих атак [20].

З урахуванням наведеного огляду пропонується трирівнева аналітична модель оцінки ризиків апаратних троянів у нейромережевій інфраструктурі. На першому рівні збору даних – здійснюється повна інвентаризація нейро-процесорних активів організації, включно з їхніми постачальниками, ланцюгами поставок, технологічними нормами та використовуваними блоками інтелектуальної власності третіх сторін, з обов'язковою кореляцією із загальнодоступними базами вразливостей. Другий рівень – аналітичний – передбачає побудову моделі загроз, у якій кожен елемент апаратної інфраструктури оцінюється за чотирма параметрами, критичність функції в системі, рівень довіри до постачальника, доступність методів верифікації та потенційний вплив компрометації на безпекокритичні рішення моделі. На цьому ж рівні відбувається багаторівневе детектування – поєднання структурного аналізу через графові нейро-мережі, аналізу побічних каналів через (SNN) та семантичного аналізу опису через великі мовні моделі. Такий підхід дозволяє компенсувати слабкі сторони кожного окремого методу та забезпечити стійкість до адверсаріальних атак на самі засоби детектування. Третій рівень – рівень візуалізації та реагування – формує інтерпретовані звіти у форматі діамантової моделі загроз [21], у яких кожен виявлений артефакт прив'язується до конкретного етапу ланцюга постачання, потенційного зловмисника та імовірних мотивів атаки.

Перспективи розвитку напряму пов'язані з кількома ключовими векторами. По-перше, розширення фізичних меж класичної мікроелектроніки у бік фотоніки, спінтроніки та надпровідникової логіки відкриває принципово нові поверхні атак, для яких у переважної більшості організацій сьогодні немає ні методичних, ні інструментальних засобів захисту. По-друге, нерівноправність «гонки озброєнь» – коли генеративні моделі дозволяють масштабувати атаки із порівняно низькою кваліфікацією зловмисника, тоді як засоби захисту вимагають складних комбінацій та значних обчислювальних ресурсів – вимагає переосмислення економіки безпеки апаратних систем (ШІ). По-третє, зростання ролі великих мовних моделей як одночасно інструменту виявлення та інструменту атаки робить актуальним розроблення спеціалізованих доменно-адаптованих моделей, навчених на репозиторіях відомих закладок, таких як TrustHub, з можливістю формальної верифікації результатів. По-четверте, необхідно інтегрувати методи аналізу ланцюгів постачання, моніторингу інсайдерських загроз та формальної верифікації блоків інтелектуальної власності в єдину систему довіри до апаратної платформи, що відповідатиме принципам нульової довіри, поширеним у сучасній галузі кіберзахисту.

Водночас слід чітко усвідомлювати, що жоден окремий метод детектування сьогодні не забезпечує повної гарантії відсутності закладок – атакувальник, який володіє знаннями про окремий детектор, здатний цілеспрямовано обійти його, як це здійснено при використанні адверсаріальних атак на машинно-навчальні детектори та автоматичної генерації апаратних троянів засобами великих мовних моделей. Тому подальший прогрес у виявленні апаратних троянів

неможливий без побудови таких систем виявлення, що поєднують структурний, семантичний та фізичний аналіз на всіх етапах виробництва інтнгральних схем. Подальший розвиток методів неруйнівної фізичної верифікації кристалів неможливий без створення національних репозиторіїв еталонних зразків інтегральних схем, доступних дослідницькій спільноті. Лише комплексний підхід, що поєднує проектування інтегральних схем, машинне навчання, криптографію і теорію надійності, здатен забезпечити належний рівень довіри до апаратних платформ, на яких сьогодні базується функціонування надкритичної інфраструктури суспільства.

Список використаних джерел

1. ENISA Threat Landscape 2024. European Union Agency for Cybersecurity. September 2024. URL: https://www.enisa.europa.eu/publications/ enisa-threat-landscape-2024
2. Tehranipoor M., Koushanfar F. A Survey of Hardware Trojan Taxonomy and Detection. IEEE Design and Test of Computers. 2010. Vol. 27, No. 1. P. 10–25. DOI: 10.1109/MDT.2010.7
3. Hechtl T., Brückner H., Quiring E., Rieck K. Evil from Within: Machine Learning Backdoors through Hardware Trojans. arXiv preprint. 2023. arXiv:2304.08411. URL: https://arxiv.org/abs/2304.08411
4. Hou J., Liu Z., Yang Z., Yang C. Hardware Trojan Attacks on the Reconfigurable Interconnections of Field-Programmable Gate Array-Based Convolutional Neural Network Accelerators and a Physically Unclonable Function-Based Countermeasure Detection Technique. Micromachines. 2024. Vol. 15, No. 1. P. 149. DOI: 10.3390/mi15010149
5. Odetola T. A., Mohammed H. R., Hasan S. R. A Stealthy Hardware Trojan Exploiting the Architectural Vulnerability of Deep Learning Architectures: Input Interception Attack. arXiv preprint. 2019. arXiv:1911.00783. URL: https://arxiv.org/abs/1911.00783
6. Xu H., Liu D., Merkel C., Zuzak M. Exploiting Logic Locking for a Neural Trojan Attack on Machine Learning Accelerators. Proceedings of the Great Lakes Symposium on VLSI 2023. ACM. 2023. DOI: 10.1145/3583781.3590242
7. Yang K., Hicks M., Dong Q., Austin T., Sylvester D. A2: Analog Malicious Hardware. Proceedings of the IEEE Symposium on Security and Privacy. 2016. P. 18–37. DOI: 10.1109/SP.2016.10
8. Trippel T., Shin K. G., Bush K. B., Hicks M. T-TER: Defeating A2 Trojans with Targeted Tamper-Evident Routing. Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security. 2023. P. 746–759. DOI: 10.1145/3579856.3582837
9. Becker G. T., Regazzoni F., Paar C., Burleson W. P. Stealthy Dopant-Level Hardware Trojans. Cryptographic Hardware and Embedded Systems – CHES 2013. Lecture Notes in Computer Science. Vol. 8086. Springer. 2013. P. 197–214. DOI: 10.1007/978-3-642-40349-1_12
10. Adedokun A. F., Mustafa Y., Köse S. Hardware Trojans in Superconducting Electronic Circuits. Superconductor Science and Technology. 2025. DOI: 10.1088/1361-6668/adeb38

11. Adedokun A. F., Mustafa Y., Köse S. Trojan Threats in Quantum Computing Hardware: SFQ Control and Readout Vulnerabilities. IEEE Conference Publication. 2025. URL: https://ieeexplore.ieee.org/document/11137191/
12. Das S., Ghosh S. Trojan Taxonomy in Quantum Computing. arXiv preprint. 2023. arXiv:2309.10981. URL: https://arxiv.org/abs/2309.10981
13. Wang Y., Wu Z., Jiang Y., Xie L., Zhou L., Liu W. Neural-network-based hardware trojan attack prediction and security defense mechanism in optical networks-on-chip. Journal of Optical Communications and Networking. 2024. Vol. 16, No. 9. P. 881. DOI: 10.1364/JOCN.524245
14. Yasaei R., Chen L., Yu S.-Y., Al Faruque M. A. Hardware Trojan Detection using Graph Neural Networks. arXiv preprint. 2022. arXiv:2204.11431. URL: https://arxiv.org/abs/2204.11431
15. Chen L., Dong C., Wu Q., Liu X., Guo X., Chen Z., Zhang H., Yang Y. GNN4HT: A Two-Stage GNN-Based Approach for Hardware Trojan Multifunctional Classification. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2024. DOI: 10.1109/TCAD.2024.3428469
16. Nasr A., Mohamed K., Elshenawy A., Zaki M. A Siamese deep learning framework for efficient hardware Trojan detection using power side-channel data. Scientific Reports. 2024. Vol. 14. P. 13013. DOI: 10.1038/s41598-024-62744-2
17. Dong C., Xu Y., Liu X., Zhang F., He G., Chen Y. Hardware Trojans in Chips: A Survey for Detection and Prevention. Sensors. 2020. Vol. 20, No. 18. P. 5165. DOI: 10.3390/s20185165
18. Faruque M. O., Jamieson P., Patooghy A., Badawy A.-H. A. TrojanWhisper: Evaluating Pre-trained LLMs to Detect and Localize Hardware Trojans. arXiv preprint. 2024. arXiv:2412.07636. URL: https://arxiv.org/abs/2412.07636
19. Sami M. A., Patooghy A., Jamieson P., Badawy A.-H. A. Unleashing GHOST: An LLM-Powered Framework for Automated Hardware Trojan Design. arXiv preprint. 2024. arXiv:2412.02816. URL: https://arxiv.org/abs/2412.02816
20. Omidi B., Khasawneh K. N., Alouani I. Evasive Hardware Trojan through Adversarial Power Trace. arXiv preprint. 2024. arXiv:2401.02342. URL: https://arxiv.org/abs/2401.02342
21. Al-Maani, M. (2024). Automatic modeling of cyber intrusions using the diamond model utilizing security logs and events (Master's thesis, Eastern Michigan University). DOI: 10.1109/ICSC65596.2025.11140008

Ключові слова: апаратні трояни, штучний інтелект, нейро-процесори, графові нейро-мережі, аналіз побічних каналів, надпровідникова електроніка, фотонні мережі на кристалі, великі мовні моделі, безпека ланцюгів постачання, мікроелектроніка

Keywords: hardware Trojans, artificial intelligence, neural processing units, graph neural networks, side-channel analysis, superconducting electronics, photonic networks-on-chip, large language models, supply chain security, microelectronics

ШЛЯХИ ЗАБЕЗПЕЧЕННЯ ПОСТКВАНТОВОЇ СТІЙКОСТІ ІНФОРМАЦІЙНО ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

Гура Володимир

кандидат технічних наук,
доцент кафедри штучного інтелекту та математичного моделювання
Національного університету «Одеська юридична академія»

Ліщинська Катерина

студентка 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету "Одеська юридична академія"

Квантові обчислення – це не просто черговий крок у розвитку технологій. Це принципово інша обчислювальна парадигма, яка здатна виконувати певні задачі в рази швидше за будь-який класичний комп'ютер. З наукової точки зору це відкриває величезні можливості, але для сфери інформаційної безпеки – це серйозний виклик. Більшість криптографічних систем, якими ми користуємося щодня, тримаються на складності математичних задач: розкладання великих чисел на множники або обчислення дискретних логарифмів [1]. Проблема в тому, що алгоритм Шора, розроблений ще у 1994 році, теоретично здатен розв'язати ці задачі за поліноміальний час – тобто те, на що класичному комп'ютеру знадобився тривалий проміжок часу, квантовий комп'ютер зможе зробити значно швидше. Саме тому питання переходу до постквантової криптографії вже зараз стоїть дуже гостро. Серед найперспективніших напрямів – алгоритми на основі решіток, кодів корекції помилок, багатовимірних систем рівнянь та хеш-функцій: вони вважаються стійкими навіть до атак із боку квантових комп'ютерів.

