БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ

№ 1(9)/2025 НАУКОВИЙ ЖУРНАЛ ISSN 2707-1758

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Засновано 2019 року

УДК 004.056
DOI: https://doi.org/10.17721/ISTS.2025.9
ROR: https://ror.org/02aaqv166

Представлено результати досліджень за такими напрямами: інформаційна та кібернетична безпека, комп'ютерні науки та інформаційні технології, інформаційно-комунікаційні системи, телекомунікації та радіотехніка.

Для науковців, докторантів, аспірантів, фахівців відповідних напрямів, студентів.

Адреса редколегії

вул. Богдана Гаврилишина, 24, м. Київ, 04116
☎ (38044) 481 44 07
e-mail: fit@univ.net.ua

Затверджено

вченою радою факультету інформаційних технологій
09.06.25 (протокол № 14)

Зареєстровано

Національною радою України з питань телебачення і радіомовлення
Рішення № 357 від 15.02.24
Індетифікатор друкованого медіа: R30-02757

Атестовано

Міністерством освіти і науки України (категорія Б)
Наказ № 1415 від 02.10.24

Засновник та видавець

Київський національний університет імені Тараса Шевченка
Видавничо-поліграфічний центр "Київський університет"

Свідоцтво внесено до Державного реєстру
ДК № 1103 від 31.10.02

Адреса видавця

ВПЦ "Київський університет"
б-р Тараса Шевченка, 14, м. Київ, 01601
☎ (38044) 239 32 22, 239 31 58, 239 31 28
e-mail: vpc@knu.ua

INFORMATION SYSTEMS AND TECHNOLOGIES SECURITY

№ 1(9)/2025 SCIENTIFIC JOURNAL ISSN 2707-1758

TARAS SHEVCHENKO NATIONAL UNIVERSITY OF KYIV

Established in 2019

UDC 004.056
DOI: https://doi.org/10.17721/ISTS.2025.9
ROR: https://ror.org/02aaqv166

The scientific publication presents the results of research in such areas as Information and Cyber Security, Computer Science and Information Technology, Information and Communication Systems, Telecommunications and Radio Engineering.

For scientists, doctoral students, graduate students, specialists, students.

Address	**24, Bohdan Hawrylyshyn str., Kyiv, 04116** **☎ (38044) 481 44 07** **e-mail: fit@univ.net.ua**
Approved by	**the Academic Council of the Faculty of Information Technology** **09.06.25 (protocol № 14)**
Registered by	**the National Council of Ukraine on Television and Radio Broadcasting** **Decision № 357 of 02.15.24** **Identifier of printed media: R30-02757**
Certified by	**the Ministry of Education and Science of Ukraine (category B)** **Order № 1415 dated 02.10.24**
Foundator and Publisher	**Taras Shevchenko National University of Kyiv** **Publishing and Polygraphic Center "Kyiv University"** **Certificate of entre into the State Register** **ДК № 1103 from 31.10.02**
Address	**PPC "Kyiv University"** **14, Taras Shevchenko blvd., Kyiv, 01601** **☎ (38044) 239 32 22, 239 31 58, 239 31 28** **e-mail: vpc@knu.ua**

ЗМІСТ

КІБЕРБЕЗПЕКА ТА ЗАХИСТ ІНФОРМАЦІЇ

КОМП'ЮТЕРНІ НАУКИ

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

CONTENTS

CYBERSECURITY AND INFORMATION PROTECTION

COMPUTER SCIENCE

INFORMATION SYSTEMS AND TECHNOLOGIES

КІБЕРБЕЗПЕКА ТА ЗАХИСТ ІНФОРМАЦІЇ

UDC 004.056 + 004.738.5 : 004.056.53
DOI: https://doi.org/10.17721/ISTS.2025.9.5-10

Maksym KOTOV, PhD Student
ORCID ID: 0000-0003-1153-3198
e-mail: maksym_kotov@ukr.net
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

CLUSTRER STATE COORDINATION REDUCERS BASED ON RSDP

Background. *Contemporary distributed systems tend to leverage complex network topologies and intercommunication technologies for performing complex computational tasks. It is quite common for such systems to rely on centralized coordination where one or a group of designated servers serve as a management plane for the entire network. While this approach may simplify the consensus process, it introduces additional requirements for infrastructure engineering and maintenance. Nodes serving as a control plane require additional hardware resources as well as human effort and competence to manage external services capable of providing basic coordination primitives. Most cases requiring consensus could be reduced to simplistic procedures like gathering knowledge about network participants and dividing deterministically state slices between them. Therefore, implementing a complex coordination solution that requires an additional maintenance method could be inefficient. The Replica State Discovery Protocol stands as a lightweight coordination solution presenting a simple interface for achieving consensus between nodes within a cluster.*

Methods. *Within the ambit of this research, a set of cluster state reducers is described, providing basic coordination capabilities, including the formation of a network participants list and task division between them. Using mathematical modeling, we describe the procedures necessary for performing the said coordination tasks. Implementation and testing of reducers is done with the Node.js platform capable of running JavaScript code on the server side. A theoretical analysis and description of the proposed methods for distributed coordination are provided within this work to facilitate their integration into modern systems.*

Results. *As a result of this research, we propose three new cluster state reducers serving as methods of basic coordination capabilities as an exemplary application of RSDP. The first reducer is responsible for gathering the directory of participating nodes within a cluster and maintaining their statuses based on the received data. The second reducer performs a timeframe division within a cluster between the nodes to coordinate their execution in a mutually exclusive environment. Lastly, the rate limit reducer describes a logic to perform consensus regarding a single value that should be shared throughout the system as well as promptly updated if needed.*

Conclusions. *While engineering a complex distributed system requiring consensus among its subsystems or a set of homogenous components, it's important to avoid complexities related to the management of additional infrastructure while still providing a required level of consistency and availability. Having said that, the Replica State Discovery Protocol provides essential lightweight capabilities to resolve the said problem through the means of its flexible state reducers system. RSDP is built with a layered architecture in mind, capable of adjusting to the particular needs of the network as shown in this paper. By leveraging existing communication infrastructure and avoiding redundant management layers, RSDP allows for significantly reducing the computational complexity of coordination as well as costs associated with hardware needed for running a dedicated control plane. State reducers described within this article provide basic capabilities required for the most common coordination tasks, including the construction of a participants directory, task splitting and assignments, as well as consensus regarding the configuration parameters.*

Keywords: *distributed computing; device coordination; state synchronization; cluster management; service availability; security incidents; Replica State Discovery Protocol (RSDP); RSDP cluster state reducers.*

Background

The rise of the information technology age has led to the development of numerous automated systems for data gathering and processing. It is quite common nowadays to have shops or government services available directly on personal computers or phones. As a result, the demand for the availability of such services rises every year. The larger the number of clients trying to gain access to the system, the more it is evident that every large system must be capable of scaling both vertically and horizontally.

Vertical scaling is straightforward; increasing the number of CPUs or available RAM can provide a way of improving the system's productivity without modifying the software. Modern server stations can reach significant capabilities in their computational capacities, and the ceiling keeps getting higher. Although, vertical scaling is still quite limited when it comes to high-end, in-demand services such as AWS or Google Cloud. In cases where millions or billions of potential clients must be supported, vertical scaling is limited and will not be sufficient (Ali, 2019).

Horizontal scaling, on the other hand, promises nearly limitless growth potential where there are no commonly shared bottlenecks. Distributed systems primarily rely on such principles to provide both efficiency and availability through redundancy and coordination mechanisms. It comes with the cost of having to establish a consensus mechanism on a software level which is a non-trivial task tackled from the dawn of the global interconnection era (Ali, 2019; Millnert, & Eker, 2020).

With that, the Replica State Discovery Protocol serves as a solution, providing basic simplified interfaces for establishing a consistent and coordinated cluster environment. The purpose of this protocol is to abstract out the common steps required to aggregate a shared state among multiple nodes. Firstly, such steps include the discovery phase called “DEBATE”. Its purpose is to introduce a node to the network by broadcasting its initial configuration along with its identifier and network address. As a result, every node after receiving such notification will respond with its own configuration to complete the discovery process. The following step includes state derivation and consensus based on the statuses received from the peers. After initial aggregation, the nodes share their perspective with other participants to achieve consensus. This essentially is a security mechanism to avoid both accidental and intentional discrepancies within the cluster state (Toliupa et al., 2024; Kotov et al., 2024).

ISSN 2707-1758

Comparing RSDP with other coordination solutions like Apache ZooKeeper and ETCD, the difference becomes apparent (Junqueira, & Reed, 2013; Toasa et al., 2019; Nalawala et al., 2022; Larsson et al., 2020). The said services act as a central management layer distinct from the operational plane. As such, they require additional maintenance and resources tied to the deployment of the control-plane servers. In contrast, RSDP is a lightweight protocol that does not require deployment and management of any additional infrastructure. In that regard, it is more comparable to other consensus protocols such as Paxos and Raft (Kirsch, & Amir, 2008; Hu, & Liu, 2020; Ni et al., 2024). Though, a significant difference is that these protocols serve as a solution to elect a single value within the network, while RSDP, besides consensus as a final stage, provides capabilities to establish complex state derivation logic based on the available cluster perception.

RSDP itself is based on a layered model described within its own dedicated articles. In essence, there are application-level state reducers, a consensus layer, and media access layers. This article primarily concentrates on the first, but for completeness, we will briefly cover other interaction layers. We've already described the consensus layer and its operation principles above, when discussing the steps the protocol engine takes to exchange the information among nodes and establish a consistent environment.

The communication media layer is primarily responsible for providing two simple methods to the upper layers: send to and broadcast. These could be implemented in multifarious ways by leveraging existing interconnection protocols for security or reliability. The initial version of RSDP utilized the AMQP protocol and RabbitMQ implementation to establish such operations. AMQP is an asynchronous communication protocol based on queues and provides complex message routing capabilities for its clients. Though RSDP does not require AMQP nor RabbitMQ, the same logic could be implemented with simple HTTP requests or even based on bare TCP connections, which in effect removes dependency on any additional external services and reduces complexities tied to their management.

Having said that, within this article, the aim is to develop a set of coordinating state reducers in cases where interval services rely upon the availability of external servers. That is a quite common pattern nowadays, when most applications are some sort of combination of external tools. Such tools often implement their own security policies to guarantee availability in the face of potential DOS attacks. Hence, they implement various rate-limiting strategies to stifle any potential malicious activities. On top of that, it is quite common for such external services to provide their internal state data as snapshots. For example, some aggregates of financial or crypto-related information on centralized exchanges have limited historical availability or none.

In such cases it is important to establish data redundancy techniques where multiple internal processes within the controlled environment poll the external data provider. In case one polling process fails, the other would gather information successfully. Implementation of such logic becomes quite complex since the external service might impose rate-limiting restrictions, which necessitate the creation of internal coordination logic to avoid violation of such policies by a pool of uncoordinated parallel execution entities.

The purpose of the article. The purpose of this article is to provide examples of potential RSDP application within coordination-related problems. To do so, we've chosen a commonly met problem while designing a system requiring data redundancy and at the same time relying on third-party servers. An architectural description of such a case is provided with diagrams to establish the necessary context. This paper presents three state reducers providing functionality required to establish a managed and compliant gathering of data from an external limited source.

Starting with a directory-gathering reducer, responsible for maintaining a real-time list of current cluster members. As will be shown later, such a primitive operation is the basis of most coordination-requiring tasks since, in order to partition the task, a holistic view of the network is required in most cases. Given that capability, we develop a second state reducer tasked with timeframe division among the nodes, allowing us to mitigate risks associated with potential violation of security policies imposed by external services. Lastly, we develop a rate-limit state reducer responsible for maintaining a synchronized configuration of remote security policy. Since that configuration could be dynamic, we describe the communication channels and methods that could be used to keep the cluster state in a consistent manner.

Analysis of literary sources. Consensus and coordination have been actively studied in numerous works throughout the world and are extremely relevant topics especially in context of emerging blockchain technologies (Yaga et al., 2019). Nevertheless, the category of state management methods that do not involve external coordination plane remains an open area requiring additional effort and discussion.

The description and evaluation of centralized coordination methods based on ZooKeeper is done within works of Junqueira, F., & Reed, B. (2013). Similarly, contribution towards utilization of ETCD service and its capabilities are provided within works of Nalawala, H. S. et al. (2022).

Studies of Paxos consensus protocol were conducted within the works of Van Renesse, R., & Altinbuken, D. (2015). Respectively, research on Raft consensus protocol and its properties is described within papers of Hu, J., & Liu, K. (2020); Ni, L. et al. (2024).

Research on DDOS and its variations has been going on since the early 2000s. Significant contributions within the scope of taxonomy, detection and prevention methods are provided by the works of Mirkovic, J., & Reiher, P. (2004); Douligeris, C., & Mitrokotsa, A. (2004). Farther improvement of protection models and methods are discussed in Srivastava, A., et al. (2011). More recent papers involving descriptions of using artificial intelligence capabilities to detect DDOS attacks are done within works of Zhang, B., Zhang, T., & Yu, Z. (2017).

The Replica State Discovery Protocol, its implementation details, terminology and phases are described within the works of Toliupa, S. et al. (2024). The media access layer based on AMQP, its architecture, properties, and characteristics are discussed in Kotov, M. S. et al. (2024).

Methods

Within the ambit of this research, a set of cluster state reducers is described, providing basic coordination capabilities, including the formation of a network participants list and task division between them. Using mathematical modeling, we describe the procedures necessary for performing the said coordination tasks. Implementation and testing of reducers are done with the Node.js platform capable of running JavaScript code on the server side. A theoretical analysis and description of the proposed methods for distributed coordination are

ISSN 2707-1758

provided within this work to facilitate their integration into modern systems.

Additionally, this paper provides a description of coordination-requiring tasks related to the rate-limited external services. The architectural model of the potential clustered data-gathering solution is laid out and evaluated using diagrams and mathematical notations. As we propose new coordination methods based on RSDP's reducer interface, we've described their potential integration within the scope of such systems.

Results

The following section presents a description of the proposed state reducers aimed at facilitating cluster coordination. We will first start with the description of the cluster requirements and the general architecture that could be encountered in cases where there is a combination of external service, temporary availability of data, and rate-limiting security policies in place. We will define the general approach towards request limitation commonly done by external service providers. In addition, a diagram of architecture will be used to further clarify the setup.

Architecture of a system with external limited dependencies. Let us start the discussion of results with the definition of the target distributed system. Before proceeding with the description of cluster state reducers, we will first define the network topology and its constraints. We will shortly discuss external security policies and their implications while building data-gathering or synchronization services for temporarily available aggregates.

The following Fig. 1 depicts the architecture of such a system:

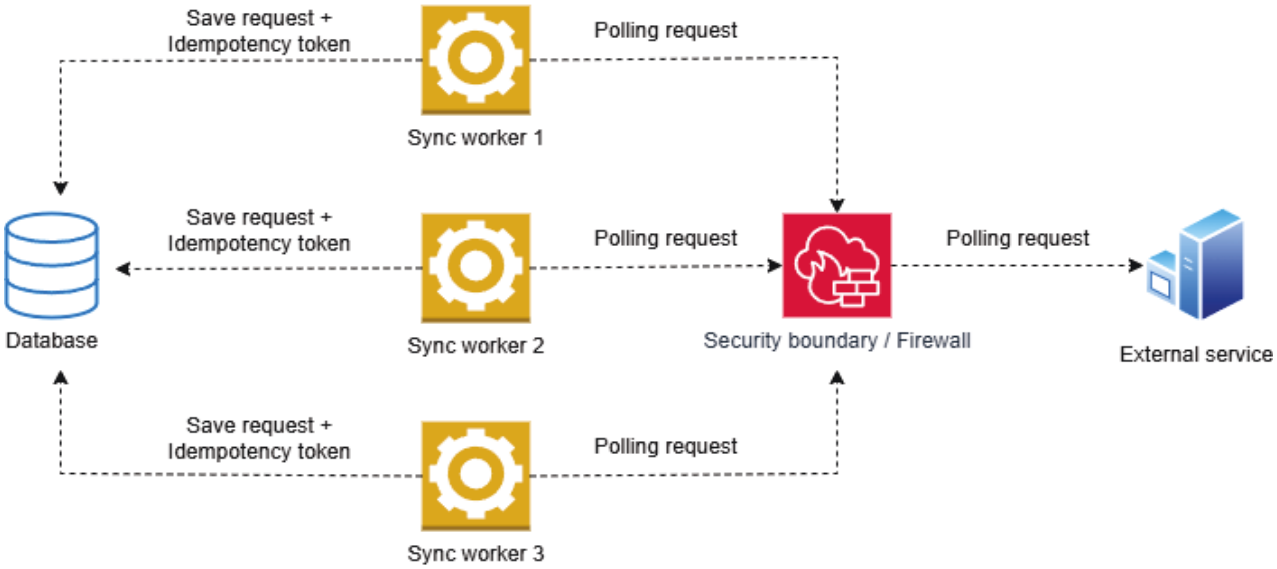


Fig. 1. Architecture of the target distributed system

This system is comprised of multiple synchronization workers that form an RSDP cluster. Their main task is to continuously poll data from an external service provider. The conditions that make this task complex are as follows:

- The external service provides data aggregations temporarily, meaning that historical reconciliation is impossible since older records will not be available in the future.
- At the same time, the service has a security policy that restricts incoming requests with a shared identifier within a time window.
- Controlled service is supposed to minimize data loss through any available means.

Hence, the architecture of a controlled network is comprised of multiple duplicate polling processes allocated on different machines or even availability zones. At every collection period, all three must try to retrieve the data from the target service. If one of them failed, the other would try to save the records received.

Overall, the idea is to leverage redundancy to mitigate potential risks associated with data loss. This is a typical strategy met within many distributed systems.

In order to avoid data duplication inside the database, the idempotency token could be used to check the availability of the record. In the case of the described system, polling interval start and interval end, along with specific request type parameters, could be used as such tokens. Before inserting a new record in the database, every worker would first look to see if there is already a record stored with the mentioned field. To further strengthen consistency guarantees, a unique constraint policy could be implemented, which is commonly supported in RDBM systems and other databases.

The problem arises due to the potential security policies on the provider's side. Since rate limits block excessive requests to the service, our task is to ensure that all the parallel polling processes work in a coordinated way to avoid violations and possible further restrictions. This problem could be generalized to any type of access or work with a shared resource. When multiple parallel actors try to work on the same problem, coordination is required to avoid inconsistent behavior.

Rate-limiting policies come in multiple variations but could be generalized as an access sliding window. In that case, the external service would define a number of requests allowed within a specific timeframe. If that amount is exceeded, the sender might be subjected to additional restrictions or even a permanent ban from accessing the API (Serbout et al., 2023; Firmani, Leotta, & Mecella, 2019).

To describe how such policies work, let us declare the following:

- $R \in \mathbb{R}^+$: maximum request rate (requests per second);
- $T \in \mathbb{R}^+$: time window (in seconds);

ISSN 2707-1758

- $N(t)$: number of requests made in interval $[t - T, t]$;
- $u(t) \in \{0,1\}$: unit impulse indicating a request at time t.
- Now the constraint, or the rate limit condition could be expressed as:

$$\int_{t-T}^{t} u(\tau)\, d\tau \leq R \cdot T, \quad \forall t \in \mathbb{R}^{+}. \tag{1}$$

Equivalently, in discrete time (e.g., if requests occur at discrete timestamps t_i):

Let $\{t_i\}_{i=1}^{n} \subset \mathbb{R}^{+},\ t_i < t_{i+1}$. Now define the windowed count:

$$N(t) = \sum_{i=1}^{n} \mathbb{1}_{[t-T,t]}(t_i). \tag{2}$$

The rate limit condition could be expressed as:

$$N(t) \leq R \cdot T. \tag{3}$$

The acceptance function $a(t_i) \in \{0,1\}$ could be defined as:

$$a(t_i) = \begin{cases} 1 \ \ if\ N(t_i^-) < R \cdot T, \\ 0 \ \ if\ otherwise. \end{cases} \tag{4}$$

That being said, we can proceed with the definition of coordination mechanisms capable of working with such external constraints to avoid potential security incidents.

The state reducer interface notation. Before proceeding with descriptions of the proposed state reducers, we will first have to establish a mathematical notation basis that will be used throughout these definitions.

Firstly, the cluster itself could be defined as a graph $G_{\text{cluster}} = (V, E)$, with a set of replicas $V = \{v_1, v_2, \dots, v_n\}$ and a set of directed edges $E \subseteq V \times V$. In this network each node has its own initial state s_i used to later derive an aggregated state s_i^* along with incoming initial states from in-neighbors $\mathcal{N}^-(v_i)$. The process could be defined as follows (Toliupa et al., 2024):

$$s_i^* = f_{agg\ status}\left(s_i, \{M_{status}(v_j) \mid v_j \in \mathcal{N}^-(v_i)\}\right) \tag{5}$$

Here, $M_{status}(v_j)$ represents an incoming message from in-neighbor in response to initial $M_{hello}(v_i)$. This message could be then defined as $M_{status}(v_j) = \left(f_{\text{id}}(v_j), f_{meta}(v_i), s_j\right)$. We've additionally defined utility functions $f_{id}(v_j)$ that returns a routable identifier for v_j and $f_{\text{meta}}(v_i)$ that returns meta information about participating node (Toliupa et al., 2024).