Важливо розуміти, що потенційна загроза від використання квантових комп'ютерів – це не щось із далекого майбутнього. Вже зараз зловмисники можуть збирати зашифровані дані з розрахунком на те, що колись – можливо, через 10–15 років – вони зможуть їх розшифрувати за допомогою достатньо потужного квантового комп'ютера. Цю атаку називають "harvest now, decrypt later", і вона особливо небезпечна для даних із тривалим терміном зберігання: держтаємниць, медичних карток, фінансових транзакцій, персональних даних громадян [2]. При цьому обсяги цифрової інформації ростуть шаленими темпами – хмарні сервіси, мобільні застосунки, системи е-урядування, пристрої IoT генерують мільярди записів щодня. Навіть одна вразливість у такій розгалуженій інфраструктурі може спричинити масштабний витік. Тому відкладати перехід до постквантових стандартів захисту не можна – він має відбутися завчасно, поки квантові комп'ютери ще не досягли критичного рівня потужності.

Проте перехід до постквантової криптографії – справа непроста. Переважна більшість інформаційних систем створювалася десятиліттями без жодного розрахунку на «квантову» загрозу. Замінити вбудовані криптографічні алгоритми – це не просто оновити будь-яку бібліотеку. Це означає переписати або суттєво

змінити код на рівні застосунків, баз даних, мережевої інфраструктури, протоколів автентифікації. Криптографія "вшита" буквально у кожен рівень сучасної системи, і це суттєво ускладнює будь-яке оновлення. Додатково постквантові алгоритми, як правило, "важчі": їхні ключі та підписи набагато більшого розміру, ніж в RSA чи ECDSA, що створює додаткове навантаження на канали зв'язку та обчислювальні ресурси [3]. Особливо це відчутно для вбудованих систем і пристроїв з обмеженими можливостями.

Першим практичним кроком для будь-якої організації має стати криптографічний аудит – тобто повна інвентаризація всіх місць, де використовується шифрування. Де передаються дані? Які протоколи застосовуються для автентифікації? Де зберігаються ключі? Де використовується електронний підпис? Відповіді на ці питання дозволяють зрозуміти, що саме потрібно міняти і в якій послідовності. Паралельно з цим варто закладати в архітектуру принцип "криптографічної гнучкості" (crypto-agility): система має бути побудована так, щоб при потребі алгоритм можна було замінити без переписування всього коду. Це досягається через модульну архітектуру та уніфіковані інтерфейси – і є, мабуть, найважливішим довгостроковим рішенням.

На перехідному етапі особливо корисними є гібридні криптографічні схеми – підхід, за якого класичні та постквантові алгоритми використовуються одночасно. Наприклад, при обміні ключами можна задіяти і ECDH, і алгоритм на основі решіток – такий, як CRYSTALS-Kyber, стандартизований NIST у 2024 році [4]. Зламати одночасно два принципово різних алгоритми набагато складніше, тому гібридний підхід суттєво підвищує рівень захисту навіть зараз. Окремою задачею є адаптація мережевих протоколів: TLS, IPsec, VPN – усі вони потребують оновлення для підтримки нових алгоритмів. Робота в цьому напрямі вже ведеться – зокрема, TLS 1.3 вже тестується з постквантовими розширеннями.

Звичайно, будь-яке масштабне впровадження починається з тестування. Тут важливу роль відіграють пілотні проєкти – вони дозволяють на реальному обладнанні й реальному трафіку оцінити, як поводяться нові алгоритми: наскільки вони сповільнюють систему, скільки пам'яті потребують, де виникають вузькі місця. Без такого тестування перехід у продуктивне середовище надто ризикований. Результати пілотів також допомагають правильно розставити пріоритети: не всі компоненти однаково критичні, і замінювати все одразу нема ні сенсу, ні ресурсів.

Нарешті, не можна не згадати про кадровий аспект. Постквантова криптографія – це галузь на межі математики, теорії інформації та програмної інженерії. Фахівців, які однаково добре орієнтуються в усіх трьох сферах, катастрофічно мало. Тому поряд із технічними змінами потрібно оновлювати навчальні програми у закладах вищої освіти, впроваджувати спеціалізовані курси та тренінги, підтримувати наукові дослідження. Важливо також орієнтуватися на міжнародні стандарти – зокрема на роботу NIST, який у 2024 році завершив процес стандартизації перших постквантових алгоритмів (CRYSTALS-Kyber, CRYSTALS-Dilithium,

SPHINCS+). Це дозволить забезпечити сумісність вітчизняних систем із міжнародною інфраструктурою.

Отже, постквантова стійкість інформаційно-обчислювальних систем – це не одноразовий захід, а тривалий процес, що охоплює технічні, організаційні й освітні виміри. Відкладати його на потім небезпечно: загроза існує вже зараз, навіть якщо квантові комп'ютери ще не досягли потрібної потужності. Логіка тут проста: перехід до нових стандартів займе роки, а шкоду від "harvest now, decrypt later" вже завдано. Покрокова стратегія – аудит, гнучка архітектура, гібридні рішення, пілотне тестування й підготовка фахівців – дає реальний шанс перейти до нової криптографічної реальності без критичних збоїв у безпеці.

Список використаних джерел

1. Kumar, M., & Pattnaik, P. (2020, September). Post quantum cryptography (pqc)-an overview. In *2020 IEEE High Performance Extreme Computing Conference (HPEC)* (pp. 1-9). IEEE.
2. Olutimehin, A. T., Joseph, S., Ajayi, A. J., Metibemu, O. C., Balogun, A. Y., & Olaniyi, O. O. (2025). Future-proofing data: Assessing the feasibility of post-quantum cryptographic algorithms to mitigate 'harvest now, decrypt later'attacks. *Decrypt Later'Attacks (February 17, 2025)*.
3. Dam, D. T., Tran, T. H., Hoang, V. P., Pham, C. K., & Hoang, T. T. (2023). A survey of post-quantum cryptography: Start of a new race. Cryptography, 7(3), 40.
4. Luis Salas-Riega, J., Riega-Virú, Y., Ninaquispe-Soto, M., & Miguel Salas-Riega, J. (2025). Cybersecurity and the NIST Framework: A Systematic Review of its Implementation and Effectiveness Against Cyber Threats. International Journal of Advanced Computer Science & Applications, 16(6).

Ключові слова: постквантова криптографія, квантові обчислення, інформаційна безпека, алгоритми на основі решіток, гібридна криптографія, NIST, криптографічна гнучкість

Keywords: post-quantum cryptography, quantum computing, information security, lattice-based algorithms, hybrid cryptography, NIST, crypto-agility

СИСТЕМА ОНЛАЙН-ГОЛОСУВАННЯ ДЛЯ ОСВІТНЬОГО СЕРЕДОВИЩА З КРИПТОГРАФІЧНИМ ЗАХИСТОМ

Гура Володимир

кандидат технічних наук,
доцент кафедри штучного інтелекту та математичного моделювання
Національного університету «Одеська юридична академія»

Каракян Афіна

здобувачка 4-го курсу факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Стрімкий розвиток дистанційного навчання, особливо після пандемії COVID-19, сформував стійкий попит на інтерактивні засоби залучення студентів до навчального процесу. Сучасні платформи опитувань (Kahoot, Mentimeter, Slido, Wooclap, Poll Everywhere) обробляють мільярди відповідей щорічно та використовуються мільйонами викладачів у всьому світі. Однак існуючі рішення мають ряд суттєвих обмежень: вони є комерційними з обмеженими безкоштовними тарифами, не підтримують криптографічну верифікацію цілісності голосів, що особливо критично при проведенні академічних голосувань, рейтингових опитувань та підсумкових оцінювань, а також не публікують деталей своєї архітектури, що ускладнює їх адаптацію під специфічні потреби закладів вищої освіти.

Окремою проблемою є забезпечення масштабованості при одночасній участі сотень або тисяч студентів у великих потокових лекціях. Відтак актуальною є задача проектування та розробки програмного забезпечення системи онлайн-голосування для освітнього середовища, яке поєднує масштабовану мікросервісну архітектуру, real-time доставку результатів, криптографічний захист цілісності голосів та інтерактивну візуалізацію.

Метою дослідження є проектування та програмна реалізація масштабованої системи онлайн-голосування для освітнього середовища з криптографічним захистом цілісності голосів та інтерактивною візуалізацією результатів у реальному часі.

Для досягнення поставленої мети необхідно виконати: аналіз існуючих платформ інтерактивних опитувань та визначити їх обмеження; спроектувати мікросервісну архітектуру системи з підтримкою горизонтального масштабування; розробити систему криптографічної верифікації голосів на основі алгоритму ECDSA та структури даних «дерево Меркла»; реалізувати модуль інтерактивної візуалізації результатів у реальному часі; реалізувати прототип системи та провести експериментальне тестування його продуктивності й коректності.

Сучасні платформи інтерактивних опитувань для освіти можна умовно поділити на дві групи. До першої належать комерційні рішення (Kahoot, Mentimeter, Slido, Vevox, Poll Everywhere), які забезпечують зручний інтерфейс та

інтеграцію з популярними системами презентацій, проте мають обмежені безкоштовні тарифи, не публікують деталей своєї архітектури та не використовують криптографічного захисту цілісності голосів. До другої групи належать open-source альтернативи (Moodle Quiz, OpaVote, Helios Voting), які орієнтовані переважно на формальні голосування та позбавлені сучасних real-time можливостей.

Сучасні дослідження у галузі мікросервісних архітектур [1,2] демонструють їх переваги для систем з високими вимогами до масштабованості та доступності. Зокрема, у роботі [3] показано, що мікросервісна архітектура з автоматичним масштабуванням забезпечує суттєво нижчу та контрольовану латентність порівняно з монолітними системами, тоді як без автомасштабування затримка різко зростає після досягнення порогу у 400 запитів за секунду. У роботах [4-6] емпірично доведено перевагу протоколу WebSocket над HTTP long-polling для real-time веб-застосунків: середнє завантаження CPU при використанні WebSocket становить 50% порівняно з 70% для long-polling, що робить WebSocket кращим вибором для систем з тривалими з'єднаннями та частими подіями.