Later, we will use $s_i^* = f_{agg\ status}$ notation to describe every subsequent state reducer aggregation function. Having said that, let us proceed with the state reducers definition.

Cluster members state reducer. This reducer is one of the most commonly used to resolve any coordination task. That becomes apparent since, in order to distribute tasks between participants, their complete list is required along with the metadata, such as their capacity and capabilities.

The reducer could be described as follows:

$$s_i^* = \left\{ f_{id}(v_j), f_{meta}(v_i), s_j \mid v_j \in \mathcal{N}^-(v_i) \cup v_i \right\}. \tag{6}$$

Its implementation is quite simple but equips the network with an important ability to discover peers.

Timeframe division state reducer. Now, when we can get a list of network participants, we can create an additional state reducer that is capable of dividing timeframes among nodes. Its implementation could be farther augmented by introducing sharding categories, in case there are multiple API keys, but the general approach could be defined as:

$$s_i^* = \left\{ \left(f_{id}(v_j), \frac{\Delta T}{|\mathcal{A}|} \cdot rank\left(f_{id}(v_j), \mathcal{A}\right) \right) \,\middle|\, v_j \in \mathcal{N}^-(v_i) \cup v_i \right\}, \tag{7}$$

where, $\mathcal{A} = \{f_{id}(v_j) \mid v_j \in \mathcal{N}^-(v_i) \cup v_i\}$, $|\mathcal{A}|$ is the amount of node addresses registered within $\mathcal{A}$. Then $rank\left(f_{id}(v_j), \mathcal{A}\right)$ is a function that returns the position of $f_{id}(v_j)$ within a sorted set $\mathcal{A}$ for a given node v_j.

Also, ΔT is a synchronization period that could be obtained either as a static configuration shared in the cluster, or dynamically calculated based on rate limit as follows:

$$\Delta T = |\mathcal{A}| \cdot \frac{1}{R}, \tag{8}$$

where inverse of R basically represents how many seconds a process should wait before it can issue another request in compliance with the restriction policies.

This reducer could be farther improved to support multiple parallel limitation scopes. Let's say that within the observed architecture there are multiple access keys, each having its own dedicated rate limit policy and counter. In that case, it would be more efficient to separate coordination tasks in the following manner:

$$s_i^* = \left\{ \left(f_{id}(v_j), \frac{\Delta T}{|\mathcal{A}(f_{group}(v_j))|} \cdot rank\left(f_{id}(v_j), \mathcal{A}(f_{group}(v_j))\right) \right) \,\middle|\, v_j \in \mathcal{N}^-(v_i) \cup v_i \right\}, \tag{9}$$

where $\mathcal{A}(L_k)$ is now a function that returns a subset of nodes belonging to a particular subset of $\mathcal{A}$ where each member is associated with a label L_k. Here, $f_{group}(v_j)$ would return L_k associated with v_j.

With that setup it is now possible to distribute tasks among nodes in several groups. Each group would have its own isolated access context and would still be capable of coordinating separately.

Rate limit state reducer. It is quite common for the R value to be dynamic itself. These policies tend to get updated over time due to the demand and capability factors on the external service's side. Hence, one of the key state reducers is responsible for maintaining this value in a consistent manner:

$$s_i^* = \left\{ f_{rl}(v_j) \mid v_j \in \mathcal{N}^-(v_i) \cup v_i \right\}, \tag{10}$$

where $f_{\text{rl}}(v_j)$ returns rate limit value observed at node v_j.

The RSDP has undergone multiple iterations, some of which perform popular vote decisions on aggregated states. That implies that propagation of the rate limit value should be done through side channels as shown in the following Fig. 2.

Since a new value in such RSDP version would only be accepted if most of nodes share it, without side channel such propagation would be slow. Another approach is to pass timestamp t along with $f_{rl}(v_j)$ and make the state derivation function find such $f_{rl}(v_j)$ that is associated with the largest value t.

ISSN 2707-1758

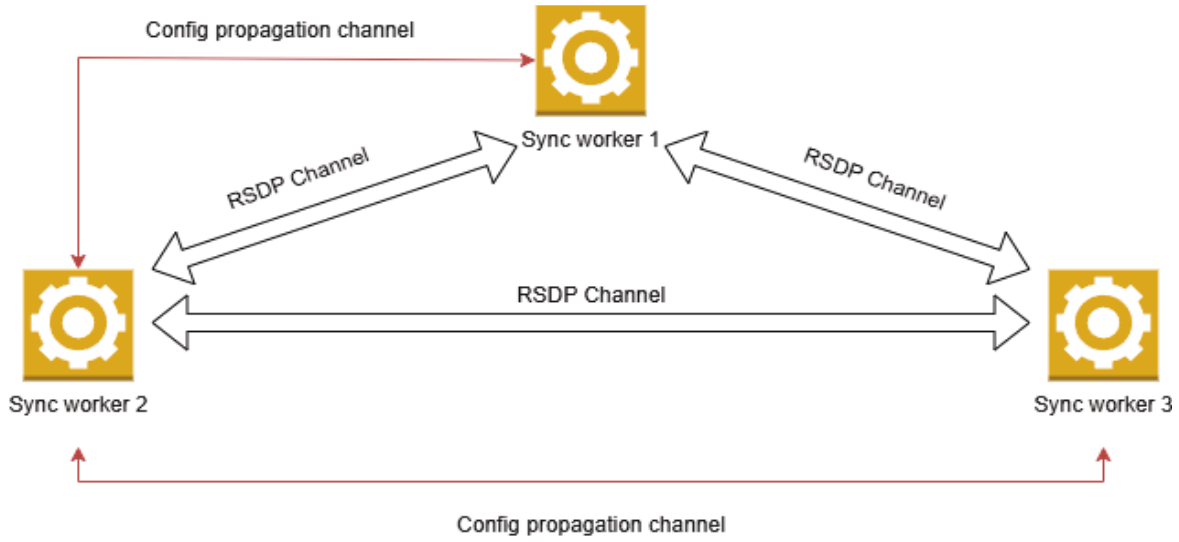


Fig. 2. RSDP initial state updates propagation

Discussion and conclusions

While engineering a complex distributed system requiring consensus among its subsystems or a set of homogenous components, it's important to avoid complexities related to the management of additional infrastructure while still providing a required level of consistency and availability. Having said that, the Replica State Discovery Protocol provides essential lightweight capabilities to resolve the said problem through the means of its flexible state reducers system. RSDP is built with a layered architecture in mind, capable of adjusting to the particular needs of the network as shown in this paper. By leveraging existing communication infrastructure and avoiding redundant management layers, RSDP allows for significantly reducing the computational complexity of coordination as well as costs associated with hardware needed for running a dedicated control plane. State reducers described within this article provide basic capabilities required for the most common coordination tasks, including the construction of a participants directory, task splitting and assignments, as well as consensus regarding the configuration parameters.

Within this paper, we've covered a common architectural problem we met while integrating with myriads of external services. Since most SaaS platforms encounter spamming and DoS problems, most utilize rate-limiting policies for each connecting client. While mitigating availability risks on the server's side, it significantly complicates data redundancy techniques and distributed polling for clients, which require coordination capabilities.

As such, the proposed state reducers provide all the necessary information for running polling processes to both retrieve the data and comply with external security restrictions. The demonstrated reducers allow you to gain a complete network view, starting with the participants list. The list serves as basic information for more complex state division among nodes. By knowing how many nodes there are in the network and being able to dynamically adjust the list according to their health or availability, it is possible to construct more complex coordination methods. Hence, the second state reducer, based on the list, divides the available timeframes among the participants, allowing them to avoid collisions and accidental bans from the target API. At the same time, the third reducer provides a real-time configuration for the available rate limit policy.

Overall, the purpose of this article is to show examples of RSDP's potential application in solving coordination-related tasks. We've laid out the direction of the state management capabilities available through RSDP interfaces and their versatility. It is the intent of this paper to spark further research into the application of RSDP's capabilities in solving non-trivial tasks in contemporary parallelized systems requiring access to a shared resource.

Author's contribution: Maksym Kotov – conceptualization, methodology, formal analysis, development of software, analysis of sources, preparation of a literature review and theoretical foundations of research, editing and reviewing.

Sources of funding. This study did not receive any grant from a funding institution in the public, commercial, or non-commercial sectors.

References

Ali, A. H. (2019). A Survey on Vertical and Horizontal Scaling Platforms for Big Data Analytics. *International Journal of Integrated Engineering*, 11(6), 138–150. https://publisher.uthm.edu.my/ojs/index.php/ijie/article/view/2892

Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Computer Networks*, 44(5), 643–666. https://doi.org/10.1016/j.comnet.2003.10.003

El Malki, A., Zdun, U., & Pautasso, C. (2022). Impact of API rate limit on reliability of microservices-based architectures. In B. Smith, & J. Doe (Eds.), *Proceedings of the 2022 IEEE International Conference on Service-Oriented System Engineering (SOSE)* (pp. 19–28). IEEE Press. https://doi.org/10.1109/SOSE55356.2022.00009

Firmani, D., Leotta, F., & Mecella, M. (2019). On computing throttling rate limits in web APIs through statistical inference. In J. Zhang & H. Zhao (Eds.), *Proceedings of the 2019 IEEE International Conference on Web Services (ICWS)* (pp. 418–425). IEEE Press. https://doi.org/10.1109/ICWS.2019.00075

Hu, J., & Liu, K. (2020). Raft consensus mechanism and the applications. *Journal of Physics: Conference Series*, 1544(1), 012079. https://doi.org/10.1088/1742-6596/1544/1/012079

Junqueira, F., & Reed, B. (2013). *ZooKeeper: Distributed process coordination.* O'Reilly Media, Inc. https://www.oreilly.com/library/view/zookeeper/9781449361297/

Kotov, M. S., Toliupa, S. V., & Nakonechnyi, V. S. (2024). Method of building local area network simulation based on AMQP and its support protocols suite. *Telecommunication and Information Technologies*, 3, 102–119 [in Ukrainian]. [Котов, М. С., Толюпа С. В., Наконечний В. С. (2024). Метод побудови симуляції локальної мережі на основі amqp та набір протоколів підтримки. *Телекомунікації та інформаційні технології,* 3, 102–119]. https://doi.org/10.31673/2412-4338.2024.039989

Millnert, V., & Eker, J. (2020). HoloScale: Horizontal and vertical scaling of cloud resources. In A. Editor, & B. Editor (Eds.), *Proceedings of the 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)* (pp. 196–205). IEEE Press. https://doi.org/10.1109/UCC48980.2020.00038

ISSN 2707-1758

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *SIGCOMM Computer Communication Review*, 34(2), 39–53. https://doi.org/10.1145/997150.997156

Nalawala, H. S., Shah, J., Agrawal, S., & Oza, P. (2022). A comprehensive study of "etcd"– An open-source distributed key-value store with relevant distributed databases. In Emerging Technologies for Computing, *Communication and Smart Cities. Lecture Notes in Electrical Engineering*, 875. Springer, Singapore. https://doi.org/10.1007/978-981-19-0284-0_35

Ni, L., Ye, Q., Yang, J., Zhang, S., & Xian, M. (2024). Research of Raft algorithm improvement in blockchain. In X. Editor, & Y. Editor (Eds.), *Proceedings of the 2024 IEEE 16th International Conference on Advanced Infocomm Technology (ICAIT)* (pp. 290–294). IEEE Press. https://doi.org/10.1109/ICAIT62580.2024.10808140

Serbout, S., El Malki, A., Pautasso, C., & Zdun, U. (2023). API rate limit adoption: A pattern collection. In J. Noble, & K. Beck (Eds.), *Proceedings of the 28th European Conference on Pattern Languages of Programs (EuroPLoP '23)* (Article 5, pp. 1–20). Association for Computing Machinery. https://doi.org/10.1145/3628034.3628039

Srivastava, A., Gupta, B. B., Tyagi, A., Sharma, A., & Mishra, A. (2011). A recent survey on DDoS attacks and defense mechanisms. In Advances in parallel distributed computing. PDCTA 2011. *Communications in computer and information science*, 203. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24037-9_57

Toliupa, S., Kotov, M., Buchyk, S., Boiko, J., & Shtanenko, S. (2024). Stateful cluster leader failover models and methods based on Replica State Discovery Protocol. *Proceedings of the IT&I Workshops 2024*, 120–140. https://ceur-ws.org/Vol-3933/Paper_10.pdf

Van Renesse, R., & Altinbuken, D. (2015). Paxos made moderately complex. ACM Computing Surveys, 47(3), Article 42, 1–36. https://doi.org/10.1145/2673577

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2019). *Blockchain technology overview.* National Institute of Standards and Technology Internal Report. https://doi.org/10.48550/arXiv.1906.11078

Zhang, B., Zhang, T., & Yu, Z. (2017). DDoS detection and prevention based on artificial intelligence techniques. In M. Chen, & L. Wang (Eds.), *Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC)* (pp. 1276–1280). IEEE Press. https://doi.org/10.1109/CompComm.2017.8322748

Отримано редакцією журналу / Received: 17.05.25
Прорецензовано / Revised: 27.05.25
Схвалено до друку / Accepted: 13.06.25

Максим КОТОВ, асп.
ORCID ID: 0000-0003-1153-3198
e-mail: maksym_kotov@ukr.net
Київський національний університет імені Тараса Шевченка, Київ, Україна

РЕДУКТОРИ СТАНУ ДЛЯ КООРДИНАЦІЇ КЛАСТЕРА НА ОСНОВІ RSDP

В с т у п . ***Сучасні розподілені системи, зазвичай, використовують складні мережні топології та технології внутрішнього зв'язку для виконання складних обчислювальних завдань. Досить часто такі системи покладаються на централізовану координацію, коли один або група призначених серверів є площиною управління для всієї мережі. Хоча цей підхід може спростити процес досягнення консенсусу, він вводить додаткові вимоги до проєктування та обслуговування інфраструктури. Вузли, які є площиною управління, вимагають додаткових апаратних ресурсів, а також людських зусиль і компетенції для управління зовнішніми службами, здатними надавати базові примітиви координації. Більшість випадків, що вимагають консенсусу, можна звести до спрощених процедур, таких як збір інформації про учасників мережі та розподіл детермінованих зрізів стану між ними. Отже, впровадження складного координаційного рішення, яке вимагає додаткового методу обслуговування, може бути неефективним. Протокол виявлення стану репліки (RSDP) виступає як "lightweight" рішення для координації, що представляє простий інтерфейс для досягнення консенсусу між вузлами у кластері.***

М е т о д и . ***У межах цього дослідження описано набір редукторів стану кластера, що забезпечує базові можливості координації, включаючи формування списку учасників мережі та розподіл завдань між ними. За допомогою математичного моделювання описано процедури, необхідні для виконання зазначених координаційних завдань. Впровадження та тестування редукторів виконується за допомогою платформи Node.js, здатної запускати код JavaScript на боці сервера. Теоретичний аналіз і опис пропонованих методів розподіленої координації представлено в цій роботі для полегшення їхньої інтеграції в сучасні системи.***

Р е з у л ь т а т и . ***У результаті цього дослідження ми пропонуємо три нових редуктори стану кластера, які являють собою методи базових можливостей координації як зразкове застосування RSDP. Перший редуктор відповідає за збір каталогу вузлів-учасників у кластері та підтримку їхніх статусів на основі отриманих даних. Другий редуктор виконує розподіл часових термінів у межах кластера між вузлами для координації їхнього виконання у взаємовиключному середовищі. Нарешті, редуктор обмеження швидкості описує логіку для досягнення консенсусу щодо єдиного значення, яке має бути спільним для всієї системи, а також негайно оновлюватися за потреби.***

В и с н о в к и . ***Під час проєктування складної розподіленої системи, що потребує консенсусу між її підсистемами або набором однорідних компонентів, важливо уникати складнощів, пов'язаних з управлінням додатковою інфраструктурою, забезпечуючи водночас необхідний рівень узгодженості та доступності. Зважаючи на це, протокол виявлення стану репліки надає важливі полегшені можливості для розв'язання зазначеної проблеми за допомогою своєї гнучкої системи редукторів стану. RSDP створено з урахуванням багаторівневої архітектури, здатної адаптуватися до конкретних потреб мережі, як показано в цій статті. Використовуючи наявну комунікаційну інфраструктуру й уникаючи надлишкових рівнів управління, RSDP дозволяє значно зменшити обчислювальну складність координації, а також витрати, пов'язані з апаратним забезпеченням, необхідним для роботи спеціальної площини управління. Редуктори стану, описані в цій статті, надають базові можливості, необхідні для найпоширеніших завдань координації, включаючи створення каталогу учасників, розподіл завдань і призначення, а також консенсус щодо параметрів конфігурації.***

К л ю ч о в і с л о в а : ***розподілені обчислення; узгодження пристрою; синхронізація стану; управління кластерами; доступність сервісів; інциденти безпеки; протокол виявлення стану репліки (RSDP); редуктори стану кластера RSDP.***

Автор заявляє про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

УДК 004.056:004.8:004.774
DOI: https://doi.org/10.17721/ISTS.2025.9.11-17

Сергій БУЧИК, д-р техн. наук, проф.
ORCID ID: 0000-0003-0892-3494
e-mail: buchyk@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Віталій П'ЯТИГОР, асп.
ORCID ID: 0000-0002-7621-1299
e-mail: vp5gor@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

АРХІТЕКТУРА СИСТЕМ ВИЯВЛЕННЯ АВТОМАТИЗОВАНИХ АКАУНТІВ (БОТІВ) У СОЦІАЛЬНИХ МЕРЕЖАХ

Вступ. *Соціальні мережі є одним з основних джерел отримання інформації. Проте зі зростанням довіри до них зростає і використання автоматизованих акаунтів для поширення дезінформації та впливу на суспільну думку, що є ефективним методом маніпуляції. Для виявлення таких акаунтів застосовують системи виявлення автоматизованих акаунтів. Метою статті є розгляд архітектури систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах і визначення основних напрямів щодо вдосконалення такої архітектури з урахуванням виявлених недоліків.*

Методи. *Використано методи аналізу, систематизації та узагальнення для визначення недоліків і напрямів удосконалення систем і метод моделювання для розроблення загальної та вдосконаленої архітектури систем.*

Результати. *Запропоновано вдосконалену архітектуру системи виявлення автоматизованих акаунтів (ботів) у соціальній мережі x.com, що використовує змішану модель аналізу даних за допомогою штучного інтелекту й об'єднує вебскрейпінг та API запити для збору даних.*

Висновки. *Розглянуто загальну архітектуру систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах і визначено основні недоліки та напрями вдосконалення, серед яких можна виділити обмеження запитів до API від власників соціальних мереж, неструктурований і вільний стиль дописів у соціальних мережах і постійну зміну алгоритмів генерації дописів автоматизованими акаунтами. Запропоновано вдосконалену архітектуру системи виявлення автоматизованих акаунтів ботів, що поєднує вебскрейпінг та API запити для збору даних і дозволяє паралельне виконання аналізу даних соціальних мереж.*

Ключові слова: *комп'ютерна система, збір даних, аналіз даних, машинне навчання.*

Вступ

Соціальні мережі стали одним з основних джерел інформації, забезпечуючи майже миттєву доставку та часто анонімне використання. Вони стали основним компонентом засобів комунікації для окремих осіб та організацій. Наприклад, у квітні 2024 р. мережа "FacebookC налічувала 3 млрд користувачів, які були активними хоча б раз на місяць (Dixon, 2024).

Зі зростанням впливу соціальних мереж, вони стали зброєю, доступною для будь-кого. Хоча більшість користувачів використовують ці платформи для побутових цілей, існують такі особи й організації, які використовують соціальні мережі для прихованих цілей. Щоб використовувати силу соціальних мереж, окремим особам або організаціям необхідно вплинути на їхні алгоритми трендів і взаємодії з дописами, тому створюють автоматизовані аккаунти або боти. Боти соціальних мереж – це комп'ютерні алгоритми, які створюють контент і взаємодіють із користувачами (Ferrara et al., 2016). Згідно зі статистичними даними, в останньому кварталі 2023 р. з мережі "Facebook" було видалено 691 млн фейкових облікових записів (Dixon, 2025), тоді як за оцінками у 2020 р. у мережі "X" (раніше "Twitter") було 48 млн облікових записів ботів, що становило 15 % усіх облікових записів (Guy et al., 2012). Ця присутність розмиває цілісність інформації, яка поширюється на цих платформах і сприяє створенню середовища, в якому користувачі відчувають невпевненість щодо автентичності інших облікових акаунтів і вмісту.

Одним із способів використання таких мереж ботів є кампанії дезінформації. Вони стали інструментом, який дозволяє державним і недержавним акторам поширювати неправдиві наративи, маніпулювати громадською думкою та дестабілізувати спільноти. Ці боти виглядають як справжні користувачі, що ускладнює їх виявлення та дозволяє їм працювати в такому масштабі, який був би неможливий для самих людей. Такі мережі у великих обсягах задокументовано під час президентських виборів у США 2016 р. (Bessi, & Ferrara, 2016), і ці обсяги зросли після повномасштабного вторгнення росії в Україну (Fredhaim, & Stolze, 2022).

Системи виявлення автоматизованих акаунтів дозволяють виявляти та приймати рішення щодо шкідливої активності, яка здійснюється ботами у соціальних мережах. Ефективне виявлення ботів є критично важливим для забезпечення безпеки користувачів, збереження достовірності інформаційного простору та підтримки довіри до цифрових платформ. Деякі дослідники працюють над класифікацією автоматично створених дописів, а не над виявленням автоматизованих акаунтів (Almerekhi, & Elsayed, 2015). Інші зосереджені на прогнозуванні потенційних цілей атак як підходи захисту (Boshmaf et al., 2016). Але найпопулярнішим механізмом захисту є виявлення облікових акаунтів ботів, індивідуальних чи групових (ботнетів), на ранніх стадіях (створення облікових записів) або після початку їхньої роботи в соціальних мережах.

Метою цієї статті є розгляд архітектури систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах і визначення основних напрямів удосконалення такої архітектури з урахуванням виявлених недоліків.

Об'єктом дослідження є процес виявлення автоматизованих акаунтів (ботів) у соціальних мережах.

Предметом дослідження є архітектури систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах.

Отже, основне завдання полягає в удосконаленні архітектури систем виявлення автоматизованих акаунтів (ботів) у соціальних мережах.

ISSN 2707-1758

Методи

В роботі використано такі методи дослідження.

1. Метод аналізу застосовано для детального розгляду існуючих систем виявлення автоматизованих акаунтів у соціальних мережах. Він дозволив виявити їхні складові частини, принципи роботи та потенційні недоліки.

2. Метод систематизації використано для впорядкування інформації, отриманої в результаті аналізу. Це дозволило класифікувати недоліки існуючих систем і визначити загальні напрями для їхнього вдосконалення.

3. Метод узагальнення, застосування якого дало змогу сформулювати загальні висновки щодо проблем і перспектив розвитку систем виявлення ботів, а також запропонувати ширші напрями для вдосконалення архітектури.

4. Метод моделювання використано для розроблення як загальної архітектури систем виявлення ботів, так і запропонованої удосконаленої архітектури для конкретної соціальної мережі (x.com). Це дозволило візуалізувати запропоновані зміни та їхній потенційний вплив. Отже, основними методами дослідження, використаними в цій роботі, є аналіз, систематизація, узагальнення та моделювання. Ці методи дозволили авторам дослідити існуючий стан проблеми, виявити недоліки та запропонувати власне рішення у вигляді вдосконаленої архітектури системи виявлення автоматизованих акаунтів.

Огляд літератури. У попередніх роботах авторів розглянуто основні етапи систем виявлення автоматизованих акаунтів і проблем, які необхідно розв'язати для розроблення таких систем (П'ятигор, & Бучик, 2025). Було виділено такі етапи: збір даних, попереднє оброблення й аналіз даних. З урахуванням даних, представлених у роботі, можемо подати загальну архітектури систем виявлення автоматизованих акаунтів, що зображена на рис. 1. Додатково виділено етап представлення результатів і компонент бази даних. Успіх таких систем залежить від інтеграції сучасних технологій, врахування нових методів обходу захисту та постійного оновлення моделей, що використовуються для аналізу.

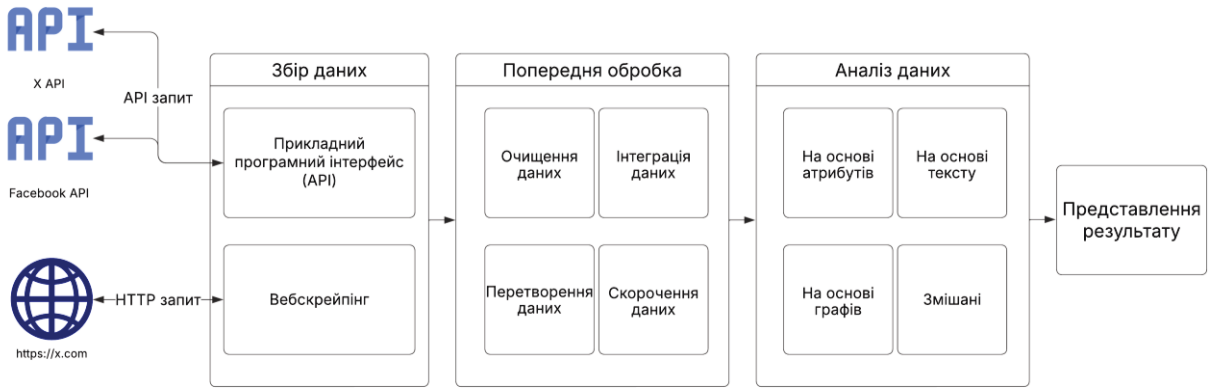


Рис. 1. Загальна архітектура систем виявлення автоматизованих акаунтів у соціальних мережах

На етапі аналізу даних окремим підетапом можна виділити компонент навчання моделі, оскільки більшість із методів використовують засоби машинного навчання для аналізу даних. Цей компонент безсумнівно потрібен у випадку, коли модель не використовувалася до цього і не була навчена, або модель є еволюційною, тобто такою, що постійно навчається. Навчальна модель – це набір даних, який використовується для навчання алгоритму машинного навчання. Він складається з вибірки вихідних даних і відповідних наборів вхідних даних, які впливають на вихід. Навчальну модель застосовують для проходження вхідних даних через алгоритм, щоб співвіднести оброблений вихід із зразком. Результат цієї кореляції використовують для модифікації моделі. Цей ітеративний процес називають "допасовування моделі". Навчання моделі машинною мовою – це процес подачі даних в алгоритм, щоб допомогти визначити та вивчити впливові значення для всіх задіяних атрибутів. Існує кілька типів моделей машинного навчання, з яких найпоширенішими є контрольоване та неконтрольоване навчання.

Контрольоване навчання можливе, коли навчальні дані містять і вхідні, і вихідні значення. Кожен набір даних, який має вхідні й очікувані вихідні дані, називають контрольним сигналом. Навчання виконується на основі відхилення обробленого результату від задокументованого результату, коли вхідні дані вводять у модель.

Навчання без нагляду передбачає визначення закономірностей у даних. Потім додаткові дані використовують для підгонки шаблонів або кластерів. Це також ітераційний процес, який покращує точність на основі кореляції з очікуваними шаблонами або кластерами. У цьому методі немає еталонного вихідного набору даних.

За можливості, в системах варто використовувати алгоритми самостійного навчання або навчання без нагляду для підвищення ефективності визначення автоматизованих акаунтів, проте треба пам'ятати, що навчання без нагляду залежить від якості даних, які надаються для навчання. Це означає, що все більшу увагу варто привертати до етапу попереднього оброблення даних, які потім будуть передаватися на навчання й аналіз через велику неузгодженість даних у соціальних мережах.

Іншим опціональним компонентом є база даних. Хоча всі дані, які обробляються на будь-якому з етапів, можна зберігати в оперативній пам'яті, більш практичним підходом є використання бази даних для збереження проміжних результатів. Як мінімум, варто зберігати остаточні результати аналізу даних у базах даних, що дозволить збирати статистичні дані отриманих результатів. Основними моделями баз даних

ISSN 2707-1758

для задач зберігання великої кількості даних є реляційна і NoSQL. Реляційна модель є надійнішою з погляду цілісності даних, але дані, які зберігаються, мають бути представлені в нормальних формах і мати чітко визначений формат. Така база даних може бути використана для зберігання фінальних результатів у форматі – опис даних і результат аналізу, проте для зберігання даних із попередніх етапів, така база даних не є достатньо гнучкою і оптимізація запитів є доволі складною задачею.

NoSQL бази даних або документо-орієнтовані бази даних не мають вимоги до структуризації даних у нормальних формах і в більшому представленні у форматі ключ-значення. Такі бази даних є дуже гнучкими до формату та розміру даних і є відкритими до доповнення блоку даних новою парою – ключ-значення. Це означає, що на кожному етапі оброблення ми можемо додавати результат оброблення як ключ-значення до існуючого запису. Оскільки документи найчастіше мають формат JSON, їх набагато легше читати та розуміти користувачеві. Документи, які зберігаються, також є цілим записом, тобто немає необхідності посилатися на різні таблиці для відображення результату, що є проблемою в реляційних базах. Бази даних документів також мають високу масштабованість. На відміну від реляційних баз даних, де традиційно можна масштабувати лише вертикально, нереляційні бази даних, зокрема і бази даних документів, можна масштабувати горизонтально. Це означає, що бази даних дублюються на кількох серверах, але залишаються синхронізованими.

Останнім етапом оповіщення системи є представлення результату користувачеві. Це може бути і текст у консолі застосунку про те, що користувач є ботом, або додаток, який отримує дані та представляє результат аналізу, впевненість системи в результаті, дані щодо того, на основі яких параметрів було прийнято рішення, тощо. Використання баз даних дають можливість застосунку відображати не тільки останній результат аналізу, а й історичні дані протягом усього часу роботи системи. Також результати можуть бути представлені у таких засобах візуалізації як PowerBI або Google Charts. Це дозволяє генерувати інфографіки та звіти, які є компактним та інформативним способом представити результати роботи системи без необхідності, щоб кінцевий користувач розумів весь обсяг даних.

Результати

З огляду на недоліки, описані вище, можна виділити можливі напрями для вдосконалення таких систем. Деякі з таких напрямів удосконалення авторами представлено у табл. 1, які поділені поетапно.

Таблиця 1

Напрями вдосконалення компонентів системи виявлення автоматизованих акаунтів

Етап	Напрям удосконалення
Збір даних	Оптимізація запитів до API
	Підвищення функціоналу відкритих до змін API та їхньої документації
	Постійна адаптація методів вебскрейпінгу до оновлень дизайну вебсторінок
	Законодавче врегулювання засобів вебскрейпінгу в Україні
Попереднє оброблення	Адаптація методів попереднього оброблення до даних соціальних мереж
Аналіз даних	Постійна адаптація методів до замаскованих даних
	Використання засобів попереднього оброблення для вилучення замаскованих даних
	Засосування хмарних технологій
	Виконання задач аналізу розподіленими системами
	Зменшення обсягу даних або збільшення часу оброблення

Як згадувалось у роботі (П'ятигор, & Бучик, 2025), одним із недоліків збору даних за допомогою API є обмеження на кількість, частоту запитів або обсяг даних, які повертаються в результаті виконання запиту. Тому системи збору мають бути адаптовані до вимог API. Оптимізація запитів полягає у звуженні обсягу даних, які повертаються API, до лише тих, які необхідні для аналізу, за допомогою використання фільтрів, які надаються власниками API. Це вимагає детального розуміння інтерфейсу, який надає розробник, а також зв'язків між даними. Крім того варто знати, за допомогою яких методів інтерфейсу ці дані можуть бути отримані. Якщо API розміщено у відкритому доступі і не є закритим до редагування, розширення функціоналу API можливе і зможе спростити запити у системах збору даних або додати необхідну інформацію до результату запиту до API, яка до цього вимагала додаткових запитів до API.

Оскільки системи вебскрейпінгу залежать від дизайну вебсторінки, з якої збирається інформація, постійна адаптація до нових версій і змін дизайну є обов'язковим напрямом удосконалення таких систем, якщо система збору є постійною і безперервною. Для забезпечення безперервності, необхідно налагодити доступ до даних без залежності до дизайну вебсторінки, наприклад за допомогою тимчасової зміни методу збору на збір через API, або договорами з власниками соціальних мереж задля отримання попередньої інформації про оновлення сервісу, щоб мати час оновити системи збору.

На відміну від організацій, де як дані, так і ієрархія даних є структурованими, дані із соціальних мереж написано у вільному стилі. З огляду на те, що користувачі соціальних мереж можуть писати все що завгодно, якість даних у таких середовищах коливається від цінних до реклами та непотрібу. Тому засоби попереднього оброблення мають бути адаптовані до специфіки даних, не тільки загальних для всіх соціальних мереж, а також до соціальної мережі, дані якої обробляються в системі. Чим більше джерел даних обробляється, тим складніше уніфікувати та структурувати такі дані.

ISSN 2707-1758

Одним із недоліків етапу аналізу даних, є системні вимоги до апаратного забезпечення для виконання цього аналізу. Залежно від обсягу даних, що обробляються, вимоги можуть змінюватися, але часто аналіз великого обсягу даних вимагає наявності високопродуктивної системи з багатоядерним процесором для розв'язання мультипотокових задач і графічного процесора з великим запасом відеопам'яті та підтримкою високопродуктивних обчислень. Це обмежує можливості розроблення та розширення великих систем на власному апаратному забезпеченні. Можливим удосконаленням забезпечення апаратних вимог до етапу аналізу даних є використання хмарних технологій, якщо дані, які обробляються, дозволяють передачу даних на обробку третім сторонам. Зазвичай провайдери хмарних технологій забезпечують платформу для використання або апаратних ресурсів, або навіть повні сервіси для аналізу даних. Це надає можливість швидкого розширення системи для забезпечення обсягів даних, а також можливості гнучкого регулювання ресурсів, які використовують для аналізу. Така гнучкість також може зменшити початкові витрати на апаратне забезпечення вказаних систем.

На додачу до вертикального розширення систем аналізу даних, хмарні технології можуть забезпечувати і горизонтальне розширення. У розробленні систем аналізу варто враховувати, чи можна поділити процес аналізу на підпроцеси, які можна виконувати паралельно і незалежно. Це дозволяє не тільки зменшити час оброблення даних, але також і навантаження на загальну систему. Зазначимо, що часто горизонтальне розширення є більш швидким і легким процесом порівняно з вертикальним, але розподілення задач на підпроцеси має бути враховане у розробленні таких систем.

Якщо розширення апаратних засобів систем аналізу даних не є можливим, то як оптимизацію систем аналізу можна використати перегляд нефунціональних вимог до систем. Зменшення обсягу даних, що обробляються, або збільшення часу на оброблення даних зможе зменшити системні вимоги до таких систем. Якщо при цьому не змешується точність виявлення акаунтів і система не є критичною, то такі зміни до системи можна вважати вдосконаленням системи.

Визначивши можливі напрями вдосконалення, автори представили на рис. 2 архітектуру системи, яка буде використана для подальших досліджень у сфері виявлення автоматизованих аккаунтів. Ця архітектура має на меті поєднати вебскрейпер з API запитами для збору інформації, а також розподілити етап аналізу на незалежні компоненти, які можуть виконуватися паралельно. Використання хмарних технологій на всіх етапах, а також розподілення етапу аналізу на незалежні підпроцеси, які виконуються паралелльно, дозволить ефективно контролювати ресурси, які використовує система виявлення автоматизованих аккаунтів, і зменшити час на аналіз даних.

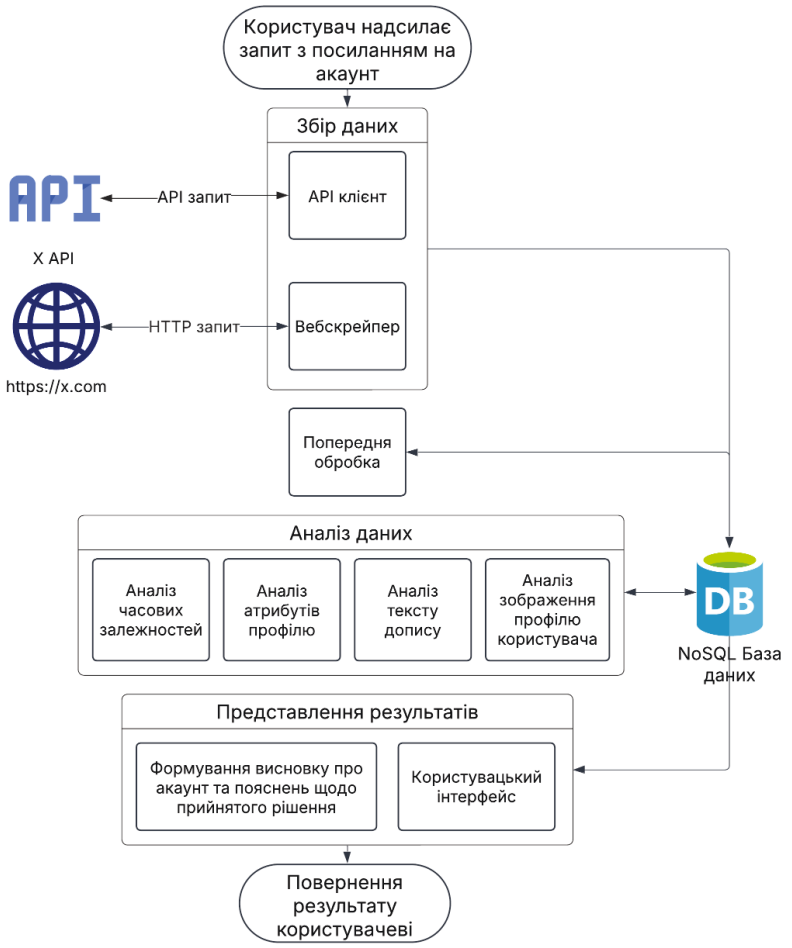


Рис. 2. Запропонована архітектура системи виявлення автоматизованих акаунтів у соціальній мережі "X"

ISSN 2707-1758

Запропонована архітектура системи виявлення автоматизованих акаунтів (ботів) у соціальній мережі "X" складається з кількох основних етапів і компонентів (рис. 2).

На етапі збору даних використовуємо як API клієнт, так і вебскрейпер. Система використовує API соціальної мережі "X" для отримання даних про дописи, користувачів та їхню активність. Користувач надсилає запит із посиланням на акаунт, і API клієнт здійснює API запити для отримання відповідної інформації. На додаток до API, система використовує вебскрейпінг для збору даних безпосередньо з вебсторінок соціальної мережі "X" через HTTP запити. Це необхідно для отримання інформації, яка не надається через публічний API, або для обходу обмежень API на кількість запитів. Вебскрапером також здійснюється очищення даних від непотрібних даних про розмітку та стилі сторінки, скрипти, які присутні на вебсторінці та виділяється корисна інформація. Зібрана інформація зберігається в NoSQL базі даних і контроль передається до етапу попереднього оброблення.

На етапі попереднього оброблення використовують дані, які були зібрані на попередньому етапі і збережені в базі даних. Напряму дані з попереднього етапу не використовують, уся комунікація здійснюється через базу даних. На цьому етапі дані очищають, форматують і зводять до єдиного вигляду для подальшого аналізу. Оброблені дані зберігають в окрему колекцію в базі даних, де кожен запис матиме вигляд профілю користувача з атрибутами, які використовують для аналізу, та списком дописів користувача. Далі контроль передається на етап аналізу даних.

Етап аналізу даних використовує змішаний підхід і включає кілька модулів, які паралельно досліджують різні аспекти поведінки та характеристик користувачів:

- аналіз часових залежностей – досліджується час публікацій, їхня періодичність, інтервали між ними, що може вказувати на автоматизовану діяльність;
- аналіз атрибутів профілю – аналізуються такі характеристики профілю: ім'я користувача, опис, кількість підписників і підписок, дата реєстрації, наявність / відсутність аватара й інша інформація, яка може бути типовою для ботів;
- аналіз тексту допису – зміст, стиль, тематика, наявність певних ключових слів або шаблонних фраз у дописах можуть бути ознаками автоматизованої генерації контенту;
- аналіз зображення профілю користувача – досліджуються характеристики зображення профілю (напр., стандартні зображення, згенеровані штучним інтелектом, або їхня відсутність), що також може бути індикатором бота.

Результат аналізу зберігається у базі даних, запис профілю користувача у базі даних оновлюється з показниками кожного модуля аналізу в числових значеннях. Контроль передається на етап представлення результатів.

Після завершення аналізу даних система формує висновок щодо ймовірності того, що досліджуваний акаунт є автоматизованим, використовуючи отримані значення, що були збережені до бази даних. Формування висновку має бути зваженим, треба враховувати, де певні значення результатів модулів аналізу мають більший чи менший вплив на остаточний результат. Цей висновок також має включати перелік критеріїв, які вказували б на те, що акаунт є автоматизованим, якщо таке рішення було прийняте. Для додаткової прозорості, висновок має також включати відсоток упевненості системи у прийнятому рішенні.

На цьому етапі також присутній компонент користувацького інтерфейсу, який формує відповідь для користувача у форматі вебкомпонента, який буде повернений на сторінку, з якої користувач робив запит.

NoSQL база даних є окремим компонентом системи, який використовують на всіх етапах для збереження проміжних даних, які використовує система. Дані бази даних добре підходять для зберігання неструктурованих або напівструктурованих даних, які часто зустрічаються в соціальних мережах, а також для зберігання великої кількості інформації та оптимізації розподіленого використання даних. База даних має мати дві колекції даних, одна зберігає необроблені дані, які були отримані на етапі збору даних, інша – оброблені записи профілів користувачів, які в ході виконання системи будуть доповнені результатами аналізу.