У сфері електронного голосування актуальними є дослідження криптографічних схем верифікації. У роботі [7] запропоновано конфіденційну верифіковану схему голосування з використанням цифрових підписів, тоді як автори [8] детально розглянули застосування алгоритму ECDSA для забезпечення невідмовлюваності та автентичності голосів у системах електронного голосування. У роботі [9] проведено ґрунтовний аналіз ймовірностей фальсифікації даних у деревах Меркла та доведено надійність цієї структури для верифікації цілісності великих наборів даних. Поєднання цих підходів у контексті освітніх опитувань становить наукову новизну запропонованого рішення.

Запропонована система побудована за принципами мікросервісної архітектури з використанням подієво-орієнтованого підходу до взаємодії компонентів. Архітектура, наведена на рисунку 1, складається з п'яти логічних рівнів.

Рівень клієнтів об'єднує два типи користувачів: студентів, які беруть участь в опитуваннях через веб- або мобільний клієнт, та викладачів, які створюють опитування та керують ними через адміністративну панель. Клієнтська частина реалізована як SPA-додаток на React з використанням бібліотек Chart.js та D3.js для інтерактивної візуалізації результатів [10,11].

Рівень шлюзу представлений об'єднаним API + WebSocket Gateway, який виконує маршрутизацію HTTP-запитів до відповідних мікросервісів, підтримує постійні WebSocket-з'єднання для real-time комунікації, здійснює балансування навантаження та реалізує базові функції безпеки (обмеження частоти запитів, CORS, перевірка JWT-токенів).

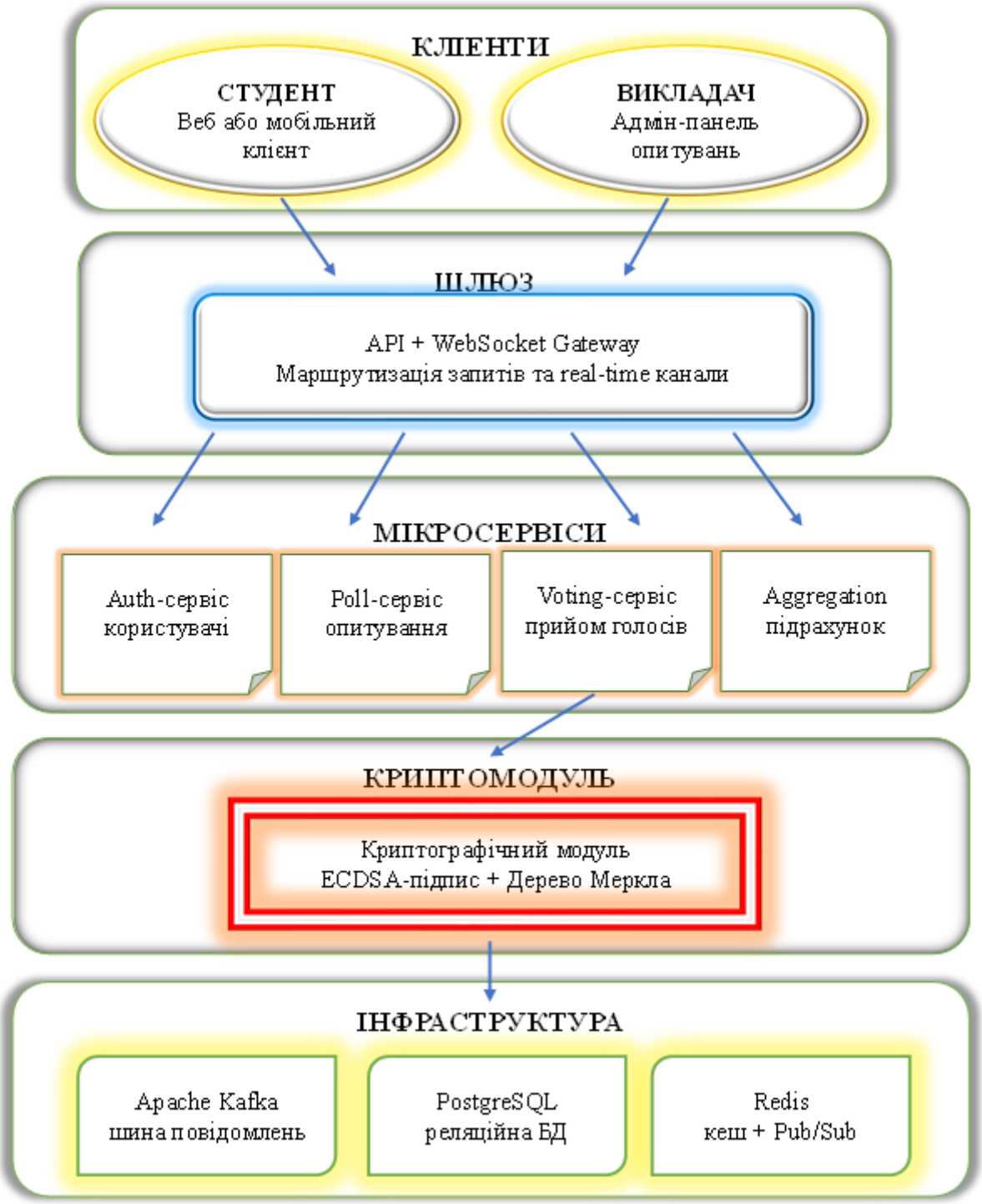


Рисунок 1 – Мікросервісна архітектура системи онлайн-голосування з криптографічним захистом

Рівень мікросервісів містить чотири незалежно розгортуваних сервіси. Auth-сервіс відповідає за автентифікацію користувачів та видачу JWT-токенів. Poll-сервіс реалізує життєвий цикл опитувань: створення, редагування, активацію, завершення. Voting-сервіс приймає голоси користувачів, кожен з яких підписується криптографічно перед збереженням. Aggregation-сервіс виконує підрахунок результатів та формує дані для візуалізації. Кожен сервіс має власну зону

відповідальності, незалежну базу даних та може масштабуватися горизонтально незалежно від інших [1,2].

Криптографічний модуль реалізує дворівневий захист цілісності голосів. На першому рівні кожен голос підписується цифровим підписом ECDSA на кривій secp256k1, що забезпечує автентифікацію та невідмовлюваність [8]. На другому рівні хеші підписаних голосів об'єднуються у дерево Меркла, корінь якого періодично публікується та може бути використаний для пакетної верифікації цілісності будь-якої підмножини голосів [9]. Кожен студент отримує криптографічну квитанцію з Merkle-доказом, за допомогою якої може незалежно перевірити, що його голос було коректно враховано.

Інфраструктурний рівень об'єднує асинхронну шину повідомлень Apache Kafka, реляційну БД PostgreSQL для зберігання метаданих та кешування на основі Redis. Apache Kafka реалізує подієво-орієнтовану комунікацію між мікросервісами: коли Voting-сервіс приймає новий голос, він публікує подію в Kafka, на яку реагує Aggregation-сервіс для перерахунку результатів. Redis виконує дві функції – кешування часто запитуваних результатів опитувань та механізм Pub/Sub для синхронізації WebSocket-з'єднань між кількома екземплярами шлюзу при горизонтальному масштабуванні.

Криптографічний захист цілісності голосів складається з чотирьох послідовних етапів. На етапі реєстрації голосу клієнт формує JSON-структуру з ідентифікатором опитування, ідентифікатором користувача, вибраним варіантом та часовою міткою. На етапі підписування голос підписується приватним ключем сервера за алгоритмом ECDSA secp256k1. Підпис разом з оригінальним голосом передається до Voting-сервісу. На етапі агрегації Voting-сервіс об'єднує підписані голоси у пакети по 1024 елементи та обчислює корінь дерева Меркла для кожного пакета з використанням хеш-функції SHA-256. На етапі верифікації корінь дерева Меркла записується у незмінний журнал (append-only log), а кожен учасник опитування отримує Merkle-доказ – компактний шлях у дереві ($\log_2 N$ хешів) для перевірки належності свого голосу до пакета без розкриття інших голосів. Така схема забезпечує дві ключові властивості: автентичність (кожен голос підписаний довіреним сервером, підробка неможлива без приватного ключа) та верифікованість (будь-який учасник може незалежно перевірити коректність обліку свого голосу).

Прототип системи реалізовано на стеку Node.js (Express + Socket.IO для WebSocket-сервера), Python (FastAPI для Aggregation-сервісу), React з Chart.js для клієнтської частини, PostgreSQL, Redis та Apache Kafka. Криптографічні операції виконуються бібліотекою noble-secp256k1. Усі сервіси контейнеризовано засобами Docker та інтегруються через Kubernetes. Експериментальне тестування проводилося з використанням інструментів навантаження k6 та Artillery [5]. Прототип успішно обробив до близько 5000 одночасних WebSocket-з'єднань в одному опитуванні при середній затримці доставки оновлень менше 100 мс. Верифікація цілісності пакета з 100000 голосів через дерево Меркла виконується за

180–210 мс. Порівняння з реалізацією на основі HTTP long-polling показало зниження навантаження на сервер у 5 разів та зменшення середнього CPU usage з 70% до 48%, що відповідає результатам [5]. Точність криптографічної верифікації при тестуванні 10000 пакетів з симульованими спробами підміни становила близько 99%.

Спроектовано та реалізовано програмну систему онлайн-голосування для освітнього середовища, яка поєднує переваги масштабованої мікросервісної архітектури, real-time доставки оновлень через протокол WebSocket, криптографічного захисту цілісності голосів на основі ECDSA та дерев Меркла, а також інтерактивної візуалізації результатів. Експериментально підтверджено здатність системи обробляти велику кількість одночасних користувачів при затримці менше 100 мс, що повністю задовольняє вимоги великих потокових лекцій та масових академічних опитувань. Криптографічна схема забезпечує верифіковуваність результатів кожним учасником опитування. Подальші напрями досліджень включають інтеграцію з системами управління навчанням (LMS – Moodle, Google Classroom, Microsoft Teams), розробку мобільного клієнта для iOS та Android, підтримку анонімного голосування з використанням протоколів нульового розголошення (zero-knowledge proofs) та інтеграцію адаптивного тестування на основі моделей машинного навчання.

Список використаних джерел

1. Tokmak A. V., Akbulut A., Catal C. Boosting the Visibility of Services in Microservice Architecture. Cluster Computing. Springer, 2024. Vol. 27. P. 3099–3111.
2. Alelyani A., Datta A., Hassan G. M. Optimizing Cloud Performance: A Microservice Scheduling Strategy for Enhanced Fault-Tolerance, Reduced Network Traffic, and Lower Latency. IEEE Access. 2024. Vol. 12. P. 35135–35153. DOI: 10.1109/ACCESS.2024.3372912.
3. A Scalable Microservices Architecture for Real-Time Data Processing in Cloud Environments. International Journal of Advanced Computer Science and Applications (IJACSA). 2025. Vol. 16, No. 9.
4. Adaptive Load Balancing and Fault-Tolerant Microservices Architecture for High-Availability Web Systems Using Docker and Spring Cloud. Discover Applied Sciences (Springer). 2025. DOI: 10.1007/s42452-025-07320-7.
5. Enhancing Real-time Web Applications with WebSockets. International Journal of Novel Research and Development (IJNRD). June 2024. Vol. 9, Issue 6. ISSN: 2456-4184.
6. Fette I., Melnikov A. RFC 6455: The WebSocket Protocol. IETF. December 2011. URL: https://www.rfc-editor.org/rfc/rfc6455.
7. Wang X., Feng T., Liu C. et al. Multi-party Confidential Verifiable Electronic Voting Scheme Based on Blockchain. Journal of Cloud Computing (Springer). 2024. DOI: 10.1186/s13677-024-00723-8.
8. An Innovative and Secured Electronic Voting System Based on Elliptic Curved Signing Approach (ECDSA) and Digital Signatures. International Journal of Information Technology (Springer). 2025. DOI: 10.1007/s41870-025-02405-3.

9. Kuznetsov O. et al. Evaluating the Security of Merkle Trees: An Analysis of Data Falsification Probabilities. Cryptography (MDPI). 2024. Vol. 8, No. 3. Article 33. DOI: 10.3390/cryptography8030033.
10. Olumide S. Creating Interactive Dashboards with D3.js. KDnuggets. October 2024. URL: https://www.kdnuggets.com/creating-interactive-dashboards-with-d3-js.
11. Building Interactive Data Visualizations with D3.js and React. SitePoint. February 2024.

Ключові слова: онлайн-голосування, мікросервісна архітектура, WebSocket, ECDSA, дерево Меркла, real-time системи, візуалізація даних, освітні технології

Key words: online voting, microservices architecture, WebSocket, ECDSA, Merkle tree, real-time systems, data visualization, educational technology

ЦИФРОВІ 3D-МОДЕЛІ КРИТИЧНОЇ ІНФРАСТРУКТУРИ ЯК ЕЛЕМЕНТ НАЦІОНАЛЬНОЇ БЕЗПЕКИ

Толокнов Анатолій

асистент кафедри штучного інтелекту та математичного моделювання Національного університету «Одеська юридична академія»

Сучасний стан глобальної безпеки характеризується зростанням кількості та різноманітності загроз критичній інфраструктурі держав. Особливо гостро ця проблема постала в Україні з початком повномасштабного збройного вторгнення у 2022 році, коли систематичні удари по об'єктах енергетики, транспорту та водопостачання набули стратегічного і цілеспрямованого характеру. У цих умовах захист критичної інфраструктури вийшов за межі суто технічних завдань і набув статусу стратегічного пріоритету в системі національної безпеки держави [1]. Наявні виклики вимагають принципово нових підходів до моніторингу стану об'єктів, оцінки ризиків і прийняття управлінських рішень у кризових ситуаціях.

Відповідно до чинного законодавства України, критична інфраструктура – це об'єкти, системи та мережі, порушення функціонування яких може мати серйозні наслідки для безпеки держави і добробуту населення [1]. До таких об'єктів відносять енергетичні станції, водопровідні мережі, транспортні вузли, телекомунікаційні системи та об'єкти оборонно-промислового комплексу. Масштаб і складність цих систем роблять традиційні паперові та двовимірні підходи до документування і захисту інфраструктури недостатніми, що зумовлює необхідність широкого впровадження новітніх цифрових технологій просторового моделювання.

Одним із найперспективніших напрямів є застосування тривимірного (3D) моделювання та цифрових двійників. Цифровий двійник – це динамічна віртуальна копія фізичного об'єкта, яка відображає його стан у реальному часі завдяки інтеграції з мережею сенсорів та IoT-пристроїв [2]. На відміну від статичних тривимірних моделей, цифровий двійник безперервно оновлюється разом зі своїм

фізичним аналогом. У сфері критичної інфраструктури ця технологія дозволяє не лише здійснювати постійний моніторинг об'єктів, а й прогнозувати потенційні збої, імітувати сценарії загроз та розробляти плани реагування без ризику для реальних систем [3].

Важливу роль відіграє технологія BIM (Building Information Modeling) – інформаційне моделювання будівель. BIM-модель містить не лише геометричні параметри споруди, але й комплексні дані про інженерні системи, матеріали та строки технічного обслуговування [4]. Це робить BIM потужним інструментом для розробки планів евакуації, оцінки вразливостей та прогнозування наслідків ударів по об'єктах. Аналіз можливостей BIM у контексті повоєнної відбудови України свідчить про значний потенціал цієї технології як для відновлення зруйнованих споруд, так і для підвищення їхньої стійкості до майбутніх загроз [4].

Поряд із BIM розвивається технологія 3D-лазерного сканування, що дозволяє з високою точністю фіксувати поточний стан об'єктів у вигляді «хмар точок». Дослідження в межах проєкту з відновлення об'єктів водопостачання після руйнування Каховської ГЕС показали: сканери Leica BLK360 забезпечують точність цифрових моделей із геометричними відхиленнями до 10 мм, а подальша обробка в середовищах Leica Cyclone та CloudCompare дозволяє виявляти приховані деформації конструкцій [5]. Така точність критично важлива не лише для контролю якості будівельних робіт, а й для юридично значущого документування пошкоджень в умовах воєнного часу.

Питання захисту самих цифрових моделей не менш важливе для безпеки держави. 3D-моделі об'єктів критичної інфраструктури містять чутливу інформацію про розташування ключових вузлів, параметри захисних систем та слабкі місця конструкцій, яка в разі витоку може бути використана для планування кінетичних або кібератак [3]. Традиційні централізовані механізми управління доступом виявляються недостатніми для захисту розподілених систем цифрових двійників, тому дослідники пропонують більш досконалі архітектурні рішення – зокрема, на основі технологій блокчейн та смарт-контрактів [6].

Геоінформаційні системи (ГІС) суттєво доповнюють можливості 3D-моделювання, прив'язуючи тривимірні об'єкти до реальної місцевості та аналізуючи просторові взаємозв'язки між ними. Платформа ArcGIS надає інструменти для комплексного аналізу безпекових загроз, планування операцій реагування та координації дій між відомствами [7]. ГІС-технології дозволяють досліджувати небезпечні зони без безпосереднього ризику для фахівців і вибудовувати обґрунтовану пріоритетність відновлювальних робіт [8]. Таким чином, поєднання 3D-моделювання та ГІС формує єдиний інформаційний простір для підтримки прийняття рішень.

Практичну цінність 3D-технологій в умовах воєнного конфлікту підтверджує проєкт «War in 3D – Ukraine» команди AERO3D, що документує наслідки агресії шляхом сканування та фотограмметрії зруйнованих міст, будівель і об'єктів інфраструктури [9]. Отримані моделі служать плановою основою для відбудови та

доказовою базою у справах про воєнні злочини перед міжнародними інстанціями. У рамках ініціативи 3D-4CH через платформу Europeana оприлюднено перші 3D-моделі пам'яток з Одеси, Херсонщини, Харківщини та Сумщини, виготовлені методом екстреної фотограмметрії [10]. Цей досвід показує, що 3D-технології ефективно поєднують завдання безпекового документування, планування відновлення та збереження культурної спадщини.

Провідні армії світу активно впроваджують технологію цифрових двійників для планування операцій, управління ресурсами та підготовки особового складу. Зокрема, Міністерство оборони США використовує цифрових двійників для моделювання мережевого середовища, безпечного тестування конфігурацій систем кібербезпеки, включно з міжмережевими екранами та правилами контролю доступу в архітектурах Zero Trust, а також для прогнозування вразливостей без ризику для реальної інфраструктури. Реальні успіхи вже зафіксовані: ВПС США за допомогою цифрових двійників досягли покращення стійкості мереж на 100-275% і підвищення продуктивності на 100-400% [11]. Паралельно цифрові двійники дозволяють моделювати маневри, логістику, артилерійські операції та реакцію супротивника із застосуванням агентного моделювання і теорії ігор, що дає командирам змогу обирати оптимальну стратегію ще до початку бою [12].

Комплексне впровадження 3D-моделювання, надійного кіберзахисту та законодавчого врегулювання статусу цифрових моделей як об'єктів критичної інфраструктури є нагальним стратегічним завданням для України в умовах тривалого воєнного конфлікту.

Список використаних джерел

1. Про критичну інфраструктуру : Закон України від 16.11.2021 № 1882-IX. URL: https://zakon.rada.gov.ua/laws/show/1882-20 (дата звернення: 25.03.2026).
2. Цифровий двійник: контроль, ефективність, безпека [Електронний ресурс] / SmartEAM. URL: https://smart-eam.com/ua/news/cifrovoj-dvojnik-kontrol-jeffektivnost-bezopasnost/ (дата звернення: 25.03.2026).
3. Malatji M., Marnewick A. Beyond the Mirror: Digital Twin Cybersecurity for National Resilience. Konbin. 2025. Vol. 55. DOI: 10.2478/kbo-2025-0010.
4. Використання інформаційного моделювання будівель (BIM) у повоєнній відбудові України [Електронний ресурс] / AIN.UA. URL: https://ain.ua/2023/06/15/vykorystannya-informaczijnogo-modelyuvannya-budivel-bim-u-povoyennij-vidbudovi-ukrayiny/ (дата звернення: 25.03.2026).
5. 3D-сканування як інструмент контролю якості будівництва об'єктів критичної інфраструктури [Електронний ресурс]. Сучасні технології в будівництві. 2025. № 113. DOI: 10.33042/2311-7257.2025.113.1.17. URL: https://svc.kname.edu.ua/index.php/svc/uk/article/view/1927 (дата звернення: 25.03.2026).
6. Смарт-контракти як механізми забезпечення приватності в розподілених системах цифрових двійників критичної інфраструктури [Електронний ресурс]. Кібербезпека: освіта, наука, техніка. 2025. URL:

https://www.csecurity.kubg.edu.ua/index.php/journal/article/view/867 (дата звернення: 25.03.2026).
7. ArcGIS для оборони та безпеки [Електронний ресурс] / Ecomm Co. URL: https://ecomm.in.ua/uk/uk/arcgis-dlya-oborony-ta-bezpeky (дата звернення: 25.03.2026).
8. Геопросторові технології для оцінки наслідків війни [Електронний ресурс] / NECU. URL: https://necu.org.ua/geoprostorovi-tehnologiyi-dlya-oczinky-naslidkiv-vijny/ (дата звернення: 25.03.2026).
9. War in 3D – Ukraine [Електронний ресурс] / AERO3D. URL: https://aero3d.ua/en/save-ukraine-in-3d/war-in-3d (дата звернення: 25.03.2026).
10. First Ukrainian 3D Models published on Europeana through 3D-4CH project [Електронний ресурс] / European Commission. URL: https://digital-strategy.ec.europa.eu/en/library/first-ukrainian-3d-models-published-europeana-through-3d-4ch-project (дата звернення: 25.03.2026).
11. Zhou Yun. Digital twins: fostering efficient network modernization [Електронний ресурс] / Tyto Athene // Military Embedded Systems. – 2025. – URL: https://militaryembedded.com/comms/communications/digital-twins-fostering-efficient-network-modernization (дата звернення: 26.03.2026).
12. Next-generation combat planning: expert explains the role of digital twins in modern armies [Електронний ресурс] // Army Inform. – 2025. – URL: https://armyinform.com.ua/en/2025/12/10/next-generation-combat-planning-expert-explains-the-role-of-digital-twins-in-modern-armies/ (дата звернення: 26.03.2026).