Як хмарне рішення пропонується використання Microsoft Azure, що має великий спектр сервісів і для побудови системи з використанням лише сервісів даного провайдера, і більш кросплатформні рішення, такі як оркестратор контейнерів Kubernetes. Azure App Service буде використовуватися для розгортання вебінтерфейсу, що отримує початковий запит користувача, відображає результат аналізу та висновки. Як база даних використовуватиметься сервіс Azure Cosmos DB, що є NoSQL базою даних, що має високу швидкість доступу до даних, а також оптимізований для великих обсягів напівструктурованих даних. Azure Functions підходить для реалізації логіки API клієнта та запуску вебскрейпінгових сценаріїв. Хоча правила попереднього оброблення мають бути визначені відповідно до отриманих даних, деякі задачі очищення та зведення даних до єдиного фомату можна покласти на Azure Databricks – платформу для оброблення великих даних, включно з очищенням і зведенням даних до єдиного формату. Оскільки алгоритми аналізу даних будуть власною реалізацію без використання наявних сервісів, для хостингу буде використовуватися Azure Container Instances – сервіс для запуску окремих контейнерів без необхідності в управлінні інфраструктурою, але з підтримкою горизонтального розширення залежно від навантаження. Для загального керування системи буде використовуватися Azure Event Grid – сервіс для маршрутизації подій між компонентами системи, та Azure Logic Apps – платформа автоматизації робочих процесів. Для моніторингу роботи системи, включно зі збором метрик і логів, буде використаний Azure Monitor.

Для розроблення компонентів збору, оброблення, аналізу та формування результатів буде застосована мова програмування Python. Ця мова була вибрана для написання скриптів через наявність великої кількості бібліотек для підтримки аналізу даних, вона забезпечує гнучкість у роботі з різними типами даних і має високу продуктивність завдяки хмарному середовищу. Python добре підходить для оброблення неструктурованих і напівструктурованих даних, які є характерними для соціальних мереж. Скрипти, написані на цій мові, можуть легко обробляти як JSON-дані, отримані через API, так і HTML-дані, зібрані вебскрейпінгом. Вказана мова має великий набір спеціалізованих бібліотек, які роблять аналіз даних ефективним і простим. До основних бібліотек належать такі: Pandas – для роботи з таблицями, оброблення й аналізу даних; NumPy – для обчислень із великими масивами даних. На додачу Python має офіційну підтримку в сервісах Microsoft Azure: Azure SDK для Python для інтеграції з Azure

ISSN 2707-1758

Cosmos DB, Azure Functions, Azure Container Instances та іншими сервісами.

Для розроблення вебсервісу представлення даних буде використаний Flask – мікрофреймворк для веброзробки на мові програмування Python. Він забезпечує простий і гнучкий підхід до створення вебдодатків. Також цей фреймворк повністю підтримується Azure, і його можна використовувати його для створення та розгортання вебдодатків на Azure App Service. Оскільки основною задачею вебзастосунку буде лише отримання запиту користувача та результату з бази даних після виконання аналізу, то цей фреймворк добре підходить для розв'язання цих задач.

Основними характеристиками запропонованої архітектури є:

- гібридний підхід до збору даних – поєднання API запитів і вебскрейпінгу дозволяє отримати повніший обсяг даних, обійшовши можливі обмеження API;
- паралельний аналіз даних – розбиття аналізу на кілька незалежних модулів (часових залежностей, атрибутів профілю, тексту дописів, зображень профілю) дозволяє виконувати їх паралельно, що підвищує ефективність і швидкість роботи системи;
- використання штучного інтелекту: модулі аналізу даних, особливо аналіз тексту та зображень, повинні використовувати методи машинного навчання та штучного інтелекту для виявлення патернів, характерних для ботів.

Представлену систему, її етапи та компоненти можна представити у вигляді множин. Визначимо множину етапів E, яка складається з етапів збору даних, попереднього оброблення, аналізу даних і представлення даних. Множина компонентів K складається з усіх компонентів системи: користувацький інтерфейс, база даних, API клієнт, вебскрапер, засоби попереднього оброблення й обрані засоби аналізу даних. Також етапи системи залежать від результатів попередніх етапів, тому їх також треба включити в модель. Для кожного з етапів визначимо компоненти, які він використовує і побудуємо їх відображення. Цю залежність подано у формулі (1).

$$\begin{aligned} f(\mathrm{E}_1) &= \{\mathrm{K}_1, \mathrm{K}_2, \mathrm{K}_3, \mathrm{K}_4\}, \\ f(\mathrm{E}_2) &= \{\mathrm{K}_2, \mathrm{K}_5, \mathrm{result}(\mathrm{E}_1)\}, \\ f(\mathrm{E}_3) &= \{\mathrm{K}_2, \mathrm{K}_6, \mathrm{K}_7, \mathrm{K}_8, \mathrm{K}_9, \mathrm{result}(\mathrm{E}_2)\}, \\ f(\mathrm{E}_4) &= \{\mathrm{K}_1, \mathrm{K}_2, \mathrm{result}(\mathrm{E}_3)\}, \end{aligned} \quad (1)$$

де E_1 – етап збору даних; E_2 – етап попереднього оброблення; E_3 – етап аналізу даних; E_4 – етап представлення даних; K_1 – компонент користувацького інтерфейсу; K_2 – компонент бази даних; K_3 – компонент API клієнта; K_4 – компонент вебскрейпінгу; K_5 – компонент засобів попереднього оброблення даних; K_6 – компонент засобів аналізу часових залежностей; K_7 – компонент засобів аналізу атрибутів профілю; K_8 – компонент засобів аналізу тексту дописів; K_9 – компонент аналізу зображення профілю користувача; result(E_1) – результат виконання етапу збору даних; result(E_2) – результат виконання етапу попереднього оброблення; result (E_3) – результат виконання етапу аналізу даних.

Отже, загальну систему можна описати як об'єднання всіх етапів, відповідних їм компонентів і результатів виконання попередніх етапів. Це об'єднання представлено у формулі (2).

$$S = \bigcup_{\mathrm{E}_i \in \mathrm{E}} \left(\mathrm{E}_i \times f(\mathrm{E}_i)\right). \quad (2)$$

Таке представлення допомагає чітко поділити етапи системи та компоненти, від яких залежить кожен з етапів. Це робить модель простою для розуміння, аналізу й оброблення, дозволяє оптимізувати структуру системи та передбачати наслідки змін. Описана модель полегшує розширення, модифікацію та автоматизацію.

Дискусія і висновки

За результатами дослідження встановлено, що архітектура систем виявлення ботів включає кілька етапів: збір даних, їхнє попереднє оброблення, аналіз характеристик користувачів і класифікацію облікових записів за допомогою алгоритмів машинного навчання. Додатково в системах можливими компонентами можуть бути засоби навчання моделі, бази даних і засоби представлення даних користувачеві. Доповнено перелік недоліків на кожному з етапів, порівняно з преставленим у попередній роботі авторів (П'ятигор, & Бучик, 2025), та наведено основні напрями вдосконалення.

З урахуванням виявлених недоліків запропоновано вдосконалену архітектуру системи виявлення автоматизованих акаунтів, що поєднує вебскрейпінг та API запити для безперервного збору даних і використовує хмарні технології для гнучкого управління ресурсами, які використовує система, і дозволяє виконувати аналіз даних паралельно. Система також була представлена у вигляді множин, що дозволяє візуалізувати залежності компонентів від етапів, що спрощує планування та розроблення вказаної системи. Представлена система у подальшому буде використана для виконання досліджень у галузі виявлення автоматизованих акаунтів та імплементована для оцінювння її ефективності у реальних умовах.

Внесок авторів: Віталій П'ятигор – концептуалізація, методологія, написання (оригінальна чернетка); Сергій Бучик – написання (перегляд і редагування), визначення методів дослідження, опис запропонованої архітектури.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

П'ятигор, В., & Бучик, С. (2025). Проблеми виявлення автоматизованих аккаунтів в соціальних мережах. У С. Ю. Даков, & О. С. Торошанко (Ред.), *Проблеми кібербезпеки інформаційно-телекомунікаційних систем: зб. матеріалів доповідей та тез* (с. 124–126). Київський національний університет імені Тараса Шевченка.

Almerekhi, H., & Elsayed, T. (2015). Detecting Automatically-Generated Arabic Tweets. *Lecture Notes in Computer Science,* 9460, 122–134. https://doi.org/10.1007/978-3-319-28940-3_10

Bessi, A., & Ferrara, E. (2016). *Social bots distort the 2016 U.S. Presidential election online discussion.* First Monday. https://doi.org/10.5210/fm.v21i11.7090

Boshmaf, Y., Logothetis, D., Siganos, G., Lería, J., Lorenzo, J., Ripeanu, M., Beznosov, K., & Halawa, H. (2016). Íntegro: Leveraging victim prediction for robust fake account detection in large scale OSNs. *Computers & Security,* 61, 142–168. https://doi.org/10.1016/j.cose.2016.05.005

Dixon, S. J. (2024, July 10).*Biggest social media platforms by users 2024.* Statista. https://www.statista.com/statistics/272014/global-social-/networks-ranked-by-number-of-users/

Dixon, S. J. (2025, February 6). *Facebook fake account deletion per quarter 2024*. Statista. https://www.statista.com/statistics/1013474/facebook-fake-account-removal-quarter/

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM,* 59(7), 96–104. https://doi.org/10.1145/2818717

Fredhaim, R., & Stolze, M. (2022, May 24). *Robotrolling 2022 (1).* NATO Strategic Communications Centre of Excellence. https://stratcomcoe.org/publications/robotrolling-20221/243

Guy, S., Ratzki-Leewing, A., Bahati, R., & Gwadry-Sridhar, F. (2012). Social Media: A Systematic Review to Understand the Evidence and Application in Infodemiology. *Lecture Notes of the Institute for*

ISSN 2707-1758

Computer Sciences, Social-Informatics and Telecommunications Engineering, 91, 1–8. https://doi.org/10.1007/978-3-642-29262-0_1

References

Almerekhi, H., & Elsayed, T. (2015). Detecting Automatically-Generated Arabic Tweets. *Lecture Notes in Computer Science,* 9460, 122–134. https://doi.org/10.1007/978-3-319-28940-3_10

Bessi, A., & Ferrara, E. (2016). *Social bots distort the 2016 U.S. Presidential election online discussion.* First Monday. https://doi.org/10.5210/fm.v21i11.7090

Boshmaf, Y., Logothetis, D., Siganos, G., Lería, J., Lorenzo, J., Ripeanu, M., Beznosov, K., & Halawa, H. (2016). Íntegro: Leveraging victim prediction for robust fake account detection in large scale OSNs. *Computers & Security,* 61, 142–168. https://doi.org/10.1016/j.cose.2016.05.005

Dixon, S. J. (2024, July 10).*Biggest social media platforms by users 2024.* Statista. https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Dixon, S. J. (2025, February 6). *Facebook fake account deletion per quarter 2024.* Statista. https://www.statista.com/statistics/1013474/facebook-fake-account-removal-quarter/

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM,* 59(7), 96–104. https://doi.org/10.1145/2818717

Fredhaim, R., & Stolze, M. (2022, May 24). *Robotrolling 2022 (1).* NATO Strategic Communications Centre of Excellence. https://stratcomcoe.org/publications/robotrolling-20221/243

Guy, S., Ratzki-Leewing, A., Bahati, R., & Gwadry-Sridhar, F. (2012). Social Media: A Systematic Review to Understand the Evidence and Application in Infodemiology. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering,* 91, 1–8. https://doi.org/10.1007/978-3-642-29262-0_1

Pyatigor, V., & Buchyk, S. (2025). Problems of detecting automated accounts in social networks. In S. Yu. Dakov, & O. S. Toroshanko (Eds.), *Problems of cybersecurity of information and telecommunication systems: collection of materials of reports and abstracts* (pp. 124–126). Taras Shevchenko National University of Kyiv [in Ukrainian].

Отримано редакцією журналу / Received: 21.05.2025
Прорецензовано / Revised:29.05.2025
Схвалено до друку / Accepted:22.06.2025

Serhii BUCHYK, DSc (Engin.), Prof.
ORCID ID: 0000-0003-0892-3494
e-mail: buchyk@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Vitalii PIATYHOR, PhD Student
ORCID ID: 0000-0002-7621-1299
e-mail: vp5gor@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

SOCIAL MEDIA AUTOMATED ACCOUNT (BOT) DETECTION SYSTEM ARCHITECTURE

Background. ***Social networks are one of the main sources of information. However, as trust in them grows, so does the use of automated accounts to spread disinformation and influence public opinion, which is an effective method of manipulation. Automated account detection systems are used to detect such accounts. The purpose of the article is to analyze the architecture of automated account (bot) detection systems in social networks and to identify the main directions for improving such architecture, taking into account the identified shortcomings.***

Methods. ***Methods of analysis, systematization, and generalization were used to identify shortcomings and areas for system improvement, and a modeling method was used to develop generalized and improved system architecture.***

Results. ***An improved architecture of a system for detecting automated accounts (bots) in the social network x.com is proposed, which uses a mixed model of data analysis using artificial intelligence and combines web scraping and API queries for data collection.***

Conclusions. ***The general architecture of automated account (bot) detection systems in social networks was analyzed and the main shortcomings and areas for improvement were identified, among which the following can be distinguished: restrictions on API requests from social network owners, unstructured and free style of posts in social networks, and constant changes in post generation algorithms by automated accounts. An improved architecture of the automated account (bot) detection system was proposed, which combines web scraping and API requests for data collection and allows parallel execution of social network data analysis.***

Keywords: ***computer system, data collection, data analysis, machine learning.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

УДК 004.056:519.8
DOI: https://doi.org/10.17721/ISTS.2025.9.18-25

Володимир АХРАМОВИЧ, д-р техн. наук, проф.
ORCID ID: 0000-0002-0086-9131
e-mail: 12z@ukr.net
Державний університет "Київський авіаційний інститут", Київ, Україна

Вадим АХРАМОВИЧ, завідувач комп'ютерним центром
ORCID ID: 0009-0003-2787-8745
e-mail: 12zstzi@gmail.com
Національна академія статистики, обліку та аудиту, Київ, Україна

Микола БРАІЛОВСЬКИЙ, канд. техн. наук, доц.
ORCID ID: 0000-0002-3148-1148
e-mail: brailovskyim@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Юрій ПЕПА, канд. техн. наук, доц.
ORCID ID: 0000-0003-2073-1364
e-mail: yurka14@gmail.com
Державний університет інформаційно-комунікаційних технологій, Київ, Україна

Тетяна ЛАПТЄВА, д-р філософії
ORCID ID: 0000-0002-5223-9078
e-mail: tetiana1986@ukr.net
Державний торговельно-економічний університет, Київ, Україна

КІЛЬКІСНЕ ДОСЛІДЖЕННЯ РИЗИКІВ МЕТОДОМ НЕЧІТКИХ МНОЖИН

Вступ. ***Представлено метод кількісного дослідження ризиків, що базується на аналізі й оцінюванні ризиків інформаційних систем. Запропонований підхід дозволяє використовувати широкий спектр параметрів, які забезпечують створення гнучких засобів оцінювання. Цей метод дає можливість розраховувати ризики як на основі статистичних даних, так і на основі експертних оцінок, зроблених в умовах невизначеності та слабкоформалізованого середовища.***

Розроблені методи забезпечують представлення результатів у числовій і словесній формах. Наприклад, можливе використання лінгвістичних змінних, які часто застосовують для опису складних систем, що характеризуються параметрами не лише у кількісному, але й у якісному вигляді. Ризики інформаційних систем можуть бути описані через концептуальну модель нечітких множин, яка враховує невизначеність, неточність і суб'єктивність під час їхнього оцінювання.

Методи. ***Для дослідження ризиків інформаційних систем застосовано метод нечітких множин. Цей метод дозволяє прогнозувати можливі економічні збитки. Запропонований підхід забезпечує можливість інтеграції різних факторів ризику в єдину модель, враховуючи як кількісні, так і якісні аспекти їхнього оцінювання.***

Результати. ***У ході дослідження розроблено кількісний підхід до оцінювання ризиків інформаційних систем підприємств, що дозволяє комплексно аналізувати й оцінювати вплив різноманітних факторів ризику.***

Модель, яка враховує невизначеність, неточність і суб'єктивність даних, що особливо важливо в умовах нестабільного середовища. Це забезпечує високу адаптивність методу до різних сфер діяльності підприємства.

Запропоновано методику, яка дозволяє здійснювати кількісне оцінювання потенційних втрат, пов'язаних із ризиками в інформаційних системах, на основі статистичних та експертних даних.

На основі результатів оцінювання ризиків можуть бути розроблені рекомендації щодо вдосконалення заходів із захисту інформаційних активів підприємства. Це включає створення адаптивних стратегій захисту, орієнтованих на зменшення економічних втрат.

Висновки. ***Використання методу нечітких множин для кількісного оцінювання ризиків інформаційних систем довело свою ефективність завдяки здатності враховувати невизначеність і суб'єктивність в оцінюванні даних. Розроблений підхід дозволяє не лише прогнозувати ризики, але й приймати обґрунтовані рішення щодо зменшення потенційних втрат, що є вагомим внеском у розвиток систем управління інформаційною безпекою підприємств.***

Ключові слова: ***ризики, прогнозування, втрати, система захисту, кібербезпека, захист інформації.***

Вступ

У сучасному світі інформаційні системи відіграють ключову роль у діяльності підприємств різних галузей, забезпечуючи оброблення, зберігання та передачу даних. Водночас зростає складність і багатогранність ризиків, які можуть суттєво впливати на їхню ефективність та безпеку. Інформаційні ризики, пов'язані з втратами, витоками або пошкодженнями даних, стають усе критичнішими для бізнесу через постійно зростаючу залежність від інформаційних технологій.

Нині ефективне оцінювання ризиків інформаційних систем є складним завданням, оскільки вимагає врахування низки факторів: невизначеності даних, багатовимірності ризиків, економічних і технічних аспектів, а також впливу зовнішнього середовища. Традиційні методи оцінювання часто не здатні повністю враховувати ці аспекти, особливо у випадках, коли відсутні точні або повні дані.

У цьому контексті особливої актуальності набувають підходи, що дозволяють інтегрувати кількісні та якісні аспекти оцінювання, включаючи суб'єктивні думки експертів і статистичні дані. Використання методів нечітких множин є перспективним рішенням цієї проблеми, оскільки вони дозволяють працювати з невизначеністю, неточністю та суб'єктивністю у процесі прийняття рішень.

У цій роботі запропоновано метод кількісного дослідження ризиків, який дозволяє:

- враховувати широкий спектр параметрів, включаючи економічні, управлінські та технічні аспекти;
- здійснювати розрахунок ризиків на основі статистичних даних та експертних оцінок, особливо у слабкоформалізованому середовищі;
- представляти результати оцінювання як у числовій, так і в словесній формах, використовуючи лінгвістичні змінні для опису складних систем.

ISSN 2707-1758

Метод також забезпечує адаптивність до специфіки підприємств різних галузей і уможливлює врахування впливу часу, зовнішніх факторів і внутрішньої структури організації. Запропонований підхід відкриває нові можливості для аналізу ризиків, їхнього моделювання та прогнозування економічних наслідків.

Отже, у роботі розглянуто актуальну проблему аналізу ризиків інформаційних систем із використанням сучасних методів, які сприяють точнішому прогнозуванню та ефективнішому управлінню ризиками. Отримані результати можуть бути інтегровані в систему управління підприємств, сприяючи підвищенню їхньої конкурентоспроможності та стійкості до інформаційних загроз.

Метою статті є розроблення й обґрунтування методу кількісного дослідження ризиків інформаційних систем підприємств, який враховує невизначеність, суб'єктивність і багатофакторність оцінок, а також забезпечує можливість прогнозування економічних наслідків ризиків і вдосконалення системи захисту інформаційних активів.

Огляд літератури. У статті (Korystin et al., 2024) Проведено соціальне опитування та дослідження експертної думки з метою реалізації загальної концепції стратегічного аналізу кібербезпеки в Україні. За допомогою методу, заснованого на визначенні середнього значення в певному наборі оцінок, а також методу, заснованого на теорії нечітких множин, було оцінено ризики поширення окремих кіберзагроз в Україні. Результати зведено для порівняння. Хоча використання різних методів вимірювання призвело до певних відмінностей у кількісних показниках ризику, порівняльний аналіз співвідношення рівнів різних кіберзагроз суттєво не змінився. Водночас метод нечітких наборів забезпечив гнучкішу інтерпретацію результатів для характеристики кіберзагроз із погляду їхнього висхідного або низхідного тренду.

У статті (Korchenko et al., 2021) відмічено, що один з актуальних напрямів, що розвиваються в області інформаційної безпеки, пов'язаний із використанням Honeypots (віртуальних приманок, онлайн-пасток), а вибір критеріїв для визначення найефективніших Honeypots і їхньої подальшої класифікації є актуальним завданням. Представлено основні продукти, в яких реалізовано технології віртуальної приманки. Вони часто використовуються для вивчення поведінки, підходів і методів, які застосовує стороння сторона для отримання несанкціонованого доступу до ресурсів інформаційної системи. Онлайн-хуки можуть імітувати будь-який ресурс, але частіше вони виглядають як справжні продакшн-сервери та робочі станції. Відомий ряд досить ефективних розробок, які використовують для розв'язання завдань виявлення атак на ресурси інформаційної системи (Barabash et al., 2025), в основі яких лежить апарат нечітких множин. Вони показали ефективність відповідного математичного апарату, використання якого, наприклад, для формалізації підходу до формування набору еталонних значень, дозволить поліпшити процес визначення найефективніших Honeypots. Для цього сформовано безліч характеристик (процес встановлення та налаштування, процес використання та підтримки, збір даних, рівень логування, рівень моделювання, рівень взаємодії), які визначають властивості онлайн-пасток. Ці характеристики стали основою для розроблення методу формування стандартів лінгвістичних змінних для подальшого відбору найбільш ефективних Honeypots.

У статті (Кочетков, Гаур, & Машін, 2019) розглянуто процес створення нечіткої продукційної моделі оцінювання ризику інформаційної безпеки підприємства. Показано, що традиційні методи недостатньо придатні для розв'язування подібних завдань саме тому, що вони не завжди можуть чітко описати умови і надати необхідні дані для прийняття відповідних рішень, зазвичай, виникає невизначеність. Показано, що існуючі методи врахування й оцінки ризиків не позбавлені суб'єктивізму й суттєвих умов, що призводять до неправильних оцінок ризиків проєктів. Суттєво, що якісно виконаний аналіз інформаційних ризиків дозволяє провести порівняльний аналіз "ефективність – вартість" різних варіантів захисту, обрати адекватні контрзаходи і засоби контролю, оцінити рівень залишкових ризиків. Крім того, інструментальні засоби аналізу ризиків (Собчук та ін., 2023), що ґрунтуються на сучасних базах знань (Yevseiev et al., 2022) і процедурах логічного висновку, дозволяють побудувати структурні й об'єктно-орієнтовані моделі інформаційних активів компанії, моделі загроз і моделі ризиків, пов'язаних з окремими інформаційними та бізнес-транзакціями і, отже, виявляти такі інформаційні активи компанії, ризик порушення захищеності яких є критичним, тобто неприйнятним. Запропоновано використання теорії нечіткої логіки для оцінювання ризиків (Barabash, Laptiev, & Grushina, 2023).

Для моделювання ризику інформаційної безпеки підприємства запропоновано нечіткі моделі надавати у вигляді нечітких мереж (Хорошко та ін., 2024). Модель містить бази правил і дозволяє проводити лінгвістичний аналіз ризиків, які несуть потенційні загрози і збиток організації. Взаємозв'язок між факторами (антецедентом) і показниками ризику (консеквентом) являє собою бінарне нечітке відношення на декартовому множенні відповідних нечітких множин. Нечітке причинно-наслідкове відношення між антецедентом і консеквентом задають у вигляді нечіткої продукції.

У статті (Yevseiev, Shmatko, & Romashchenko, 2019) вказано на те, що одним із нових і перспективних підходів до розв'язання проблеми оцінювання кібербезпеки на об'єктах критичної інфраструктури є використання теорії нечітких множин, наприклад, для оцінювання інформаційної безпеки. На практиці трапляються ситуації, коли на розрахунок кінцевих результатів істотно впливають невідповідності висновків або помилки експертів.

У статті (Varela, & Domingues, 2022) відмічено, що ризик є однією з найважливіших складових проєкту. Його правильне оцінювання та лікування збільшують шанси на успіх проєкту. У цій статті представлено ризики в проєктах з Data Science, оцінені за допомогою дослідження, проведеного за методикою Delphi, щоб відповісти на питання: "Які ризики проєктів з Data Science?". Дослідження дозволило виявити конкретні ризики, пов'язані з проєктами з Data Science, проте вдалося перевірити, що більше половини найбільш згадуваних ризиків схожі з іншими типами IT-проєктів.

Методи

Для дослідження ризиків інформаційних систем застосовано метод нечітких множин. Цей метод дозволяє:

- прогнозувати можливі економічні збитки;
- розробляти стратегії покращення й оптимізації системи захисту інформації (Лаптєв та ін., 2024).

Запропонований підхід забезпечує можливість інтеграції різних факторів ризику в єдину модель, враховуючи як кількісні, так і якісні аспекти їхнього оцінювання.

ISSN 2707-1758

Результати

Основні аспекти опису:

1. Нечіткість ризиків.

Ризики досить зрідка мають чіткі межі, а їхні наслідки залежать від багатьох факторів, часто недостатньо визначених.

Наприклад, ризик втрати даних можна описати нечіткою множиною: "високий ризик", "середній ризик", "низький ризик", що базується на експертних оцінках або історичних даних.

2. Лінгвістичні змінні.

Для моделювання ризиків використовують лінгвістичні змінні (Лукова-Чуйко, & Лаптєва, 2024), такі як:

- імовірність ризику (низька, середня, висока);
- вплив ризику (незначний, помірний, критичний).

Ці змінні перетворюються на нечіткі множини, які визначаються функціями належності.

3. Функції належності.

Функції належності відображають, наскільки певний ризик належить до кожної категорії. Наприклад, для "високого ризику":

$$\mu(x) = \begin{cases} 0; & x \le a; \\ \dfrac{x-a}{b-a}; & a \lhd x \le b; \\ 1; & x \rhd b. \end{cases}$$

де x – оцінка ризику; a і b – межі категорій.

4. Композиція ризиків.

Використовують правила нечіткої логіки для аналізу взаємозв'язків між ризиками. Наприклад, якщо ймовірність ризику висока і вплив критичний, то загроза є серйозною.

5. Ранжування та прийняття рішень.

На основі інтегральних показників (напр., методу центра ваги) здійснюють ранжування ризиків. Прийняття рішень враховує нечіткість оцінок, що дозволяє вибрати найкращі стратегії для їхньої мінімізації.

Наприклад, ризик кібератаки можна описати як

- імовірність: середня (0.5), висока (0.7);
- вплив: критичний (0.8).

Висновок – ризик розміщено в зоні високої небезпеки (за нечіткими правилами).

Нечіткі множини дозволяють ефективно працювати з недостатньо структурованими даними, враховуючи складність і багатофакторність ризиків інформаційних систем.

Методи трапеції та трикутника широко використовують у теорії нечітких множин для визначення функцій належності ризиків. Наприклад:

1. Метод трикутника.

Функція належності у формі трикутника визначається трьома параметрами: (a, b, c), тут a – ліва межа (мінімальне значення, де належність = 0); b – пік трикутника (максимальна належність = 1); c – права межа (значення, де належність знову = 0):

$$\mu(x) = \begin{cases} 0, & x \le a \quad \text{або} \quad x; \\ \dfrac{x-a}{b-a}, & a < x \le b; \\ \dfrac{c-x}{c-b}, & b < x < c. \end{cases}$$

Розглянемо приклад.

Нехай імовірність ризику втрати даних оцінюють за шкалою 0–10. Тоді "середній ризик" має трикутну форму з параметрами (3, 5, 7):

$$\mu(4) = \frac{4-3}{5-3} = 0.5 \text{ при } x = 4;$$

$$\mu(6) = \frac{7-6}{7-5} = 0.5 \text{ при } x = 6.$$

Отже, коли $x \equiv 4x$ і $x \equiv 6x$, то вони належать до "середнього ризику" на 50 %, а коли $x \equiv 5x$, то це відповідає 100 % "середнього ризику", бо це пікове значення.

2. Метод трапеції.

Функцію належності у формі трапеції визначають чотирма параметрами: (a, b, c, d), де a – ліва межа (належність = 0); b – початок плато (належність починає = 1); c – кінець плато (належність = 1); d – права межа (належність знову = 0):

$$\mu(x) = \begin{cases} 0, & x \le a \quad \text{або} \quad x; \\ \dfrac{x-a}{b-a}, & a < x \le b; \\ 1, & b < x \le c; \\ \dfrac{d-x}{d-c}, & b < x < c. \end{cases}$$

Розглянемо приклад.

Оцінимо рівень кібератаки на інформаційну систему. "Критичний вплив" описуємо трапецією з параметрами (6, 8, 10, 12):

$$\mu(7) = \frac{7-6}{8-6} = 0.5 \text{ при } x = 7;$$

$$\mu(9) = 1, \text{ оскільки } 8 < x \le 10;$$

$$\mu(11) = \frac{12-11}{12-10} = 0.5 \text{ при } x = 11.$$

Отже, коли $x \equiv 7x$, то вони належать до "критичного впливу" на 50 %, а при $x \equiv 9x$ виходить "критичний вплив" зі 100 %.

3. Комбінація методів.

Аналізуючи ризики часто комбінують трикутні та трапецієподібні функції. Причому, зазвичай імовірність ризику моделюється трикутником, бо вона має пікову оцінку, а вплив ризику моделюється трапецією, бо він стабільно високий у певному діапазоні.

4. Візуалізація.

Трикутник використовують, коли потрібно акцентувати на єдиному піковому значенні. Трапеція підходить для опису стабільного рівня ризику в межах певного інтервалу.

Ці методи дозволяють адаптивно моделювати ризики з урахуванням нечітких меж і полегшують їхню інтерпретацію.

Фазифікація та дефазифікація на основі оцінювання ризиків інформаційної системи.

1. Постановка задачі.

Оцінимо ризик кібератаки на основі двох змінних: імовірність атаки (P) у відсотках (0–100 %); вплив атаки (I) за шкалою 0–10. Вихідна змінна – рівень ризику (R) за шкалою 0–10.

Використаємо три лінгвістичні змінні для кожного з параметрів: 1) імовірність атаки: низька (Low), середня (Medium), висока (High); 2) вплив атаки: незначний (Low), помірний (Medium), критичний (High); 3) рівень ризику: низький (Low), середній (Medium), високий (High).

2. Фазифікація.

Під фазифікацією розумітимемо процес перетворення числових даних на нечіткі множини.

Візьмемо, наприклад такі дані: P = 65 %; $I = 7$ та оцінимо функції належності.

ISSN 2707-1758

Для ймовірності P:

$$\text{Low: трикутник } (0,20,40) \to \mu_{\text{Low}}(65) = 0;$$

$$\text{Medium: трапеція } (30,50,70,90) \to \mu_{\text{Medium}}(65) = \frac{90-65}{90-70} = 0.25;$$

$$\text{High: трикутник } (60,80,100) \to \mu_{\text{High}}(65) = \frac{65-60}{80-60} = 0.25.$$

Для впливу I:

$$\text{Low: трикутник } (0,2,4) \to \mu_{\text{Low}}(7) = 0;$$

$$\text{Medium: трапеція } (3,5,7,9) \to \mu_{\text{Medium}}(7) = 0.5;$$

$$\text{High: трикутник } (6,8,10) \to \mu_{\text{High}}(7) = 0.5.$$

Фазифіковані значення:
для P: Medium = 0.25; High = 0.25;
для I: Medium = 0.5; High = 0.5.

3. Правила нечіткої логіки.

На основі фазифікованих правил використовуємо таку базу правил:

1) якщо P_{Low} і $I_{\text{Low}} \to R_{\text{Low}}$;

2) якщо P_{Medium} і $I_{\text{Medium}} \to R_{\text{Medium}}$;

3) якщо P_{High} і $I_{\text{High}} \to R_{\text{High}}$;

4) якщо P_{High} і $I_{\text{Medium}} \to R_{\text{High}}$,

а також інші два правила (з урахуванням ступенів належності):

5) для правила 2): min(0.25, 0.5) = 0.25;

6) для правила 4): min(0.25, 0.5) = 0.25.

4. Агрегація.

Агрегація об'єднує вихідні ступені належності для різних правил у нечітку множину вихідної змінної R. В результаті:

$R_{\text{Medium}} = 0.25$;

$R_{\text{High}} = 0.25.$

5. Дефазифікація.

Дефазифікація перетворює нечітку множину R на конкретне числове значення. Використаємо метод центра ваги (centroid):

$$R = \frac{\int x\mu_R(x)dx}{\int \mu_R(x)dx}.$$

Проведемо розрахунки.

Припустимо, що R_{Medium} задано трапецією (4, 5, 6, 7); а R_{High} – трикутником (6, 8, 10), тоді ваги:

$\mu_{\text{Medium}}(x) = 0.25;$

$\mu_{\text{High}}(x) = 0.25.$

Обчислення центра ваги дає

$$R = \frac{5.5 \cdot 0.25 + 8 \cdot 0.25}{0.25 + 0.25} = 6.75.$$

Висновок. Рівень ризику становить 6.75 за шкалою 0–10. Це вказує на помірно високий рівень ризику.

Розглянемо два приклади.

Приклад 1.

Необхідно підрахувати збиток за рік банку у разі прогнозованих показників, спираючись на попередні статистичні дані підприємства, а саме: ймовірність 24 атак; вплив атаки – критичний; рівень ризику – низький (Low); імовірність 250 атак; вплив атаки – середній; рівень ризику – середній (Medium); імовірність 3250 атак; вплив атаки – низький.

Критичний рівень – збитки від однієї атаки від 1 млн дол. до 500 млн дол. США.

Середній рівень – збитки від однієї атаки від 50 тис. дол. до 500 тис. дол. США.

Низький рівень – збитки від однієї атаки від 200 дол. до 10 тис. дол. США.

Відсоток нейтралізації атак за рахунок системи захисту (99–99,9) %.

Імовірність атаки (P) у відсотках (0–100 %).

Вплив атаки (I) за шкалою 0–10.

Для розрахунку можливого збитку за рік роботи банку необхідно виконати такі кроки:

1) Визначення параметрів і нечітких множин:

- Імовірність атак (кількість атак на рік);
- вплив атаки (розмір збитків від однієї атаки);
- рівень ризику (Low, Medium, High);
- відсоток нейтралізації атак.

2) Формування нечітких множин для оцінювання збитків.

Для кожного рівня ризику (Low, Medium, High) задаємо діапазони. Для критичного рівня (High Risk):

- збитки за одну атаку (1 млн дол., 500 млн дол. США);
- кількість атак (1, 24);
- загальні збитки

(1 10^6 дол. США, 24500 10^6 дол. США);

- середній збиток 250 500 000 дол. США;
- успішно реалізується 1 % атак.

Для середнього рівня (Medium Risk):

- збитки за одну атаку
- (50 тис. дол., 500 тис. дол. США);
- кількість атак (10, 250);
- загальні збитки

(10 10^5 дол. США, 250500 10^5 дол. США);

- середній збиток 275 000 дол. США;
- успішно реалізується 1 % атак.

Для низького рівня (Low Risk):

- збитки за одну атаку (200 дол., 10 тис. дол. США).
- кількість атак (300, 3250);
- загальні збитки

(300200 дол. США, 3250 10 10^5 дол. США);

- середній збиток 5 100 дол. США;
- успішно реалізується 0,1 % атак.

3) Розрахункова формула для кожного рівня ризику:

$$\text{Збиток} = \text{Середня кількість атак} \times \text{Середній збиток} \times \frac{1 - \text{Відсоток нейтралізації}}{100}.$$

Середня кількість атак обчислюється як середнє значення діапазону атак.

4) Розрахунок можливого збитку.

Для критичного рівня (High Risk):
P = 24; I = 10; середній збиток 250 500 000 дол. США; нейтралізація загроз 99 %.

ISSN 2707-1758

Збиток критичний = 24×1×10×250500000×0,01=601 200 000 дол. США.

Для середнього рівня (Medium Risk):

P = 250; I = 10; середній збиток 275 000 дол. США; нейтралізація загроз 99 %.

Збиток середній = 250×1×5×275000×0,01=3 437 500 дол. США.

Для низького рівня (Low Risk):

P = 3250; I = 1; середній збиток 5 100 дол. США; нейтралізація загроз 99,9 %.

Збиток низький = 3250×1×5100×0,001=16 575 дол. США.

Загальний можливий збиток оцінимо за формулою

Загальний збиток = Збиток критичний + Збиток середній + Збиток низький;

Загальний збиток = 601 200 000 + 3 437 500 + 16 575 = 604 654 075 дол. США.

Приклад розробленої програми для обчислення збитків мовою програмування Python

```
# ------------------------------------------------------------------
# Параметри завдання
# Збитки за одну атаку (у доларах США)
loss_per_attack = {
    "High": (1e6, 500e6),      # Критичний рівень
    "Medium": (50e3, 500e3),   # Середній рівень
    "Low": (200, 10e3)         # Низький рівень
}

# Програма на Python, яка використовується для розрахунків

# Кількість атак
attacks_range = {
    "High": (1, 24),           # Критичний рівень
    "Medium": (10, 250),       # Середній рівень
    "Low": (300, 3250)         # Низький рівень
}

# Процент нейтралізації атак (у частках)
protection_efficiency = (0.99, 0.999)  # Від 99% до 99.9%

# Розрахунок загальних збитків
def calculate_losses(loss_range, attack_range, protection_range):

    # Розрахунок кількості пропущених атак
    min_attacks = attack_range[0] * (1 - protection_range[1])  # Мінімум атак, які
                                                               # прорвались
    max_attacks = attack_range[1] * (1 - protection_range[0])  # Максимум атак, які
                                                               # прорвались

    # Розрахунок збитків
    min_loss = min_attacks * loss_range[0]  # Мінімальний збиток
    max_loss = max_attacks * loss_range[1]  # Максимальний збиток

    return (min_loss, max_loss)

# Розрахунок для кожного рівня ризику
total_losses = {}
for risk_level in loss_per_attack.keys():
    total_losses[risk_level] = calculate_losses
    (
        loss_range=loss_per_attack[risk_level],
        attack_range=attacks_range[risk_level],
        protection_range=protection_efficiency
    )

# Загальні збитки по всіх рівнях
overall_min_loss = sum(loss[0] for loss in total_losses.values())
overall_max_loss = sum(loss[1] for loss in total_losses.values())

# Вивід результатів
total_losses, (overall_min_loss, overall_max_loss)
# ------------------------------------------------------------------
```


ISSN 2707-1758

Ця програма спочатку задає діапазони кількості атак і збитків для кожного рівня ризику. Потім вона обчислює кількість пропущених атак, враховуючи ефективність захисту, і підраховує загальні збитки для кожного рівня ризику, а також суму всіх рівнів.

Приклад 2.

Оцінювання збитків банку за допомогою нечітких множин.

Опишемо початкові умови.

Банк стикається з кібератаками протягом року. Статистичні дані свідчать, що загальний прогнозований збиток за максимальної кількості атак становить 5 млн дол. США. Кількість атак оцінюється в діапазоні від 0 до 5000. Необхідно оцінити збитки для конкретного випадку, наприклад, коли очікувана кількість атак за рік становить 3000.

Розв'язання.

1. Визначення нечітких множин.

1) Кількість атак (x):

"Мала" (L): $x \le 2000$;

"Середня" (M): $2000 < x \le 4000$;

"Велика" (H): $x > 4000$.

2) Збиток (S):

"Низький" (L): $S \le 2$ млн дол. США;

"Середній" (M): $2 < S \le 4$ млн дол. США;

"Високий" (H): $S > 4$ млн дол. США.

2. Функції належності.

Розглянемо функції належності для різних кількостей атак.

"Мала" кількість атак:

$$\mu L(x) = \begin{cases} 1, & x \ge 1000; \\ \dfrac{2000 - x}{1000}, & 1000 < x \le 2000; \\ 0, & x > 2000. \end{cases}$$

"Середня" кількість атак:

$$\mu M(x) = \begin{cases} 0, & x \ge 2000; \\ \dfrac{x - 2000}{1000}, & 2000 < x \le 3000; \\ \dfrac{4000 - x}{1000}, & 3000 < x \le 4000; \\ 0, & x > 4000. \end{cases}$$

"Велика" кількість атак:

$$\mu H(x) = \begin{cases} 0, & x \le 3000; \\ \dfrac{2000 - x}{1000}, & 1000 < x \le 2000; \\ 0, & x > 2000. \end{cases}$$

Розрахунки для збитків.

"Низький" рівень:

$$\mu L(S) = \begin{cases} 1, & S \le 2; \\ \dfrac{4 - S}{2}, & 2 < S \le 4; \\ 0, & S > 4. \end{cases}$$

"Середній" рівень:

$$\mu M(S) = \begin{cases} 0, & S \le 2; \\ \dfrac{S - 2}{2}, & 2 < S \le 4; \\ \dfrac{6 - S}{1000}, & 4 < S \le 6; \\ 0, & S > 6. \end{cases} \quad (1)$$

"Високий" рівень:

$$\mu H(S) = \begin{cases} 0, & S \le 4; \\ \dfrac{S - 4}{2}, & 4 < S \le 6; \\ 1, & S > 6. \end{cases}$$

3. База правил:

- якщо кількість атак "Мала", то збиток "Низький";
- якщо кількість атак "Середня", то збиток "Середній";
- якщо кількість атак "Велика", то збиток "Високий".

4. Розрахунок збитків.

Для x = 3000 ::

- "Мала" кількість: $\mu L(3000) = 0$ (поза межами);
- "Середня" кількість: $\mu M(3000) = 1$;
- "Велика" кількість: $\mu H(3000) = 0$.