Ключові слова: критична інфраструктура, 3D-моделювання, цифровий двійник, BIM-технології, лазерне сканування, кібербезпека, геоінформаційні системи, національна безпека, воєнні загрози, цифрова трансформація

Keywords: critical infrastructure, 3D modeling, digital twin, BIM technologies, laser scanning, cybersecurity, geoinformation systems, national security, military threats, digital transformation

ШТУЧНИЙ ІНТЕЛЕКТ У КІБЕРЗЛОЧИННОСТІ: ПРОБЛЕМИ ЕФЕКТИВНОСТІ КРИМІНАЛЬНО-ПРАВОВОГО ЗАХИСТУ

Пунтусова Ілона

студентка 1-го курсу факультету прокуратури та слідства (кримінальна юстиція) Національного університету «Одеська юридична академія»

Технології штучного інтелекту (ШІ) дедалі активніше використовуються у цифровому середовищі та впливають на різні сфери суспільного життя, що зумовлює появу нових проблем у сфері кримінально-правового захисту. Якщо раніше кіберзлочинність переважно асоціювалася з несанкціонованим доступом до інформаційних систем, створенням шкідливого програмного забезпечення або викраденням даних, то сьогодні злочинні схеми дедалі частіше базуються на використанні генеративних моделей, алгоритмів машинного навчання та автоматизованих систем аналізу інформації. Особливу небезпеку становить те, що

технології ШІ дозволяють зменшити витрати часу й ресурсів на вчинення злочинів та одночасно збільшити масштаб потенційної шкоди.

Чинний Кримінальний кодекс України передбачає відповідальність за низку правопорушень у сфері використання інформаційних технологій, зокрема за статтями 190, 361 та 361-1 КК України [1].

Водночас сучасні технології ШІ створюють нові механізми вчинення злочинів, які не завжди охоплюються традиційними конструкціями кримінально-правового регулювання. Унаслідок цього розвиток цифрової злочинності випереджає розвиток законодавства та механізмів державного реагування.

Як зазначають С.І. Хом'яченко та М.Є. Дроздов, використання технологій штучного інтелекту у злочинній діяльності формує систему міжгалузевих ризиків, що виходять за межі традиційного кримінального права [2, с. 460]. Науковці звертають увагу на здатність сучасних інтелектуальних систем автоматизувати значну частину кіберзлочинної діяльності – від фішингових кампаній до створення фальсифікованого мультимедійного контенту. У результаті зростає економічна ефективність злочинів, оскільки витрати на їх реалізацію зменшуються, а потенційний масштаб шкоди – збільшується.

Одним із найпоширеніших прикладів використання ШІ у злочинній діяльності є автоматизовані фішингові атаки. Генеративні мовні моделі дозволяють створювати персоналізовані повідомлення, які майже не відрізняються від офіційної комунікації банківських установ, державних органів або великих компаній [2, с. 462]. Якщо раніше шахрайські розсилки часто містили мовні помилки та легко ідентифікувалися користувачами, то сучасні генеративні системи здатні формувати переконливий текст із врахуванням індивідуальних особливостей потенційної жертви. Це ускладнює виявлення шахрайства та підвищує результативність злочинних схем.

Важливою особливістю сучасних злочинів із використанням ШІ є високий рівень масштабованості. Автоматизовані системи можуть одночасно здійснювати тисячі однотипних операцій без значного збільшення витрат. Унаслідок цього виникає асиметрія між витратами злочинців та витратами держави. Для запуску автоматизованої шахрайської кампанії достатньо відносно невеликих ресурсів, тоді як правоохоронні органи змушені витрачати значний час і кошти на розслідування таких правопорушень.

Окрему небезпеку становить використання дипфейк-технологій. Сучасні системи ШІ дозволяють генерувати відео- та аудіоматеріали, які імітують зовнішність, голос або поведінку конкретної особи. Такі технології можуть використовуватися для фінансового шахрайства, маніпуляцій на ринку, дискредитації суб'єктів господарювання або фальсифікації доказів у судовому процесі. Як зазначають С.І. Хом'яченко та М.Є. Дроздов, чинне процесуальне законодавство не містить достатньо ефективних механізмів перевірки автентичності цифрового контенту, створеного за допомогою ШІ [2, с. 462].

У цьому контексті важливого значення набуває стаття 96 Господарського процесуального кодексу України, яка регулює використання електронних доказів [3].

Однак розвиток генеративних технологій демонструє, що традиційні підходи до оцінки цифрових доказів поступово втрачають ефективність. Якщо раніше електронний файл сприймався як відносно надійне джерело інформації, то сьогодні генеративний ШІ дозволяє створювати високоякісні фальсифікації, виявлення яких потребує спеціальних знань та експертних процедур.

Серйозною проблемою є також невідповідність чинного кримінального законодавства сучасним технологічним реаліям. Положення Розділу XVI Кримінального кодексу України формувалися в період, коли генеративні моделі та автономні системи ще не набули поширення. Унаслідок цього законодавство орієнтоване переважно на класичні форми втручання в роботу інформаційних систем, тоді як нові технології ШІ часто використовуються без безпосереднього порушення функціонування комп'ютерних мереж.

Додаткову складність створює проблема встановлення суб'єкта злочину та причинно-наслідкового зв'язку між діями особи і результатами функціонування автономної системи. Використання ШІ частково розмиває традиційні уявлення про вину та механізм вчинення кримінального правопорушення. У деяких випадках інтелектуальні системи можуть самостійно генерувати рішення або модифікувати алгоритми поведінки в межах поставлених завдань.

Базовим нормативним актом у сфері забезпечення кібербезпеки є Закон України «Про основні засади забезпечення кібербезпеки України» [4].

Закон визначає фундаментальні поняття кіберзахисту, кіберзагроз та кіберінцидентів, однак практично не містить спеціальних положень щодо використання технологій штучного інтелекту. Подібна ситуація спостерігається і щодо Стратегії кібербезпеки України, затвердженої Указом Президента України від 26.08.2021 №447/2021 [5].

Документ формувався ще до масштабного поширення генеративного ШІ та не передбачає спеціалізованих механізмів реагування на нові цифрові загрози.

Проблема набуває особливої актуальності з огляду на транснаціональний характер сучасної кіберзлочинності. Багато злочинів із використанням ШІ здійснюються одночасно на території кількох держав, що ускладнює міжнародне співробітництво правоохоронних органів. Крім того, генеративні системи можуть використовуватися анонімно або через інфраструктуру, розташовану в інших юрисдикціях.

На міжнародному рівні проблема використання ШІ у сфері кібербезпеки також активно обговорюється. Європейське агентство з кібербезпеки (ENISA) звертає увагу на ризики використання генеративних моделей для автоматизації кіберзлочинної діяльності та дезінформаційних кампаній [6].

Подібні застереження містяться і в аналітичних матеріалах Europol, де підкреслюється, що сучасні технології ШІ спрощують реалізацію складних злочинних схем та створюють нові загрози для цифрової безпеки держав [7].

У зв'язку з цим виникає необхідність оновлення кримінального та процесуального законодавства України. Одним із можливих напрямів удосконалення є запровадження спеціальної кваліфікуючої ознаки – «вчинення кримінального правопорушення з використанням технологій штучного інтелекту». Це дозволило б враховувати підвищений рівень суспільної небезпеки таких діянь та їхню здатність до масштабування.

Доцільним є запровадження кримінальної відповідальності за створення та поширення шкідливих дипфейків, а також механізмів перевірки автентичності цифрових доказів, створених за допомогою ШІ.

Підсумовуючи викладене, слід зазначити, що технології ШІ підвищують ефективність кіберзлочинності та ускладнюють її виявлення. Чинне кримінальне і процесуальне законодавство України не повною мірою враховує специфіку генеративних систем, тоді як витрати держави на розслідування таких злочинів постійно зростають.

Список використаних джерел

1. Кримінальний кодекс України: Закон України від 05.04.2001 №2341-III. URL: https://zakon.rada.gov.ua/laws/show/2341-14 (дата звернення: 30.04.2026).
2. Хом'яченко С.І., Дроздов М.Є. Кіберзлочини з використанням штучного інтелекту: виклики для кримінального права та господарського процесу України //Аналітично-порівняльне правознавство. 2026. №2, ч. 2. С. 460–464. DOI: 10.24144/2788-6018.2026.02.2.72 (дата звернення: 30.04.2026).
3. Господарський процесуальний кодекс України: Закон України від 06.11.1991 №1798-XII. URL: https://zakon.rada.gov.ua/laws/show/1798-12 (дата звернення: 30.04.2026).
4. Про основні засади забезпечення кібербезпеки України: Закон України від 05.10.2017 №2163-VIII. URL: https://zakon.rada.gov.ua/laws/show/2163-19 (дата звернення: 30.04.2026).
5. Стратегія кібербезпеки України: Указ Президента України від 26.08.2021 №447/2021. URL: https://zakon.rada.gov.ua/laws/show/447/2021 (дата звернення: 30.04.2026).
6. European Union Agency for Cybersecurity (ENISA). URL: https://www.enisa.europa.eu (дата звернення: 30.04.2026).
7. Europol Cybercrime Reports. Europol. URL: https://www.europol.europa.eu/crime-areas-and-statistics/crime-areas/cybercrime (дата звернення: 30.04.2026).