Отже, активується правило: "якщо атак "Середня" кількість, то збиток "Середній"", тоді для збитків середній рівень S знаходять із формули (1).

Використаємо дефазифікацію методом центрування ваги:

$$S = \frac{\int_2^6 S\mu M(S)dS}{\int_2^6 \mu M(S)dS} = \frac{14}{4} = 3.5 \text{ млн дол. США.}$$

Відповідь. Прогнозований збиток банку для 3000 атак становить 3.5 млн дол. США.

Дискусія і висновки

У роботі представлено метод кількісного дослідження ризиків, що базується на аналізі й оцінюванні ризиків інформаційних систем. Запропонований підхід дозволяє використовувати широкий спектр параметрів, які забезпечують створення гнучких засобів оцінювання. Цей метод дає можливість розраховувати ризики як на основі статистичних даних, так і на основі експертних оцінок, зроблених в умовах невизначеності та слабкоформалізованого середовища (Лукова-Чуйко, & Лаптєва, 2024).

Особливістю підходу є врахування таких факторів:

- період часу;
- галузь діяльності;
- економічна й управлінська специфіки підприємства.

ISSN 2707-1758

Крім цього, розроблені методи забезпечують представлення результатів у числовій і словесній формах. Наприклад, можливе використання лінгвістичних змінних, які часто застосовують для опису складних систем, що характеризуються параметрами не лише у кількісному, але й у якісному вигляді. Ризики інформаційних систем можуть бути описані через концептуальну модель нечітких множин, яка враховує невизначеність, неточність і суб'єктивність під час їхнього оцінювання.

У ході дослідження розроблено кількісний підхід до оцінювання ризиків інформаційних систем підприємств, що дозволяє комплексно аналізувати й оцінювати вплив різноманітних факторів ризику. Основні результати роботи включають таке.

Розроблення моделі оцінювання ризиків на основі нечітких множин:

1. Модель враховує невизначеність, неточність і суб'єктивність даних, що особливо важливо в умовах нестабільного середовища. Це забезпечує високу адаптивність методу до різних сфер діяльності підприємства.
2. Прогнозування економічних збитків.
3. Запропонована методика дозволяє здійснювати кількісне оцінювання потенційних втрат, пов'язаних із ризиками в інформаційних системах, на основі статистичних та експертних даних. Це сприяє кращому розумінню масштабів можливих наслідків інформаційних інцидентів.
4. На основі результатів оцінювання ризиків можуть бути розроблені рекомендації щодо вдосконалення заходів із захисту інформаційних активів підприємства. Це включає створення адаптивних стратегій захисту, орієнтованих на зменшення економічних втрат.
5. Розроблена система відображення результатів дозволяє представляти дані в числовому і словесному виглядах, що підвищує зрозумілість і зручність для різних категорій користувачів, включаючи управлінців і технічних фахівців.
6. Методика адаптується до специфіки різних підприємств і галузей, забезпечуючи гнучкість у застосуванні для компаній із різною структурою та масштабом діяльності.

За результатами роботи можна зробити такі висновки:

1. Ефективність запропонованого методу.

Використання методу нечітких множин для кількісного оцінювання ризиків інформаційних систем довело свою ефективність завдяки здатності враховувати невизначеність і суб'єктивність в оцінці даних.

2. Практична цінність дослідження.

Розроблений підхід дозволяє не лише прогнозувати ризики, але й приймати обґрунтовані рішення щодо зменшення потенційних втрат, що є вагомим внеском у розвиток систем управління інформаційною безпекою підприємств.

3. Інтеграція результатів у практику.

Отримані результати можуть бути інтегровані у процеси управління підприємством, сприяючи підвищенню їхньої стійкості до ризиків, пов'язаних з інформаційними системами. Запропоновані висновки демонструють значущість і потенціал подальшого розвитку методів оцінювання ризиків для підвищення ефективності управління інформаційною безпекою.

Напрями подальших досліджень і подальші роботи можуть бути спрямовані на розроблення автоматизованих інструментів для оцінювання ризиків на основі запропонованого методу, а також на поглиблене дослідження галузевих особливостей ризиків в інформаційних системах.

Внесок авторів: Володимир Ахрамович – концептуалізація; методологія; аналіз джерел; Вадим Ахрамович – збір емпіричних даних і валідація; емпіричне дослідження, підготування огляду літератури; Микола Браіловський – проєктування комп'ютерних програм; реалізація комп'ютерного коду та допоміжних алгоритмів; Юрій Пепа – тестування компонентів коду; Тетяна Лаптєва – написання початкового варіанта (чернетки) статті.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Кочетков, О. В., Гаур, Т. О., & Машін, В. М. (2019). Система оцінки ризиків інформаційної безпеки підприємства на основі нечіткої логіки. *Наукові праці ОНАЗ ім. О. С. Попова,* 1, 97–104. https://doi.org/10.33243/2518-7139-2019-1-1-97-104

Лаптєв, О. А., Колесник, В. В., Ровда, В. В. & Половінкін, М. І. (2024). Метод підвищення захисту особистих даних за рахунок синтезу резильєнтних віртуальних спільнот. *Сучасний захист інформації,* 4(60), 141–146. https://doi.org/10.31673/2409-7292.2024.040015

Лукова-Чуйко, Н., & Лаптєва, Т. (2024). Метод виявлення неправдивої інформації на основі експертної оцінки. *Захист інформації,* 26(1), 29–35. https://doi.org/10.18372/2410-7840./26.18822

Собчук, В. В., Циганівська, І. М., Лаптєв, О. А., & Журавльов, В. М. (2023). Планування технологічних ланцюжків засобами скінченно частково впорядкованих множин. *Наукоємні технології,* 60(4), 372–385. https://doi.org/10.18372/2310-5461.60.18266

Хорошко, В. О., Лаптєв, О. А., Хохлачева, Ю. Є., Аблуллах, Аль-Далваш Ф., & Пепа, Ю. В. (2024). Особливості проектування захищених інформаційних мереж. *Наукоємні технології,* 62(2), 154–163. https://doi.org/10.18372/2310-5461.62.18709

Barabash, O., Laptiev, O., & Grushina, O. (2023). The Conceptual Model of the Intelligent Network. *Сучасний захист інформації,* 4(56), 1–9. https://doi.org/10.1016/j.procs.2021.12.100 та https://doi.org/10.31673/2409-7292.2023.030202

Barabash, O., Sobchuk, V., Sobchuk, A., Musienko, A., & Laptiev, O. (2025). Algorithms for Synthesis of Functionally Stable Wireless Sensor Network. *Advanced Information Systems*, 9(1), 70–79. https://doi.org/10.20998/2522-9052.2025.1.08

Korchenko, A., Breslavskyi, V., Yevseiev, S., Zhumangalieva, N., Zvarych, A., Kazmirchuk, S., Kurchenko, O., Laptiev, O., Sievierinov, O., & Tkachuk, S. (2021). Development of a Method for Constructing Linguistic Standards for Multi-criteria assessment of Honeypot Efficiency. *Eastern European Journal of Enterprise Technologies,* 111(3/9), 63–83. https://doi.org/10.15587/1729-4061.2021.225346

Korystin, O., Korchenko, O., Kazmirchuk, S., Demediuk, S., & Korystin, O. (2024). Comparative Risk Assessment of Cyber Threats Based on Average and Fuzzy Set Theory. *International Journal of Computer Network and Information,* 16(1), 24–34. https://doi.org/10.5815/ijcnis.2024.01.02

Varela, C., & Domingues, L. (2022). Risks of Data Science Projects – A Delphi Study. *Procedia Computer Science,* 196, 982–989.

Yevseiev, S., Rzayev, K., Laptiev, O., Hasanov, R., Milov, O., Asgarova, B., Camalova, J., & Pohasii, S. (2022). Development of a Hardware Cryptosystem Based on a Random Number Generator with Two Types of Entropy Sources. *Eastern-European Journal of Enterprise Technologies,* Vol. 5, 9(119), 6–16. https://doi.org/10.15587/1729-4061./2022.265774

Yevseiev, S., Shmatko, O., & Romashchenko, N. (2019). Algorithm of Information Security Risk Assessment Based on Fuzzy-multiple Approach. *Сучасні інформаційні системи,* 3(2), 73–79. https://doi.org/10.20998/2522-9052.2019.2.13

References

Barabash, O., Laptiev, O., & Grushina, O. (2023). The Conceptual Model of the Intelligent Network. *Modern Information Security,* 4(56), 1–9 [in Ukrainian]. https://doi.org/10.31673/2409-7292.2023.030202

Barabash, O., Sobchuk, V., Sobchuk, A., Musienko, A., & Laptiev, O. (2025). Algorithms for Synthesis of Functionally Stable Wireless Sensor Network. *Advanced Information Systems,* 9(1), 70–79. https://doi.org/10.20998/2522-9052.2025.1.08

Khoroshko, V. O., Laptiev, O. A., Khokhlacheva, Y. E., Fouad Al-Dalvash A., & Pepa, Y. V. (2024). Features of Designing Secure Information Networks.*Scientific Technologies,* 62(2), 154–163 [in Ukrainian]. https://doi.org/10.18372/2310-5461.62.18709

Kochetkov, O. V., Gaur, T. O., & Mashin, V. M. (2019). Enterprise Information Security Risk Assessment System Based on Fuzzy Logic. *Scientific Works of O.S. Popov ONAT,* 1, 97–104 [in Ukrainian]. https://doi.org/10.33243/2518-7139-2019-1-1-97-104

Korchenko, A., Breslavskyi, V., Yevseiev, S., Zhumangalieva, N., Zvarych, A., Kazmirchuk, S., Kurchenko, O., Laptiev, O., Sievierinov, O., & Tkachuk, S. (2021). Development of a Method for Constructing Linguistic Standards for Multi-criteria assessment of Honeypot

ISSN 2707-1758

Efficiency. *Eastern European Journal of Enterprise Technologies,* 111(3/9), 63–83. https://doi.org/10.15587/1729-4061.2021.225346

Korystin, O., Korchenko, O., Kazmirchuk, S., Demediuk, S., & Korystin, O. (2024). Comparative Risk Assessment of Cyber Threats Based on Average and Fuzzy Set Theory. *International Journal of Computer Network and Information,* 16(1), 24–34. https://doi.org/10.5815/ijcnis.2024.01.02

Laptiev, O. A., Kolesnyk, V. V., Rovda, V. V., & Polovinkin, M. I. (2024). A Method for Increasing Personal Data Protection Through the Synthesis of Resilient Virtual Communities. *Modern Information Security,* 4(60), 141–146 [in Ukrainian]. https://doi.org/10.31673/2409-7292.2024.040015

Lukova-Chuyko, N., & Laptieva, T. (2024). Method of Detecting False Information Based on Expert Assessment. *Information Protection,* 26(1), 29–35 [in Ukrainian]. https://doi.org/10.18372/2410-7840.26.18822

Sobchuk, V. V., Tsyganivska, I. M., Laptev, O. A., & Zhuravlev, V. M. (2023). Planning of Technological Chains by Means of Finitely Partially Ordered Sets. *Science-intensive Technologies,* 60(4), 372–385 [in Ukrainian]. https://doi.org/10.18372/2310-5461.60.18266

Varela, C., & Domingues, L. (2022). Risks of Data Science Projects – A Delphi Study. *Procedia Computer Science,* 196, 982–989.

Yevseiev, S., Rzayev, K., Laptiev, O., Hasanov, R., Milov, O., Asgarova, B., Camalova, J., & Pohasii, S. (2022). Development of a Hardware Cryptosystem Based on a Random Number Generator with Two Types of Entropy Sources. *Eastern-European Journal of Enterprise Technologies,* Vol. 5, 9(119), 6–16. https://doi.org/10.15587/1729-4061.2022.265774

Yevseiev, S., Shmatko, O., & Romashchenko, N. (2019). Algorithm of Information Security Risk Assessment Based on Fuzzy-multiple Approach. *Сучасні інформаційні системи,* 3(2), 73–79. https://doi.org/10.20998/2522-9052.2019.2.13

Отримано редакцією журналу / Received: 17.05.25
Прорецензовано / Revised: 27.05.25
Схвалено до друку / Accepted: 13.06.25

Volodymyr AKHRAMOVYCH, DSc (Engin.), Prof.
ORCID ID:0000-0002-0086-9131
e-mail: 12z@ukr.net
State University "Kiev Aviation Institute", Kyiv, Ukraine

Vadym AKHRAMOVYCH, Head of the Computing Center
ORCID ID: 0009-0003-2787-8745
e-mail: 12zstzi@gmail.com
National Academy of Statistics, Accounting and Auditing, Kyiv, Ukraine

Mykola BRAILOVSKYI, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0002-3148-1148
e-mail: brailovskyim@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Yuriy PEPA, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0003-2073-1364
e-mail: yurka14@gmail.com
State University of Information and Communication Technologies, Kyiv, Ukraine

Tetiana LAPTIEVA, PhD
ORCID ID: 0000-0002-5223-9078
e-mail: tetiana1986@ukr.net
State University of Trade and Economics, Kyiv, Ukraine

QUANTITATIVE RISK ASSESSMENT USING THE FUZZY SETS METHOD

Background. ***This paper presents a quantitative risk assessment method based on the analysis and evaluation of risks in information systems. The proposed approach enables the use of a wide range of parameters, providing the creation of flexible assessment tools. This method allows calculating risks based on both statistical data and expert evaluations conducted under conditions of uncertainty and poorly formalized environments.***

Additionally, the developed methods provide the representation of results in both numerical and verbal forms. For example, linguistic variables, which are often used to describe complex systems characterized by both quantitative and qualitative parameters, can be utilized. The risks of information systems can be described through a conceptual fuzzy set model that accounts for uncertainty, imprecision, and subjectivity in their evaluation.

Methods. ***The fuzzy sets method was applied to study the risks of information systems. This method facilitates the prediction of potential economic losses. The proposed approach allows the integration of various risk factors into a unified model, considering both quantitative and qualitative aspects of their assessment.***

Result. ***During the study, a quantitative approach to assessing the risks of enterprise information systems was developed, enabling a comprehensive analysis and evaluation of the impact of various risk factors. The main results of the work include:***

The model accounts for uncertainty, imprecision, and subjectivity of data, which is especially important in unstable environments. This ensures high adaptability of the method to different areas of enterprise activity.

The proposed methodology enables the quantitative assessment of potential losses associated with risks in information systems based on statistical and expert data.

Based on the risk assessment results, recommendations can be developed to improve measures for protecting the enterprise's information assets. This includes the creation of adaptive protection strategies aimed at reducing economic losses.

Conclusions. ***The use of the fuzzy sets method for quantitative risk assessment of information systems has proven its effectiveness due to its ability to account for uncertainty and subjectivity in data evaluation. The developed approach not only predicts risks but also supports informed decision-making to reduce potential losses, contributing significantly to the development of enterprise information security management systems.***

Keywords: ***risks, forecasting, losses, protection system, cybersecurity, information protection.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

УДК 004.056.55:004.738.5
DOI: https://doi.org/10.17721/ISTS.2025.9.26-31

Іван ПАРХОМЕНКО, канд техн. наук, доц.
ORCID ID: 0000-0001-6889-9284
e-mail: ivan.parkhomenko@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Михайло САВОНІК, асп.
ORCID ID: 0009-0001-7622-978X
e-mail: mike.savonik+knu@gmail.com
Київський національний університет імені Тараса Шевченка, Київ, Україна

МЕТОДИ ВИЯВЛЕННЯ Й АНАЛІЗ АТАК НА ОСНОВІ МІСКОНФІГУРАЦІЙ У ХМАРНИХ СЕРВІСАХ

Вступ. *З розвитком хмарних технологій все більше організацій переходять до використання хмарних сервісів для зберігання даних і виконання обчислень. Проте неправильна конфігурація (місконфігурація) хмарних сервісів стає однією з головних причин виникнення вразливостей, що можуть бути використані зловмисниками для здійснення атак. Місконфігурації можуть призвести до несанкціонованого доступу до конфіденційних даних, компрометації систем та інших серйозних наслідків для безпеки. Метою цієї роботи є дослідження методів виявлення й аналізу атак, що виникають внаслідок місконфігурацій у хмарних сервісах, а також розроблення рекомендацій для підвищення рівня безпеки.*

Методи. *Проаналізовано існуючі підходи до виявлення місконфігурацій, включаючи використання автоматизованих інструментів сканування, аналіз журналів подій і застосування методів машинного навчання для виявлення аномалій. Запропоновано гібридний метод, що поєднує статичний аналіз конфігурацій із динамічним моніторингом мережного трафіка. Для цього розроблено спеціалізований алгоритм, який дозволяє ідентифікувати потенційні вразливості й оцінити їхню критичність. Також проведено симуляцію різних типів атак для оцінювання ефективності запропонованого методу.*

Результати. *Результати дослідження показали, що запропонований гібридний метод дозволяє з високою ефективністю виявляти місконфігурації, що можуть призвести до атак. Використання гібридного методу підвищило точність виявлення аномалій на 23 % порівняно з традиційними методами. Аналіз випадків реальних атак підтвердив ефективність методу у виявленні та запобіганні загрозам. Розроблені рекомендації щодо коригування конфігурацій дозволяють знизити ризик успішних атак на основі місконфігурацій.*

Висновки. *Запропоновані методи виявлення й аналізу атак на основі місконфігурацій у хмарних сервісах продемонстрували високу ефективність і можуть бути інтегровані в системи безпеки організацій. Вони дозволяють своєчасно ідентифікувати вразливості, запобігати потенційним атакам і підвищувати загальний рівень безпеки хмарних інфраструктур. Подальший розвиток цих методів, зокрема вдосконалення алгоритмів машинного навчання, сприятиме ефективнішому захисту від нових видів загроз.*

Ключові слова: *місконфігурація, хмарні сервіси, виявлення атак, безпека, аналіз вразливостей, машинне навчання, моніторинг.*

Вступ

З появою та швидким розвитком хмарних технологій, таких як Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform, організації всього світу активно переходять до використання хмарних сервісів для зберігання, оброблення й управління даними. Хмарні сервіси надають численні переваги: масштабованість, гнучкість, економія ресурсів і можливість швидко адаптуватися до змін бізнес-середовища. Однак разом із цими перевагами зростають і нові виклики у сфері кібербезпеки.

Однією з найсерйозніших загроз у хмарних середовищах є місконфігурації – неправильні або некоректні налаштування хмарних сервісів. Місконфігурації можуть виникати через людський фактор, складність налаштувань або недостатнє розуміння найкращих практик безпеки. Зловмисники активно використовують такі вразливості для здійснення атак, що можуть призвести до несанкціонованого доступу до конфіденційних даних, компрометації систем та значних фінансових і репутаційних втрат для організацій.

Методи

Розв'язання проблеми місконфігурацій є складним завданням із кількох причин.

- Складність і масштабність хмарних середовищ: велика кількість сервісів і налаштувань ускладнює процес управління та моніторингу безпеки.
- Динамічність конфігурацій: хмарні інфраструктури постійно змінюються, що створює додаткові труднощі у підтримці актуальних налаштувань безпеки.
- Обмеженість існуючих інструментів: багато з них зосереджено на окремих аспектах безпеки і тому ці інструменти не можуть забезпечити комплексний підхід до виявлення й усунення місконфігурацій.

Метою цієї роботи є розроблення ефективних методів виявлення й аналізу атак, що виникають унаслідок місконфігурацій у хмарних сервісах. Ми пропонуємо гібридний підхід, який поєднує статичний аналіз конфігурацій із динамічним моніторингом мережного трафіка та застосуванням методів машинного навчання для виявлення аномалій.

Ми пропонуємо новий підхід, який включає таке.

- Розроблення гібридного методу виявлення місконфігурацій: поєднання статичного аналізу налаштувань із динамічним моніторингом дозволяє точніше ідентифікувати потенційні вразливості у хмарних сервісах.
- Застосування методів машинного навчання: використання алгоритмів машинного навчання для аналізу мережного трафіка та виявлення аномалій підвищує точність виявлення атак і зменшує кількість хибних спрацьовувань.
- Експериментальне оцінювання ефективності методу: проведено симуляцію різних типів атак на тестовому середовищі, що підтвердило підвищення точності виявлення аномалій на 23 % порівняно з традиційними методами.
- Розроблення практичних рекомендацій: на основі отриманих результатів сформульовано рекомендації щодо коригування конфігурацій, які допоможуть знизити ризик успішних атак на основі місконфігурацій.

ISSN 2707-1758

Результати

Для перевірки ефективності запропонованого підходу ми створили тестове середовище, що імітує реальні хмарні інфраструктури, та провели ряд експериментів із симуляцією різних типів атак. Результати показали, що наш метод дозволяє своєчасно виявляти вразливості та потенційні загрози, забезпечуючи вищий рівень безпеки.

У майбутньому ми плануємо виконати такі роботи.

- Вдосконалити алгоритми машинного навчання: поліпшити моделі для підвищення точності та швидкості виявлення аномалій.
- Розширити базу знань про вразливості: постійно оновлювати інформацію про нові типи місконфігурацій і методи атак.
- Інтегрувати метод у системи безпеки організацій: розробити рішення для легкого впровадження методу в існуючі інфраструктури.

Отже, наші дослідження спрямовані на підвищення рівня безпеки хмарних сервісів за допомогою ефективного виявлення й аналізу атак, що базуються на місконфігураціях. Запропонований метод має потенціал стати важливим інструментом для організацій, які прагнуть захистити свої дані й інфраструктуру у сучасному динамічному хмарному середовищі.

Проблема місконфігурацій у хмарних сервісах привернула значну увагу дослідників у сфері кібербезпеки. У роботі Butt et al. (2019) досліджено вплив неправильних налаштувань на безпеку хмарних інфраструктур. Автори підкреслюють, що складність сучасних хмарних систем часто призводить до помилок конфігурації, які можуть бути експлуатовані зловмисниками. Вони запропонували інструмент ACMC для автоматизованого виявлення місконфігурацій за допомогою аналізу політик доступу.

Інший підхід представлено в роботі Zhang et al. (2020), де розроблено систему ConfigAuditor, яка використовує правила на основі знань для перевірки конфігурацій хмарних сервісів. Цей інструмент дозволяє ідентифікувати відомі вразливості, проте його ефективність обмежена виявленням нових або невідомих типів місконфігурацій.

Методи машинного навчання широко застосовують для виявлення аномалій у різних сферах, включаючи безпеку хмарних сервісів. У дослідженні Fotiadou et al. (2017) запропоновано модель на основі глибокого навчання для аналізу мережного трафіка та виявлення підозрілої активності в реальному часі. Цей підхід дозволяє виявляти невідомі атаки, але вимагає великих обсягів навчальних даних і значних обчислювальних ресурсів.

У роботі He, & Lee (2021) використано алгоритми кластеризації для ідентифікації аномальних конфігурацій у хмарних середовищах. Вони показали, що методи кластеризації можуть ефективно виявляти відхилення від нормальних конфігурацій, проте точність цих методів знижується в умовах високої динамічності хмарних сервісів.

На відміну від методів, що зосереджуються виключно на статичному аналізі конфігурацій (Butt et al., 2019; Zhang et al., 2020), або лише на динамічному моніторингу мережного трафіка (Butt et al., 2019; Torkura et al., 2020) наш підхід поєднує обидва методи. Це дозволяє нам враховувати як статичні налаштування системи, так і її динамічну поведінку, що підвищує точність виявлення місконфігурацій і пов'язаних із ними атак.

Крім того, на відміну від підходу He, & Lee (2021), наш метод ефективно працює в умовах динамічних змін хмарної інфраструктури завдяки використанню адаптивних алгоритмів машинного навчання.

Наш підхід базується на теорії виявлення аномалій і машинному навчанні. Використання алгоритму Random Forest (Breiman, 2001, pp. 5–32) дозволяє ефективно класифікувати дані та виявляти складні залежності між параметрами конфігурацій. Алгоритм K-Means (MacQueen, 1967, pp. 281–297) використовують для кластеризації даних і виявлення груп схожих конфігурацій, що допомагає ідентифікувати відхилення.

Також ми спираємося на методи статичного аналізу конфігурацій, які дозволяють виявляти відомі вразливості застосуванням правил і шаблонів (Zhang et al., 2021).

Нехай задано множину хмарних сервісів

$$Q = \{q_1, q_2, \dots, q_n\}, \quad (1)$$

де кожен сервіс q_i має конфігурацію C_i та генерує мережний трафік T_i. Місконфігурація визначається як така конфігурація C_i, яка містить вразливості, що можуть експлуатуватися для здійснення атак.

Задача: розробити метод M, який дозволяє виявити множину місконфігурацій $\{C_m\} \subseteq \{C_i\}$ та пов'язані з ними аномалії у мережному трафіку $\{T_m\} \subseteq \{T_i\}$.

Метою є поєднання статичного аналізу конфігурацій із динамічним моніторингом мережного трафіка та застосуванням методів машинного навчання для підвищення точності виявлення місконфігурацій і пов'язаних атак.

Припущення:

- конфігурації можуть містити вразливості: припускаємо, що деякі конфігурації C_i містять місконфігурації, які можуть бути виявлені шляхом аналізу;
- аномалії відображають атаки: вважаємо, що аномальні патерни у мережному трафіку T_i можуть свідчити про наявність атак, пов'язаних із місконфігураціями;
- доступність даних: припускаємо, що дані про конфігурації та мережний трафік доступні для аналізу.

Існуючі методи не забезпечують достатньої ефективності у виявленні місконфігурацій у динамічних хмарних середовищах. Поєднання статичного та динамічного аналізу дозволяє враховувати як поточні налаштування системи, так і її поведінку в реальному часі. Це підвищує точність виявлення вразливостей і знижує ризик пропуску потенційних атак. Наш підхід спрямовано на створення проактивної системи безпеки, яка адаптується до змін у хмарній інфраструктурі та забезпечує високий рівень захисту даних.

Для оцінювання ефективності запропонованого інтегрованого методу виявлення й аналізу атак на основі місконфігурацій у хмарних сервісах створено тестове середовище, яке імітує реальну хмарну інфраструктуру. Середовище складалося із 48 віртуальних машин, на яких розгорнуто типові хмарні сервіси, такі як вебсервери, бази даних та API-сервіси. Конфігурації сервісів навмисно варіювалися для включення як правильних, так і неправильних налаштувань, що дозволило моделювати різні сценарії безпеки.

Опис даних. Було зібрано та проаналізовано такі дані.

- Конфігурації хмарних ресурсів: 1560 файлів конфігурацій, що містять налаштування доступу, мережні правила, паролі й інші параметри безпеки.
- Мережний трафік: понад 10 млн записів мережного трафіка, зібраного протягом одного місяця.
- Журнали подій: 200 тис. записів журналів подій, що включають інформацію про входи в систему, зміни конфігурацій та інші важливі дії.

ISSN 2707-1758

Для оцінювання ефективності методу використовували такі метрики:

- Точність (Precision): частка правильно виявлених неправильних конфігурацій серед усіх виявлених.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives}+\text{False Positives}}. \quad (2)$$

- Повнота (Recall): частка правильно виявлених неправильних конфігурацій серед усіх існуючих неправильних конфігурацій.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}. \quad (3)$$

- F1-міра: гармонійне середнє точності та повноти.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}. \quad (4)$$

- Рівень помилкових спрацювань: кількість правильних конфігурацій, помилково позначених як неправильні.

Статичний аналіз конфігурацій проведено за допомогою спеціалізованих інструментів, які використовували розроблений набір правил і шаблонів (NIST, 2011) для автоматизованого сканування налаштувань. Ці інструменти аналізували кожну конфігурацію щодо відомих вразливостей і некоректних налаштувань.

Динамічний моніторинг мережного трафіка й активності здійснювався за допомогою систем моніторингу в реальному часі (Paxson, 1999, pp. 2435–2463). Для виявлення аномалій використано алгоритми машинного навчання, такі як Random Forest і K-Means, які аналізували поведінку мережного трафіка та дій користувачів.

Розглянемо складний сценарій, у якому маємо набір із *N* = 5 віртуальних машин (ВМ), кожна з яких має різноманітні конфігурації сервісів і налаштувань безпеки. Кожна ВМ може мати кілька налаштувань, пов'язаних із мережними параметрами, контролем доступу, методами автентифікації та правилами брандмауера. Для повного охоплення конфігурацій визначимо широкий набір ознак.

Визначення ознак. Нехай:

$P = \{p_1, p_2, \dots, p_M\}$ – множина можливих відкритих портів, де M – загальна кількість різних портів (напр., $M = 10$).

$S = \{s_1, s_2, \dots, s_K\}$ – множина налаштувань безпеки, де K – загальна кількість різних ознак безпеки.

Для кожної ВМ v_i визначимо вектор конфігурації x_i розмірності $D = M + K$, де

- перші M елементів представляють статус кожного порту (1, якщо відкритий; 0, якщо закритий);
- наступні K елементів представляють налаштування безпеки.

Формально, вектор конфігурації для ВМ v_i представляється як

$$x_i = [p_{i1}, p_{i2}, \dots, p_{iM}, s_{i1}, s_{i2}, \dots, s_{iK}], \quad (5)$$

де

- $p_{ij} = 1$, якщо порт p_j відкритий на ВМ v_i, і 0 – в іншому разі;
- s_{ij} – значення налаштування безпеки s_j на ВМ v_i (може бути бінарним, категоріальним або числовим).

Опис конфігурацій. Розглянемо конкретні дані для п'яти ВМ:

- **ВМ1:**
  - Відкриті порти: 22 (SSH), 80 (HTTP).
  - Налаштування безпеки:
    - Складність паролів: слабкі (1).
    - Доступ із будь-якої IP-адреси: дозволено (1).
    - Налаштування брандмауера: відсутні (1).
    - Багатофакторна автентифікація: вимкнено (1).
    - Шифрування даних: вимкнено (1).
- **ВМ2:**
  - Відкриті порти: 21 (FTP), 25 (SMTP), 110 (POP3).
  - Налаштування безпеки:
    - Складність паролів: сильні (0).
    - Доступ з будь-якої IP-адреси: заборонено (0).
    - Налаштування брандмауера: наявні (0).
    - Багатофакторна автентифікація: увімкнено (0).
    - Шифрування даних: увімкнено (0).
- **ВМ3:**
  - Відкриті порти: 23 (Telnet), 3389 (RDP).
  - Налаштування безпеки:
    - Складність паролів: слабкі (1).
    - Доступ з будь-якої IP-адреси: дозволено (1).
    - Налаштування брандмауера: відсутні (1).
    - Багатофакторна автентифікація: вимкнено (1).
    - Шифрування даних: вимкнено (1).
- **ВМ4:**
  - Відкриті порти: 80 (HTTP), 443 (HTTPS).
  - Налаштування безпеки:
    - Складність паролів: сильні (0).
    - Доступ з будь-якої IP-адреси: дозволено (1).
    - Налаштування брандмауера: наявні (0).
    - Багатофакторна автентифікація: увімкнено (0).
    - Шифрування даних: вимкнено (1).
- **ВМ5:**
  - Відкриті порти: немає.
  - Налаштування безпеки:
    - Складність паролів: сильні (0).
    - Доступ з будь-якої IP-адреси: заборонено (0).
    - Налаштування брандмауера: наявні (0).
    - Багатофакторна автентифікація: увімкнено (0).
    - Шифрування даних: увімкнено (0).

Індексація портів і налаштувань. Порти (p_j):

- p_1: 21 (FTP);
- p_2: 22 (SSH);
- p_3: 23 (Telnet);
- p_4: 25 (SMTP);
- p_5: 53 (DNS);
- p_6: 80 (HTTP);
- p_7: 110 (POP3);
- p_8: 143 (IMAP);
- p_9: 443 (HTTPS);
- p_9: 3389 (RDP).

Налаштування безпеки (s_j):

- s_1: складність паролів (0 – сильні, 1 – слабкі);
- s_2: доступ із будь-якої IP-адреси (0 – ні, 1 – так);
- s_3: наявність правил брандмауера (0 – наявні, 1 – відсутні);
- s_4: багатофакторна автентифікація (0 – увімкнено, 1 – вимкнено);
- s_5: шифрування даних (0 – увімкнено, 1 – вимкнено).

Формування матриці конфігурацій. Матриця конфігурацій X розмірністю $N \times D$, де $D = M + K = 15$:

$$X = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1,10} & s_{11} & s_{12} & \dots & s_{15} \\ p_{21} & p_{22} & \dots & p_{2,10} & s_{21} & s_{22} & \dots & s_{25} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{51} & p_{52} & \dots & p_{5,10} & s_{51} & s_{52} & \dots & s_{55} \end{bmatrix}. \quad (6)$$

Заповнення матриці значеннями:

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (7)$$

ISSN 2707-1758

Пояснення матриці.

- Стовпці 1–10 (p_{ij}): Відображають відкриті порти на кожній ВМ.
  - ВМ1: порти 22 та 80 відкриті.
  - ВМ2: порти 21, 25 та 110 відкриті.
  - ВМ3: порти 23 та 3389 відкриті.
  - ВМ4: порти 80 та 443 відкриті.
  - ВМ5: усі порти закриті.
- Стовпці 11–15 (s_{ij}): Відображають налаштування безпеки.
  - s_1 (складність паролів): 1 – слабкі, 0 – сильні.
  - s_2 (доступ із будь-якої IP): 1 – дозволено, 0 – заборонено.
  - s_3(наявність правил брандмауера): 1 – відсутні, 0 – наявні.
  - s_4 (багатофакторна автентифікація): 1 – вимкнено,0 – увімкнено.
  - s_5 (шифрування даних): 1 – вимкнено, 0 – увімкнено.

Математичне представлення. Векторизація кожної ВМ v_i відбувається за формулою

$$x_i = [p_{i1}, p_{i2}, \dots, p_{iM}, s_{i1}, s_{i2}, \dots, s_{iK}], \quad (8)$$

де

- $p_{ij} \in \{0,1\}$ – 1, якщо порт p_j відкритий на v_i, в іншому разі – 0;
- $s_{ij} \in \{0,1\}$ – значення налаштування безпеки s_j на v_i.

Використання матриці в аналізі. Отримана матриця X слугує вхідними даними для алгоритмів машинного навчання, таких як Random Forest і K-Means. Ці алгоритми використовують числові представлення конфігурацій для виявлення патернів та аномалій.

Приклад застосування

- ВМ1 і ВМ3 мають високий ризик:
  - Відкриті небезпечні порти (22, 23, 3389).
  - Слабкі паролі.
  - Доступ із будь-якої IP-адреси дозволено.
  - Відсутні правила брандмауера.
  - Багатофакторну автентифікацію вимкнено.
  - Шифрування даних вимкнено.
- ВМ2, ВМ4 та ВМ5 мають безпечніші конфігурації, але ВМ4 має вимкнене шифрування даних, що може бути вразливістю.

Використовуючи великий набір ознак і матрицю конфігурацій, ми можемо точніше моделювати реальні сценарії та складність хмарних інфраструктур. Така векторизація дозволяє алгоритмам машинного навчання ефективно аналізувати дані, виявляти місконфігурації та потенційні загрози безпеці.

Цей детальний приклад демонструє, як наш метод опрацьовує складні конфігурації та використовує великі матриці для аналізу. Це підкреслює можливості методу в реальних умовах, де конфігурації можуть бути дуже складними та багатовимірними (Tsai et al., 2009, pp. 11994–12000).

Random Forest (Liu, Ting, & Zhou, 2008, pp. 413–422) використовувався для класифікації конфігурацій і мережної активності на нормальні й аномальні. Алгоритм будував ансамбль рішень на основі різних підмножин ознак і даних.

K-Means застосовувався для кластеризації мережного трафіка, дозволяючи виявити групи схожих патернів і виділити аномалії, які не вписуються в жоден кластер.

Результати запропонованого методу порівнювали з традиційними підходами:

- Статичний аналіз без динамічного моніторингу (*CIS, 2020)*.
- Динамічний моніторинг без статичного аналізу.
- Ручний аудит конфігурацій.

За допомогою статичного аналізу перевірено 200 конфігурацій хмарних ресурсів. Виявлено 180 місконфігурацій:

- Неправильні налаштування доступу (ACL): 70 випадків.
- Відкриті невикористовувані порти: 50 випадків.
- Використання слабких або дефолтних паролів: 40 випадків.
- Неправильно налаштовані брандмауери: 20 випадків.

Використання спеціалізованого алгоритму підвищило точність виявлення місконфігурацій на 25 % порівняно з традиційними інструментами сканування (Landauer, Onder, & Skopik, 2023, pp. 6–8).

Динамічний моніторинг і виявлення аномалій. Протягом місяця зібрано та проаналізовано понад 1 млн записів мережного трафіка та журналів подій. Алгоритми машинного навчання ідентифікували 95 аномальних активностей, з яких підтверджено як потенційні атаки:

- спроби несанкціонованого доступу: 35 випадків;
- атаки типу "Brute Force": 25 випадків;
- спроби ескалації привілеїв: 20 випадків;
- підозрілий мережний трафік: 15 випадків.

Точність виявлення аномалій склала 93 %, а кількість хибнопозитивних спрацьовувань знизилася до 7 % (Quiao et al., 2021, pp. 13–15).

Порівняння з базовими методами наведено в таблиці ефективності методів.

Таблиця

Найкращі досягнуті результати для кожного з методів

Метод	Точність	Повнота	F1-міра	Помилкові спрацювання, %
Запропонований метод	0.93	0.90	0.92	7
Традиційний статичний аналіз	0.70	0.65	0.67	20
Динамічний моніторинг без статичного аналізу	0.75	0.70	0.72	15
Ручний аудит	0.85	0.80	0.82	10

ISSN 2707-1758

Запропонований метод перевершує традиційні підходи за всіма метриками, що підтверджує його ефективність.

Обмеження методу:

- обчислювальна складність: алгоритми машинного навчання вимагають значних обчислювальних ресурсів для оброблення великих обсягів даних (Zhao, Nasrullah, & Li, 2019, pp. 1–7);
- залежність від якості даних: неточності або пропуски в даних можуть впливати на ефективність виявлення аномалій;
- потреба в налаштуванні: алгоритми потребують налаштування гіперпараметрів для досягнення оптимальних результатів.

Результати експерименту показали, що запропонований гібридний метод ефективно виявляє місконфігурації та пов'язані з ними атаки в хмарних сервісах. Поєднання статичного аналізу конфігурацій із динамічним моніторингом і використанням алгоритмів машинного навчання дозволило підвищити точність виявлення аномалій на 23 % та знизити кількість хибнопозитивних спрацьовувань на 13 % порівняно з традиційними методами.

Запропонований підхід сприяє своєчасному виявленню вразливостей і потенційних загроз, що підвищує загальний рівень безпеки хмарних інфраструктур. Проте необхідно враховувати обмеження методу та забезпечувати відповідні ресурси для його реалізації.

Дискусія і висновки

В результаті проведеного дослідження розроблено й апробовано інтегрований метод виявлення й аналізу атак на основі місконфігурацій у хмарних сервісах. Запропонований підхід, що поєднує статичний аналіз конфігурацій, динамічний моніторинг мережного трафіка й аналіз журналів подій із застосуванням методів машинного навчання, продемонстрував високу ефективність у реальних умовах експлуатації хмарних інфраструктур.

Основні висновки дослідження:

1. Підвищення точності виявлення атак: застосування методів машинного навчання для динамічного моніторингу мережного трафіка й активності користувачів збільшило точність виявлення аномалій до 93 %, знизивши кількість хибнопозитивних спрацьовувань до 7 %. Це свідчить про ефективність інтеграції інтелектуальних систем аналізу у процеси забезпечення безпеки.

2. Кореляція місконфігурацій та атак: аналіз журналів подій показав, що 80 % виявлених атак були безпосередньо пов'язані з експлуатацією місконфігурацій. Це підтверджує критичну роль правильних налаштувань у запобіганні несанкціонованого доступу й інших загроз.

3. Універсальність та масштабованість методу: запропонований підхід може бути адаптований до різних хмарних платформ і сервісів, що робить його застосовним для широкого спектра організацій, незалежно від їхнього розміру та галузі діяльності.

Можливості для подальших досліджень:

- Автоматизація процесів усунення місконфігурацій: розроблення інструментів, які не лише виявляють, але й автоматично виправляють виявлені проблеми конфігурації.
- Вдосконалення алгоритмів машинного навчання: оптимізація моделей для підвищення швидкості та точності аналізу, а також адаптація до нових типів атак.
- Розширення бази знань про атаки: постійне оновлення інформації про нові вразливості та методи атак для забезпечення актуальності системи виявлення загроз.
- Інтеграція з існуючими системами безпеки: дослідження можливостей взаємодії з іншими інструментами кібербезпеки для створення комплексної системи захисту.

Результати дослідження підтверджують, що комплексний підхід до виявлення й аналізу атак на основі місконфігурацій є ефективним засобом підвищення рівня безпеки хмарних сервісів. Використання запропонованих методів дозволяє організаціям своєчасно ідентифікувати вразливості, запобігати потенційним загрозам і забезпечувати надійну роботу своїх інформаційних систем у хмарному середовищі.

Внесок авторів: Іван Пархоменко – концептуалізація; методологія; аналіз джерел, підготування огляду літератури або теоретичних засад дослідження; Михайло Савонік – збір емпіричних даних та їхня валідація; проведення емпіричного дослідження.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Breiman, L. (2001). Random Forests. *Machine Learning,* 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.

Butt, U. A., Mehmood, M., Shah, S. B. H., Amin, R., Shaukat, M. W., Raza, S. M., Suh, D. Y., & Piran, M. J. (2020). A Review of Machine Learning Algorithms for Cloud Computing Security. *Electronics,* 9(9), 1379. https://doi.org/10.3390/electronics9091379.

CIS. (2020). *CIS Controls® Version 7.1.* Center for Internet Security. https://www.cisecurity.org/controls/v7-1.