Ключові слова: штучний інтелект, кіберзлочинність, дипфейки, електронні докази, кримінальна відповідальність, кібербезпека

Keywords: artificial intelligence, cybercrime, deepfakes, electronic evidence, criminal liability, cybersecurity

Науковий керівник: доцент Чанишев Р.І.

ІНФОРМАЦІЙНА БЕЗПЕКА ТА ЗАХИСТ ІНФОРМАЦІЇ: СОЦІАЛЬНИЙ, ПРАВОВИЙ І ТЕХНІЧНИЙ АСПЕКТ

Пустовий Сергій

студент 1-го курсу факультету прокуратури та слідства (кримінальна юстиція) Національного університету «Одеська юридична академія»

Інформація сьогодні має важливе значення для функціонування держави, економіки та суспільства. Активний розвиток мережі Інтернет, електронних сервісів, соціальних мереж і хмарних технологій суттєво спростив обмін даними та доступ до інформаційних ресурсів. Водночас поширення цифрових технологій спричинило появу нових ризиків, пов'язаних із кібератаками, витоком даних, дезінформацією та незаконним втручанням у роботу інформаційних систем. За таких умов забезпечення інформаційної безпеки набуває особливого значення для захисту державних інтересів, стабільної роботи органів влади та реалізації прав громадян [1, с. 51].

Особливої актуальності проблема захисту інформації набула для України в умовах повномасштабної війни та постійних кібератак і дезінформаційних кампаній з боку російської федерації. Інформаційний простір став одним із головних напрямів ведення гібридної війни, де поряд із військовими методами активно використовуються інформаційно-психологічні операції, кібератаки, маніпулятивні повідомлення та пропаганда [2, с. 31].

Інформаційна безпека визначається як стан захищеності інформаційного середовища, за якого забезпечуються конфіденційність, цілісність і доступність інформації, а також запобігається негативний вплив інформаційних загроз на особу, суспільство та державу. Сучасне розуміння інформаційної безпеки виходить далеко за межі технічного захисту комп'ютерних систем і охоплює соціальні, правові, психологічні та організаційні аспекти.

Соціальний аспект інформаційної безпеки пов'язаний насамперед із впливом інформації на свідомість людини та суспільства. Соціальні мережі, месенджери та відеохостинги стали основними джерелами отримання інформації для значної частини населення. Водночас саме через них активно поширюються фейкові новини, маніпулятивний контент і дезінформація. Особливу небезпеку становлять інформаційно-психологічні операції, спрямовані на деморалізацію населення, створення панічних настроїв та підрив довіри до державних інституцій.

Однією з причин поширення дезінформації є недостатній рівень медіаграмотності населення. Користувачі часто не перевіряють достовірність джерел, поширюють неперевірені повідомлення та стають жертвами шахрайських схем. Саме тому важливим напрямом державної політики є формування цифрової грамотності, розвиток критичного мислення та навчання громадян правилам безпечної поведінки в інформаційному середовищі [3, с. 15].

Соціальні мережі створюють також ризики витоку персональних даних. Користувачі нерідко добровільно розміщують конфіденційну інформацію у відкритому доступі, що може бути використано зловмисниками для соціальної інженерії, фінансового шахрайства або незаконного збору інформації. У сучасних умовах персональні дані стали цінним цифровим ресурсом, який потребує належного правового й технічного захисту.

Правовий аспект інформаційної безпеки полягає у формуванні системи нормативно-правових актів, які регулюють інформаційні відносини та визначають механізми захисту інформації. Основу правового регулювання в Україні становлять Конституція України, закони України «Про інформацію», «Про захист інформації в інформаційно-комунікаційних системах», «Про основні засади забезпечення кібербезпеки України», «Про державну таємницю» та інші нормативні акти [4]

Законодавство України гарантує право громадян на доступ до інформації, але водночас передбачає захист державної таємниці, персональних даних та іншої інформації з обмеженим доступом. Одним із важливих завдань держави є забезпечення балансу між свободою слова та необхідністю протидії інформаційним загрозам.

Важливим напрямом правового забезпечення є боротьба з кіберзлочинністю. Незаконне втручання в роботу інформаційних систем, поширення шкідливого програмного забезпечення, викрадення даних та кібершахрайство створюють значні ризики для держави, бізнесу та громадян. У зв’язку з цим виникає необхідність постійного вдосконалення законодавства та його гармонізації з міжнародними стандартами, зокрема положеннями Будапештської конвенції про кіберзлочинність.

Важливу роль у забезпеченні цифрової безпеки держави відіграють Державна служба спеціального зв’язку та захисту інформації України, Служба безпеки України, Національний координаційний центр кібербезпеки та інші державні органи. Їх діяльність спрямована на захист державних інформаційних ресурсів, координацію дій у сфері кіберзахисту та реагування на кіберінциденти [5, с. 6].

Технічний аспект інформаційної безпеки охоплює систему програмних, апаратних та організаційних заходів, спрямованих на запобігання несанкціонованому доступу до інформації, її модифікації, знищенню або витоку. Основними принципами технічного захисту залишаються конфіденційність, цілісність та доступність даних [6, с. 18].

Серед основних засобів технічного захисту використовуються системи шифрування, антивірусне програмне забезпечення, міжмережеві екрани, багатофакторна автентифікація, резервне копіювання даних та системи виявлення вторгнень. Особливого значення набуває захист хмарних сервісів, мобільних пристроїв і корпоративних мереж, оскільки значна частина інформації зберігається саме у цифровому середовищі.

Однією з найбільш небезпечних загроз залишаються кібератаки на об'єкти критичної інфраструктури. В Україні неодноразово фіксувалися атаки на енергетичний сектор, банківські установи та державні інформаційні ресурси. Такі інциденти свідчать про необхідність постійного вдосконалення систем технічного захисту та підготовки кваліфікованих фахівців у сфері кібербезпеки.

Важливим напрямом технічного захисту є забезпечення безпеки каналів зв'язку та телекомунікаційних систем. Несанкціонований доступ до каналів передачі інформації може призвести до витоку конфіденційних даних або перехоплення службової інформації. Для запобігання таким загрозам застосовуються криптографічні методи захисту та сучасні системи моніторингу мереж [7, с. 143].

Інформаційна безпека має також міжнародний характер, оскільки кіберзагрози не обмежуються державними кордонами. Україна активно співпрацює з Європейським Союзом і НАТО у сфері кібербезпеки, бере участь у міжнародних програмах обміну інформацією про кіберінциденти та впроваджує міжнародні стандарти захисту інформації.

Таким чином, інформаційна безпека є комплексною сферою, яка поєднує соціальні, правові та технічні аспекти захисту інформації. Сучасні інформаційні загрози потребують системного підходу до захисту цифрового середовища держави та суспільства. Ефективне забезпечення інформаційної безпеки можливе лише за умови поєднання сучасних технічних рішень, дієвого законодавства, міжнародного співробітництва та високого рівня інформаційної культури населення. В умовах глобальної цифровізації та гібридної війни питання інформаційної безпеки стає одним із ключових чинників стабільного розвитку держави та захисту національних інтересів України.

Список використаних джерел

1. Торяник В.М. Інформаційна безпека як складова національної безпеки держави. Роль ЗМІ в забезпеченні інформаційного суверенітету України //Науковий вісник Національної академії внутрішніх справ. 2015. №6-2. С. 51–56. URL: https://elar.navs.edu.ua/server/api/core/bitstreams/583213ad-b22e-417b-8171-c38b68d1e5a6/content (дата звернення: 17.05.2026).
2. Ільницька У. Інформаційна безпека України: сучасні виклики, загрози та механізми протидії негативним інформаційно-психологічним впливам //Гуманітарні візії. 2016. Т. 2, №1. С. 27–32. URL: https://science.lpnu.ua/sites/default/files/journal-paper/2017/jun/4352/ilnicka0.pdf (дата звернення: 17.05.2026).
3. Мехед Д. Інформаційна безпека в соціальних мережах. Методи поширення інформації в соціальних мережах //Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2015. Вип. 2(30). С. 14–18. URL: https://ela.kpi.ua/server/api/core/bitstreams/0d0be072-5226-4ac7-a421-95fb2b1e978a/content (дата звернення: 17.05.2026).
4. Про основні засади забезпечення кібербезпеки України : Закон України від 05.10.2017 № 2163-VIII Відомості Верховної Ради України 2017. №45. Ст. 403. (дата звернення: 17.05.2026).

5. Фролов О. Система технічного захисту інформації в Україні. Стан та перспективи розвитку //Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2008. Вип. 1(16). С. 6–7. URL: https://core.ac.uk/download/pdf/47227963.pdf (дата звернення: 17.05.2026).
6. Інформаційна безпека та захист даних в комп'ютерних технологіях і мережах. Вибрані розділи: навч. посіб. /уклад.: В.П. Полторак, О.В. Савчук. Київ: КПІ ім. Ігоря Сікорського, 2021. 384 с. URL: https://ela.kpi.ua/items/5443ea35-dd2c-4081-aba0-744852a23c2f (дата звернення: 17.05.2026).
7. Цибуляк Б. З. Захист інформації від витоку каналами телефонного зв'язку //Вісник НТУУ «КПІ». Радіотехніка, радіоапаратобудування. 2013. № 55. С. 143–148. URL: https://surli.cc/lbbgxj (дата звернення: 17.05.2026).

Ключові слова: інформаційна безпека, кібербезпека, захист інформації, інформаційні загрози, кіберзлочинність, дезінформація, персональні дані, цифрова безпека

Keywords: information security, cybersecurity, information protection, information threats, cybercrime, disinformation, personal data, digital security

Науковий керівник: доцент Чанишев Р.І.

Секція 3. УПРАВЛІНСЬКІ ТА ЕКОНОМІЧНІ АСПЕКТИ РОЗВИТКУ СУЧАСНОЇ ІТ ІНДУСТРІЇ

СТРАТЕГІЧНЕ УПРАВЛІННЯ ІННОВАЦІЯМИ В ІТ-ІНДУСТРІЇ: УКРАЇНСЬКИЙ ДОСВІД

Горбаченко Станіслав

доктор економічних наук., професор,
завідувач кафедри штучного інтелекту та математичного моделювання
Національного університету "Одеська юридична академія"

Інформаційні технології посідають центральне місце в економічному розвитку сучасної України, формуючи значну частину експорту послуг та забезпечуючи робочі місця для тисячі фахівців. Динамічна природа галузі та зовнішні шоки – від глобальних криз до повномасштабної війни – ставлять перед українськими ІТ-компаніями завдання, які неможливо розв'язати без системного підходу до управління інноваціями. За таких умов стратегічне управління інноваціями стає ключовим інструментом, який дозволяє компаніям не лише адаптуватися, а й створювати нові ринкові переваги.

Сучасна наукова література визначає стратегічне управління інноваціями як інтегровану систему планування, організації та контролю інноваційних процесів задля досягнення довгострокових конкурентних переваг. Дослідники наголошують, що інтеграція інновацій до стратегічного управління дозволяє скоротити час на розробку продуктів, підвищити ефективність операцій та забезпечити більш гнучке реагування на ринкові зміни [1]. Окремою категорією виступають проривні інновації, які суттєво впливають на управління конкурентоспроможністю підприємств ІТ-сфери та вимагають перегляду традиційних теорій конкурентної переваги [2].

Українська ІТ-індустрія пройшла унікальний шлях розвитку: від виконання простих аутсорсингових завдань до повного циклу розробки інноваційних продуктів. Такі компанії, як MacPaw, Genesis, GitLab та Ajax, перейшли від простого аутсорсингу до надання послуг повного циклу розробок. Аналіз трансформації ІКТ-сектору в Україні засвідчує, що ключовим фактором сталого розвитку галузі є формування інноваційних екосистем, які об'єднують компанії, освітні заклади, інвесторів та державу [3].

Важливим інструментом державної підтримки інноваційного розвитку став спеціальний правовий режим Дія City, який створює передумови для зростання інвестиційної привабливості українських компаній завдяки інструментам англійського права, спрощеному оподаткуванню та альтернативній моделі найму. Сучасні дослідження стратегій економічного розвитку ІТ-компаній рекомендують стратегічне фокусування на перспективних технологічних нішах, збільшення інвестицій у R&D та активну інтеграцію в глобальні інноваційні екосистеми [4].

Повномасштабна війна стала найсерйознішим випробуванням для української IT-індустрії. Незважаючи на безпрецедентні труднощі, галузь продемонструвала високий рівень стійкості. Економічний аналіз стану галузі підкреслює, що в умовах воєнного стану та повоєнного відновлення IT-індустрія стає важливим фактором економічного розвитку та зміцнення обороноздатності країни [5]. Це формує нову стратегічну роль галузі – не лише як експортера послуг, а й як стратегічного активу національної безпеки. Українські IT-компанії розробили інноваційні управлінські практики: децентралізовані команди, забезпечення безперервності бізнесу через автономне живлення та супутниковий інтернет, гнучкі робочі графіки з урахуванням повітряних тривог, психологічна підтримка співробітників.

Аналіз стратегічних напрямів дозволяє виокремити кілька ключових векторів розвитку: фокусування на високоінтелектуальних нішах (штучний інтелект, кібербезпека, фінтех, медичні IT, технології подвійного призначення); інтеграція в європейський цифровий простір через Horizon Europe та Digital Europe Programme; розбудова власних R&D-центрів і продуктів замість аутсорсингової моделі.

Український досвід демонструє, що ефективне стратегічне управління інноваціями в IT-індустрії вимагає системного підходу, який інтегрує державну політику, корпоративні стратегії, інвестиції у R&D та екосистемне мислення. Подальші дослідження доцільно спрямувати на формалізацію цього досвіду та розробку рекомендацій для повоєнного відновлення галузі.

Список використаних джерел

1. Гвоздь М., Олинець А.-М., Остащук Р. Синергія стратегічного управління та інновацій для розвитку підприємства в умовах цифрової економіки. *Цифрова економіка та економічна безпека*. 2024. № 5 (14). С. 110-115. DOI: https://doi.org/10.32782/dees.14-17.
2. Босовська М., Прожогін Д. Проривні інновації у стратегічному розвитку IT-сфери. *Scientia fructuosa*. 2026. № 2. 89–105. DOI:https://doi.org/10.31617/1.2026(166)07.
3. Алексеєвська Г., Чайковська М. Трансформація ікт-сектору в україні: аналіз тенденцій та стратегії сталого розвитку. *Економіка та суспільство*. 2024. № 60. DOI: https://doi.org/10.32782/2524-0072/2024-60-98.
4. Пилипенко О., Процко Я. СТратегії економічного розвитку іт-компаній у контексті структурних перетворень. *Вчені записки Університету «КРОК»*. 2025. № 4 (80). С. 71–80. DOI: https://doi.org/10.31732/2663-2209-2025-80-71-80.
5. Гайда Ю., Верета Н. Проблеми та перспективи розвитку IT-індустрії в Україні під час воєнного стану і повоєнного відновлення економіки. *Економічний аналіз*. 2024. № 34(1). С. 315–325. DOI: https://doi.org/10.35774/econa2024.01.315.

Ключові слова: стратегічне управління, інновації, IT-індустрія, Україна, R&D, Дія City, цифрова трансформація

Keywords: strategic management, innovation, IT industry, Ukraine, R&D, Diia City, digital transformation

ЕКОНОМІЧНІ ТА УПРАВЛІНСЬКІ АСПЕКТИ ВПРОВАДЖЕННЯ CLOUD-ТЕХНОЛОГІЙ У ДЕРЖАВНОМУ СЕКТОРІ

Разінкін Нікіта

асистент кафедри штучного інтелекту та математичного моделювання Національного університету "Одеська юридична академія"

Цифрова трансформація державного сектору є однією з ключових передумов підвищення ефективності публічного управління, якості адміністративних послуг та прозорості функціонування державних інституцій. Серед усього спектру технологічних рішень е-урядування особливе місце посідають хмарні (cloud) технології – вони забезпечують гнучкість обчислювальних ресурсів, високу доступність державних сервісів та можливість швидкого масштабування, що особливо актуально для країни, яка водночас здійснює цифрову модернізацію та функціонує в умовах повномасштабної війни. Аналіз цифрової трансформації економіки України демонструє, що ефективне використання хмарних технологій безпосередньо впливає на якість публічного управління та конкурентоспроможність бізнесу [1].

Cloud-технології передбачають надання обчислювальних ресурсів (серверів, сховищ даних, мережевих можливостей, програмного забезпечення) у вигляді послуги через мережу інтернет за моделями IaaS (інфраструктура як послуга), PaaS (платформа як послуга) та SaaS (програмне забезпечення як послуга). У державному секторі застосовують кілька моделей розгортання: публічна хмара, приватна, гібридна та суверенна (sovereign cloud), яка передбачає розміщення даних під юрисдикцією конкретної держави. Вибір моделі визначається співвідношенням економічної ефективності, рівня кібербезпеки та чутливості оброблюваних даних.

Економічна привабливість хмарних рішень для держави полягає у переході від моделі капітальних видатків (CAPEX) на побудову власних центрів обробки даних до операційних видатків (OPEX) за фактичним споживанням ресурсів. Це дозволяє суттєво знизити сукупну вартість володіння IT-інфраструктурою (TCO), скоротити цикл впровадження нових електронних сервісів та оптимізувати штат технічного персоналу. Наукові дослідження економічних аспектів використання хмарних технологій підкреслюють, що їх впровадження повинне бути нормативно врегульованим, безпечним, економічно ефективним та максимально враховувати індивідуальні потреби суб'єкта хмарної трансформації [2]. Поряд із прямою економією бюджету, хмарні технології створюють непрямі економічні ефекти: пришвидшення надання послуг громадянам, зменшення корупційних ризиків через автоматизацію процесів та підвищення інвестиційної привабливості економіки.

Український досвід міграції державних даних у хмару став одним з найбільш цитованих кейсів у світі. У перші тижні повномасштабного вторгнення українська

держава здійснила безпрецедентну операцію з міграції критичних даних та сервісів у хмарні інфраструктури за межами країни. Дослідження європейської практики впровадження хмар для потреб національної безпеки констатує, що російські сили цілеспрямовано атакували українські центри обробки даних, використовуючи як кінетичні удари, так і кібератаки, з метою порушити функціонування цифрових сервісів, а попередня хмарна міграція забезпечила безперервність роботи уряду [3]. Цей кейс довів, що хмарні технології є не лише економічним інструментом, а й елементом стратегічної стійкості держави.

Управлінські аспекти впровадження хмарних технологій у держсекторі охоплюють кілька ключових напрямів: цифровий суверенітет та контроль над даними (визначення категорій інформації, що можуть оброблятися в комерційних хмарах); ризик технологічної залежності від одного постачальника (vendor lock-in); питання кібербезпеки – концентрація даних робить хмарні платформи привабливою мішенню для атак. Експертні дослідження звертають увагу на необхідність залучення представників уряду, цифрової індустрії, постачальників хмарних послуг та громадянського суспільства для досягнення консенсусу щодо нормативного регулювання обробки даних державних реєстрів у комерційних хмарах [4].

Правовий вимір впровадження cloud-технологій у державному секторі регулюється Законом України «Про хмарні послуги», який встановлює базові правила надання та використання таких послуг публічними користувачами – державними органами, органами місцевого самоврядування, а також державними підприємствами та установами. Аналіз правового регулювання ринку хмарних послуг України свідчить, що подальше формування законодавчої бази є необхідним для нормалізації відносин між постачальниками та споживачами хмарних послуг, а також для наділення компетентних державних органів повноваженнями регулятора та формування державної політики у цій сфері [5]. Без сталого правового поля впровадження хмарних рішень у публічному секторі залишається фрагментарним та залежить від ініціативи окремих відомств.

Хмарні технології формують технологічну основу для впровадження інших інновацій у публічному управлінні – штучного інтелекту, аналітики великих даних, інтернету речей. Аналіз сучасних трендів технологічних інновацій у публічному управлінні підкреслює, що інтеграція цих технологій підвищує ефективність державного управління, але водночас вимагає подолання культурних та організаційних бар'єрів, бюрократичних перешкод і підвищення цифрової обізнаності державних службовців [6]. Отже, впровадження хмарних рішень потребує цілісної управлінської трансформації та інвестицій у людський капітал.

Стратегічні напрями розвитку cloud-технологій у державному секторі України включають: формування національної гібридної хмарної інфраструктури, що поєднує суверенний компонент для критичних реєстрів з міжнародним хмарним резервом; уніфікацію стандартів закупівлі хмарних послуг; запровадження прозорих критеріїв оцінки ROI для проєктів цифровізації; розвиток компетенцій з

cloud-архітектури серед державних службовців; інтеграцію хмарних сервісів з європейською цифровою екосистемою.

Впровадження cloud-технологій у державному секторі України має значний економічний та управлінський потенціал – оптимізацію бюджетних видатків на IT-інфраструктуру, прискорення надання адміністративних послуг та забезпечення стійкості державних сервісів. Цей процес потребує системного підходу, що інтегрує правове регулювання, технічні стандарти, кібербезпеку та цифровий суверенітет.

Список використаних джерел

1. Shcherban T., Hoblyk V., Chernychko T., Pigosh V., Kozyk I. Assessment of the digital transformation of Ukraine's economy: Challenges, opportunities, and strategic prospects. *Scientific Bulletin of Mukachevo State University. Series "Economics"*. 2025. № 12 (1). P. 159-168. DOI: https://doi.org/10.52566/msu-econ1.2025.159.
2. Шевчук І., Депутат Б. Економічний аспект використання хмарних технологій у діяльності органів публічної влади та бізнес-структур. *Економіка та суспільство*. 2021. № 31. DOI: https://doi.org/10.32782/2524-0072/2021-31-26.
3. Jarnecki J. European Cloud Adoption for National Security. Research Papers. RUSI. 2025. 52 p. URL: https://static.rusi.org/european-cloud-adoption-for-national-security.pdf (дата звернення: 18.05.2026).
4. Пазюк А., Слабко Т. Використання комерційних хмарних технологій для обробки даних державних реєстрів України. Міжнародною фундацією виборчих систем (IFES). 2023. URL: https://www.ifesukraine.org/wp-content/uploads/2024/01/ifes-ukraine-the-use-of-commercial-cloud-technologies-for-processing-data-from-state-registers-of-ukraine-1.pdf (дата звернення: 18.05.2026).
5. Koverznev V., Ponomarov S., Ivanov A. . Legal Regulation of the Cloud Services Market of Ukraine. *European Journal of Sustainable Development*. 2024. № 13 (1). P. 73-86. DOI: https://doi.org/10.14207/ejsd.2024.v13n1p73.
6. Бабічев А., Котковський В., Чередниченко Т. Технології та інновації у публічному управлінні: сучасні тренди, вплив на якість державних послуг. *Економіка та суспільство*. 2024. № 65. DOI: https://doi.org/10.32782/2524-0072/2024-65-82.

Ключові слова: хмарні технології, державний сектор, цифрова трансформація, електронне урядування, економічна ефективність, кібербезпека, цифровий суверенітет

Keywords: cloud technologies, public sector, digital transformation, e-government, economic efficiency, cybersecurity, digital sovereignty

МОТИВАЦІЯ ТА УТРИМАННЯ IT-ФАХІВЦІВ В УМОВАХ ГЛОБАЛЬНОЇ КОНКУРЕНЦІЇ ЗА ТАЛАНТИ

Скідан Владислав

асистент кафедри штучного інтелекту та математичного моделювання Національного університету "Одеська юридична академія"

Інформаційні технології є однією з найбільш людиноцентричних галузей сучасної економіки, де ключовим виробничим ресурсом виступає кваліфікований фахівець. У глобальному вимірі ця галузь характеризується гострим дефіцитом талантів, який отримав у міжнародному менеджменті стійку назву «війна за таланти». Для української IT-індустрії цей виклик помножується специфічними чинниками: повномасштабною війною, мобілізаційними процесами, релокацією значної частини фахівців за кордон та посиленням конкуренції з боку польських, румунських і балтійських ринків. У таких умовах ефективна система мотивації та утримання співробітників перетворюється на головний інструмент стратегічного управління людським капіталом IT-компаній.

Сучасна теорія управління талантами (talent management) розглядає роботу з персоналом як цілісний процес, що охоплює ідентифікацію, адаптацію, розвиток, мотивацію та збереження працівників, а також оцінку ефективності всієї системи. Дослідники наголошують, що для збереження талантів необхідна мотивація як у вигляді матеріальних винагород з різноманітними методами систем оплати праці, так і у вигляді нематеріальних заохочень [1]. Серед методів розвитку талановитих працівників особливе значення мають асесмент-центри, коучинг, наставництво та внутрішньокорпоративні навчальні програми, які одночасно виконують функцію мотивації через надання співробітнику відчуття професійного зростання.

Особливості мотиваційного профілю молодих IT-фахівців (покоління Z та молодших мілленіалів) суттєво відрізняються від традиційних HR-моделей. Дослідження мотивації молодих фахівців засвідчують, що мотиваційний профіль цієї категорії має ряд особливостей, пов'язаних з прагненням молоді набути достатній рівень професійних компетенцій, і проявляється у цінностях розвитку [2]. Для молодих IT-фахівців характерним є пріоритет таких цінностей, як гнучкий графік, можливість віддаленої роботи, доступ до сучасних технологічних стеків, участь у проєктах з реальним впливом на продукт, а також прозорі траєкторії кар'єрного зростання. Стабільність робочого місця та розмір зарплати залишаються важливими, але дедалі частіше програють у конкуренції з нематеріальними чинниками.

Український контекст управління IT-талантами у 2022–2026 роках характеризується унікальним поєднанням викликів. За даними галузевих досліджень, у 2024 році українська IT-індустрія втратила близько 2 тисяч фахівців, що на 8 тисяч менше за показник 2023 року, при цьому компанії найняли 7,2 тисячі нових

спеціалістів за друге півріччя [3]. Водночас більшість компаній мали мобілізованих співробітників, що змусило галузь розробити нові HR-практики – гарантоване збереження робочих місць, фіксовані виплати та часткову компенсацію заробітної плати для мобілізованих.

Емпіричні дослідження мотивації персоналу в умовах воєнного стану свідчать, що середні значення показників професійної мотивації перебувають на середньому рівні, а ефективна система мотивації стає одним з ключових факторів підвищення ефективності системи управління підприємством [4]. Для IT-сектору це означає необхідність переходу від традиційних матеріальних стимулів до комплексних програм підтримки, що враховують психоемоційний стан співробітників, ризики мобілізації, проблеми безпеки членів сім'ї та інші специфічні обставини. Серед практик, які довели свою ефективність: психологічна підтримка, гнучкі графіки з урахуванням повітряних тривог, забезпечення автономного живлення та супутникового інтернету, благодійні програми та волонтерська діяльність як корпоративна цінність.

Технологічна трансформація HR-функції відкриває нові можливості для персоналізації систем мотивації. Наукові дослідження використання технологій штучного інтелекту в управлінні персоналом демонструють значний потенціал AI для створення ефективних та інноваційних стратегій мотивації, аналізу залученості співробітників та прогнозування ризиків звільнення [5]. У практиці IT-компаній застосовуються інструменти sentiment-аналізу комунікацій команд, predictive analytics для виявлення фахівців, схильних до звільнення, а також алгоритми для індивідуалізації пакетів компенсацій та навчальних траєкторій.

Важливим виміром утримання талантів є інвестиції у розвиток soft skills та цілісної професійної компетентності фахівців. Аналіз розвитку soft skills у майбутніх фахівців IT-профілю як наукової проблеми підкреслює, що сучасний ринок праці потребує не лише hard-skills (програмування, архітектура систем), а й розвинених комунікативних навичок, емоційного інтелекту, здатності до командної роботи та критичного мислення [6]. Компанії, які інвестують у розвиток цих компетенцій своїх співробітників, отримують подвійний ефект: підвищення продуктивності команд та зростання лояльності фахівців, які цінують можливості внутрішнього зростання.

На основі аналізу літератури та практики українських IT-компаній можна виокремити кілька стратегічних рекомендацій щодо мотивації та утримання фахівців: побудова прозорих кар'єрних траєкторій з чіткими критеріями просування; впровадження гнучких пакетів компенсацій з можливістю індивідуалізації; розвиток внутрішніх програм навчання та сертифікацій; формування культури open feedback та регулярних one-to-one зустрічей; підтримка психологічного добробуту та програм well-being; залучення співробітників до соціально значущих проєктів, у тому числі оборонного характеру; стратегічне використання дистанційного та гібридного формату роботи як конкурентної переваги.

В умовах глобальної конкуренції за IT-таланти ефективна система мотивації та утримання фахівців перестає бути функцією виключно HR-департаменту і перетворюється на стратегічний пріоритет вищого керівництва. Український досвід демонструє, що навіть у складних умовах війни можна зберігати кадровий потенціал галузі, поєднуючи матеріальні стимули з нематеріальними цінностями та інноваційними HR-технологіями. Подальші дослідження доцільно спрямувати на формалізацію українських HR-практик воєнного періоду.

Список використаних джерел

1. Кравченко О., Кравченко Ю. Інвестиція в майбутнє: розвиток та утримання талановитих працівників. Економіка та суспільство. 2024. № 62. DOI: https://doi.org/10.32782/2524-0072/2024-62-71.
2. Данилевич Н. С., Поплавська О. М., Пузиревська Ю. О. Мотивація молодих фахівців: особливості, рекомендації. *Економічний простір*. 2019. № 142. С. 53-64. DOI: http://dx.doi.org/10.30838/P.ES.2224.260219.53.378.
3. Минулого року значно зменшився відтік кадрів в українському IT-секторі - що відомо. URL: https://www.kyivpost.com/uk/post/47399 (дата звереннння: 18.05.2026).
4. Борисюк О. М., Клюка А. І. Мотивація персоналу як фактор підвищення ефективності системи управління в умовах війни. *Науковий вісник Львівського державного університету внутрішніх справ (серія психологічна)*. 2022. № 1. С. 10-16. DOI: https://doi.org/10.32782/2311-8458/2022-1-2.
5. Базака Р. В. The mechanism of using artificial intelligence technologies in the processes of motivation of the company's employees. *Таврійський науковий вісник. Серія: Економіка*. 2024. № 20. С. 33-37. DOI: https://doi.org/10.32782/2708-0366/2024.20.3.
6. Гальчинський В., Дембіцька С. Розвиток soft skills у майбутніх фахівців IT-профілю як наукова проблема. *Педагогіка безпеки*. 2026. № 11 (1). С. 18–25. DOI: https://doi.org/10.31649/2524-1079-2026-11-1-018-025.

Ключові слова: мотивація персоналу, утримання талантів, IT-фахівці, війна за таланти, HR-менеджмент, soft skills, релокація

Keywords: personnel motivation, talent retention, IT specialists, war for talent, HR management, soft skills, relocation

ЗМІСТ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ

МАТЕРІАЛИ XI ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

22 травня 2026 року
м. Одеса

Відповідальній редактор: Н. І. Логінова

Дизайнер: Р. Р. Дробожур

Електронне видання

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

МАТЕРІАЛИ XI ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: «ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ»

22 травня 2026 року, м. Одеса