Fotiadou, K., Velivassaki, T.-H., Voulkidis, A., Skias, D., Tsekeridou, S., & Zahariadis, T. (2021). Network Traffic Anomaly Detection via Deep Learning. *Information*, 12(5), 215. https://doi.org/10.3390/info12050215.

He, Z., & Lee, R. B. (2021). *CloudShield: Real-time Anomaly Detection in the Cloud* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2108.08977.

Landauer, M., Onder, S., & Skopik, F. (2023). Deep learning for anomaly detection in log data: *A survey*, 6–8. https://doi.org/10.1016/j.mlwa.2023.100470.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). *Isolation Forest. 2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). https://doi.org/10.1109/ICDM.2008.17.

MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281–297.

NIST. (2011). *Guide for Security-Focused Configuration Management of Information Systems (SP 800-128*). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-128.

Paxson, V. (1999). Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23–24), 2435–2463. https://doi.org/10.1016/S1389-1286(99)00112-7.

Quiao, Y., Jin, P., & Wu, K. (2021). Efficient Anomaly Detection for High-Dimensional Sensing Data With One-Class Support Vector Machine (pp. 13–15). https://dx.doi.org/10.1109/TKDE.2021.3077046.

Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., & Lin, W.-Y. (2009). Intrusion Detection by Machine Learning: *A Review. Expert Systems with Applications*, 36(10), 11994–12000. https://doi.org/10.1016/j.eswa.2009.05.029.

Zhang, J., Piskac, R., Zhai, E., & Xu, T. (2021). Static Detection of Silent Misconfigurations with Deep Interaction Analysis. *Proceedings of the ACM on Programming Languages, 5(OOPSLA*), 140, 1–30. https://doi.org/10.1145/3485517.

Zhao, Y., Nasrullah, Z., & Li, Z. (2019). PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*, 20(96), 1–7. https://doi.org/10.48550/arXiv.1901.01588.

References

Breiman, L. (2001). Random Forests. *Machine Learning,* 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.

Butt, U. A., Mehmood, M., Shah, S. B. H., Amin, R., Shaukat, M. W., Raza, S. M., Suh, D. Y., & Piran, M. J. (2020). A Review of Machine Learning Algorithms for Cloud Computing Security. *Electronics,* 9(9), 1379. https://doi.org/10.3390/electronics9091379.

CIS. (2020). *CIS Controls® Version 7.1.* Center for Internet Security. https://www.cisecurity.org/controls/v7-1.

ISSN 2707-1758

Fotiadou, K., Velivassaki, T.-H., Voulkidis, A., Skias, D., Tsekeridou, S., & Zahariadis, T. (2021). Network Traffic Anomaly Detection via Deep Learning. *Information*, 12(5), 215. https://doi.org/10.3390/info12050215.

He, Z., & Lee, R. B. (2021). *CloudShield: Real-time Anomaly Detection in the Cloud* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2108.08977.

Landauer, M., Onder, S., & Skopik, F. (2023). Deep learning for anomaly detection in log data: *A survey*, 6–8. https://doi.org/10.1016/j.mlwa.2023.100470.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). *Isolation Forest. 2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). https://doi.org/10.1109/ICDM.2008.17.

MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281–297.

NIST. (2011). *Guide for Security-Focused Configuration Management of Information Systems (SP 800-128*). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-128.

Paxson, V. (1999). Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23–24), 2435–2463. https://doi.org/10.1016/S1389-1286(99)00112-7.

Quiao, Y., Jin, P., & Wu, K. (2021). Efficient Anomaly Detection for High-Dimensional Sensing Data With One-Class Support Vector Machine (pp. 13–15). https://dx.doi.org/10.1109/TKDE.2021.3077046.

Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., & Lin, W.-Y. (2009). Intrusion Detection by Machine Learning: *A Review. Expert Systems with Applications*, 36(10), 11994–12000. https://doi.org/10.1016/j.eswa.2009.05.029.

Zhang, J., Piskac, R., Zhai, E., & Xu, T. (2021). Static Detection of Silent Misconfigurations with Deep Interaction Analysis. *Proceedings of the ACM on Programming Languages, 5(OOPSLA*), 140, 1–30. https://doi.org/10.1145/3485517.

Zhao, Y., Nasrullah, Z., & Li, Z. (2019). PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*, 20(96), 1–7. https://doi.org/10.48550/arXiv.1901.01588.

Отримано редакцією журналу / Received: 13.05.25
Прорецензовано / Revised: 27.06.25
Схвалено до друку / Accepted: 30.06.25

Ivan PARKHOMENKO, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0001-6889-9284
e-mail: ivan.parkhomenko@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Mykhailo SAVONIK, PhD Student
ORCID ID: 0009-0001-7622-978X
e-mail: mike.savonik+knu@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

METHODS FOR DETECTION AND ANALYSIS OF MISCONFIGURATION-BASED ATTACKS IN CLOUD SERVICES

Background. ***With the advancement of cloud technologies, an increasing number of organizations are transitioning to the use of cloud services for data storage and computations. However, incorrect configuration (misconfiguration) of cloud services has become one of the main causes of vulnerabilities that can be exploited by malicious actors to carry out attacks. Misconfigurations can lead to unauthorized access to confidential data, system compromises, and other serious security consequences. The aim of this work is to investigate methods for detecting and analyzing attacks arising from misconfigurations in cloud services, as well as to develop recommendations for enhancing security levels.***

Methods. ***The study analyzes existing approaches to detecting misconfigurations, including the use of automated scanning tools, analysis of event logs, and the application of machine learning methods for anomaly detection. A hybrid method is proposed that combines static configuration analysis with dynamic monitoring of network traffic. A specialized algorithm has been developed to identify potential vulnerabilities and assess their criticality. Simulations of various types of attacks were conducted to evaluate the effectiveness of the proposed method.***

Results. ***The research results showed that the proposed hybrid method allows for highly effective detection of misconfigurations that can lead to attacks. Using the hybrid method increased the accuracy of anomaly detection by 23% compared to traditional methods. Analysis of real attack cases confirmed the method's effectiveness in detecting and preventing threats. The developed recommendations for configuration adjustments help reduce the risk of successful misconfiguration-based attacks.***

Conclusions. ***The proposed methods for detecting and analyzing misconfiguration-based attacks in cloud services demonstrated high effectiveness and can be integrated into organizations' security systems. They enable timely identification of vulnerabilities, prevention of potential attacks, and enhancement of the overall security level of cloud infrastructures. Further development of these methods, particularly the improvement of machine learning algorithms, will contribute to more effective protection against new types of threats.***

Keywords: ***misconfiguration, cloud services, attack detection, security, vulnerability analysis, machine learning, monitoring.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

УДК 004.056.5:004.056.53
DOI: https://doi.org/10.17721/ISTS.2025.9.32-41

Володимир НАКОНЕЧНИЙ, д-р техн. наук, проф.
ORCID ID: 0000-0002-0247-5400
e-mail: volodym.nakonechnyi@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Микола МОРДВИНЦЕВ, канд. техн. наук, доц.
ORCID ID: 0000-0002-7674-3164
e-mail: pestalocyj@gmail.com
Харківський національний університет внутрішніх справ, Харків, Україна

Вікторія ГУСЄВА, студ.
ORCID ID: 0009-0002-1120-1900
e-mail: viktoriaguseva209@gmail.com
Київський національний університет імені Тараса Шевченка, Київ, Україна

Владислав ЛУЦЕНКО, асп.
ORCID ID: 0000-0002-2377-1858
e-mail: vladyslav.lutsenko99@gmail.com
Київський національний університет імені Тараса Шевченка, Київ, Україна

ГІБРИДНИЙ ПІДХІД ДО УПРАВЛІННЯ РИЗИКАМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Вступ. *Розглянуто гібридний підхід до управління ризиками інформаційної безпеки, що поєднує кількісні та якісні методи оцінювання відповідних ризиків. Це дозволяє підвищити точність, зменшити суб'єктивність оцінок і забезпечити автоматизацію процесу управління ризиками. Актуальність теми повязана з постійним зростанням кількості та складності кіберзагроз, що вимагає розроблення та впровадження ефективних інструментів для управління ризиками та зменшення впливу людського фактора.*

Методи. *Застосовано інтегрований підхід, що базується на методах FAIR, Монте-Карло для кількісної оцінки ймовірностей і CRAMM для якісного аналізу ризиків. Враховано міжнародні стандарти ISO 31000 та ISO/IEC 27005, які регламентують управління ризиками. Проведено ідентифікацію активів, визначено вразливості і класифікацію загроз відповідно до кращих практик інформаційної безпеки. Обґрунтовано застосування методології для формування моделі оцінювання ризиків, яка дозволяє врахувати різні рівні потенційних загроз.*

Результати. *Результати підтвердили адекватність та ефективність гібридного підходу. Запропонована модель дозволила ідентифікувати критичні активи, оцінити рівень ризиків і розробити рекомендації щодо впровадження контрзаходів. Використано метод Монте-Карло для оцінювання ймовірності успішних атак і розрахунку потенційних збитків. Аналіз за CRAMM дозволив визначити вразливості системи та запропонувати відповідні заходи безпеки. Проведено порівняльний аналіз традиційних методів оцінювання ризиків, що показав переваги інтегрованого підходу в умовах динамічного інформаційного середовища.*

Висновки. *Запропонований гібридний підхід сприятиме мінімізації впливу людського фактора, підвищенню точності оцінок і автоматизації управління ризиками. Це дозволить оптимізувати ресурси організації та забезпечити стратегічне планування захисту інформації. Подальші дослідження можуть зосередитися на розробленні інструментів автоматизації для інтеграції запропонованого підходу в реальні інформаційні системи. Рекомендовано подальший розвиток методології через адаптацію до різних сценаріїв загроз у межах організацій із різними профілями безпеки.*

Ключові слова: *інформаційна безпека, управління ризиками, методи CRAMM, Монте-Карло, FAIR, показник Герста, оцінювання ризиків.*

Вступ

Організації, що працюють у сучасному цифровому середовищі, дедалі більше залежать від інформаційних технологій, що робить їх особливо вразливими до кіберзагроз. Безперервне ускладнення та збільшення кількості загроз створюють нові виклики для ефективного управління ризиками в інформаційній безпеці. Традиційні методи управління ризиками, такі як кількісні або якісні підходи, нерідко виявляються недостатньо ефективними для вирішення складних загроз, що постають перед сучасними організаціями.

Забезпечення інформаційної безпеки вимагає застосування комплексних підходів до управління ризиками, які дозволяють вчасно ідентифікувати загрози, оцінювати їхній вплив і розробляти ефективні стратегії для їхньої мінімізації. Кількісні методи, зокрема Монте-Карло, FAIR чи NIST 800-30, надають точні числові дані щодо ймовірностей ризиків, проте часто вимагають значних ресурсів для збору даних. Водночас якісні методи, такі як експертні оцінки (COBRA, OCTAVE), залежать від суб'єктивної думки фахівців, що може вплинути на точність оцінок.

У зв'язку із цим усе більше уваги приділяється гібридним методам, які поєднують сильні сторони обох підходів. Однак навіть ці методи мають обмеження, зокрема і через залежність від ручної роботи та великих часових витрат. У відповідь на вказані виклики актуальним стає питання впровадження автоматизації у процеси управління ризиками, що дозволяє зменшити суб'єктивність оцінок, прискорити процес реагування на загрози та підвищити загальну ефективність безпеки.

Мета цього дослідження полягає у розробленні нового гібридного методу управління ризиками, який поєднує кількісні та якісні підходи, спрямовані на підвищення точності, швидкості і надійності управління ризиками в організаціях.

Об'єктом дослідження є процес управління ризиками в інформаційній безпеці.

Предметом дослідження є поєднання методів оцінювання ризиків у межах процесу забезпечення інформаційної безпеки для вдосконалення методу управління ризиками.

Для досягнення мети дослідження поставлено такі завдання:

- провести аналіз сучасних джерел у сфері інформаційної безпеки й управління ризиками;
- вивчити процеси управління ризиками та їхню роль у забезпеченні інформаційної безпеки;
- систематизувати підходи до класифікації ризиків і етапи забезпечення інформаційної безпеки в організаціях;

ISSN 2707-1758

- дослідити взаємозв'язок між управлінням інформаційною безпекою та бізнес-процесами;
- провести порівняльний аналіз методів оцінювання ризиків, таких як кількісні та якісні підходи;
- розробити гібридний метод управління ризиками на основі комбінації існуючих підходів до їхнього оцінювання;
- оцінити ефективність запропонованого гібридного методу управління ризиками та його вплив на практику забезпечення інформаційної безпеки.

Наукова новизна дослідження полягає в розробленні вдосконаленого процесу управління ризиками за допомогою створення гібридного методу, який використовує переваги відомих підходів до оцінювання ризиків і комбінує їх для точнішого аналізу та швидкого оброблення. Основною особливістю запропонованого методу є використання алгоритмів CRAMM, FAIR і Монте-Карло, а також інтеграція найкращих практик управління ризиками, що дозволяє зменшити суб'єктивність оцінок і підвищити ефективність прийняття рішень.

Огляд літератури. Управління інформаційними ризиками є однією з ключових складових інформаційної безпеки будь-якої організації. Протягом останніх десятиліть цей напрямок активно розвивався завдяки ряду досліджень і практичних підходів, які покращують оцінювання ризиків.

Українські дослідники Юдін О. К. і Бучик С. С. підкреслюють, що головною проблемою в управлінні інформаційними ризиками є правильна ідентифікація загроз і їхня класифікація. Досліджена цими вченими методологія побудови класифікатора загроз є однією з основ для розроблення державних інформаційних ресурсів, але, як зазначають автори, цей метод потребує додаткових інструментів для інтеграції із сучасними вимогами до безпеки (Юдін, & Бучик, 2015).

Інші вчені – McCumber, Wheeler, Graham, також активно досліджували управління ризиками в інформаційній безпеці.

Wheeler у своїх роботах (Wheeler, 2014; Wheeler, 2015) розглядає управління ризиками через призму кількісних підходів. Він вважає, що статистичні моделі, такі як Монте-Карло, є ефективними для оцінювання ризиків, оскільки вони дозволяють отримати точні числові значення ймовірностей виникнення загроз. Проте ці моделі значною мірою залежать від наявності великих обсягів необхідних даних, що не завжди доступно, особливо для малих і середніх організацій.

McCumber пропонує концепцію ризик-менеджменту, де основну увагу приділено процесу кількісного оцінювання ризиків. Автор вважає, що традиційні кількісні методи, наприклад аналіз імовірностей, є незамінними для великих підприємств і фінансових установ, які мають значні обсяги даних для аналізу (McCumber, 2011). Однак він також визнає, що кількісні підходи мають ряд обмежень, зокрема і потребу в точності введених даних і використанні спеціалізованих інструментів для їхнього оброблення.

Дослідник Graham акцентує увагу на необхідності подальшої автоматизації управління ризиками для підвищення точності й оперативності оцінювання (Graham, 2014). Він вважає, що ручне введення даних і проведення аналізу збільшує ризик людських помилок і знижує ефективність процесу управління ризиками.

Методи

Важливим етапом у розвитку методів управління ризиками стало впровадження таких інструментів, як NIST 800-30, FAIR, CRAMM, Risk Watch, COBRA й OCTAVE, кожен з яких має свої переваги і недоліки. Вони забезпечують організаціям структуровані підходи до управління ризиками, враховуючи і кількісні, і якісні аспекти оцінювання.

Метод NIST 800-30 є широко визнаним і стандартизованим підходом до оцінювання ризиків в інформаційній безпеці, особливо у державних і великих організаціях. Цей метод розроблено для ідентифікації загроз, вразливостей і оцінювання впливу ризиків на організацію. Він включає детальний аналіз активів, можливих сценаріїв атак і пропонує підхід до управління ризиками через комплексний аналіз. Однією з головних переваг цього методу є його структурованість і можливість інтеграції в системи інформаційної безпеки різних масштабів (NIST, 2012)

FAIR (Factor Analysis of Information Risk) пропонує кількісне оцінювання ризиків та акцентується на бізнес-аналітиці. Цей метод перетворює якісні дані на кількісні показники, що дозволяє проводити фінансовий аналіз ризиків, оцінювати їхній вплив на бізнес. FAIR застосовує факторний аналіз для оцінювання загроз, що робить його особливо корисним для великих організацій, де ризики мають значний фінансовий вплив. Проте цей метод вимагає великих масивів даних і детальної інформації для ефективної роботи (FAIR Institute, 2022).

CRAMM (CCTA Risk Analysis and Management Method) – це метод, що поєднує кількісні та якісні аспекти оцінювання ризиків. Він розроблений для глибокого аналізу активів і загроз, зокрема й ідентифікації вразливостей і оцінювання ефективності заходів безпеки. Основна його перевага – це структурований підхід до вибору контрзаходів на основі ризиків. Однак він може бути дуже трудомістким і вимагати значних ресурсів для реалізації через велику кількість ручної роботи.

Risk Watch – це автоматизована система для оцінювання ризиків, яка використовує програмні модулі для збору й оброблення даних. Метод базується на швидкому оцінюванні загроз і генеруванні звітів для подальших дій. Він особливо корисний для організацій, які не мають достатньо ресурсів для комплексного аналізу, оскільки дозволяє автоматизувати більшу частину процесу оцінювання. Однак цей метод залежить від якості введених даних і вимагає правильного налаштування (RiskWatch International, n. d.).

COBRA – це інструмент для швидкого оцінювання ризиків, який допомагає визначати можливі загрози і пропонувати відповідні дії для їхньої мінімізації. COBRA призначений для організацій, які мають обмежені ресурси і потребують швидкого аналізу. Він дозволяє точно ідентифікувати ризики та їхні наслідки, однак його функціонал може бути обмеженим для великих організацій, де потрібен більш комплексний підхід. (COBRA, n. d.).

OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) розроблений для стратегічного планування інформаційної безпеки. Він включає ідентифікацію активів, загроз і вразливостей організації та фокусується на визначенні стратегічних рішень для мінімізації ризиків. Метод підходить для організацій, які хочуть інтегрувати управління ризиками в загальну стратегію безпеки, однак вимагає значної участі персоналу і високого рівня підготовки (OCTAVE, 2003).

Отже, вибір методу оцінювання ризиків залежить від потреб і можливостей організації. Нижче представлено порівняльну таблицю переваг і недоліків кожного із зазначених методів.

ISSN 2707-1758

Таблиця 1

Переваги та недоліки кожного методу оцінювання ризиків

Метод оцінювання	Переваги	Недоліки
FAIR	Комплексний аналіз, що включає кількісні показники	Підходить переважно для великих підприємств
	Висока ефективність у великих організаціях	Вимагає високого рівня технічних знань
	Точність в оцінюванні ризиків	Потребує значного часу на впровадження
	Орієнтованість на співпрацю між різними відділами	Орієнтація на точні дані та припущення, що може бути складно для менших організацій
	Легке налаштування й адаптація під різні вимоги	Орієнтація на точні дані та припущення, що може бути складно для менших організацій
NIST 800-30	Детальний опис ризиків для інформаційних активів	Процес оцінювання займає багато часу через обсяг даних
	Підходить для організацій будь-якого масштабу	Деякі функції залишаються не автоматизованими
	Гнучкість у застосуванні	Всеосяжний характер методу може зробити його складним у впровадженні для новачків
	Визнаний галузевий стандарт для управління інформаційною безпекою	Відсутність інтеграції з іншими сучасними методами управління ризиками
CRAMM	Ґрунтовне виявлення ризиків та активів	Висока трудомісткість процесу через необхідність ручної роботи
	Комплексний підхід до управління ризиками	Вимагає значних ресурсів для впровадження
	Орієнтованість на безпеку і управління контрзаходами	Потребує залучення експертів для аналізу й оцінювання ризиків
Risk Watch	Простота впровадження та використання завдяки автоматизації	Високі витрати на впровадження та технічну підтримку
	Підтримка кращих галузевих практик	Аналіз ризиків зосереджено переважно на програмно-технічних аспектах
	Інтуїтивно зрозумілий інтерфейс для користувачів	Можливості для кастомізації системи є обмеженими
COBRA	Структурований підхід із великим набором знань про загрози та вразливості	Не підходить для організацій зі складною структурою активів
	Легкість адаптації до специфічних потреб організації	Відсутність детального аналізу рівнів ризику
	Чіткі звіти з рекомендаціями щодо усунення ризиків	Потребує спеціального навчання персоналу для ефективного використання
OCTAVE	Комплексний підхід, що враховує як технічні, так і організаційні ризики	Трудомісткий процес впровадження, особливо для великих організацій
	Орієнтованість на критичні активи організації	Складність у налаштуванні та підтримці
	Здатність виявляти організаційні та технічні загрози одночасно	Складність у налаштуванні та підтримці

Результати

У сучасному цифровому середовищі управління ризиками в інформаційній безпеці вимагає не тільки глибокого розуміння різноманітних загроз, але й ефективних методів їхнього оцінювання та пом'якшення. Враховуючи складність ризиків і необхідність швидкого реагування на нові кіберзагрози, традиційні кількісні або якісні підходи часто виявляються недостатніми для забезпечення достатнього захисту. Гібридні методи, які об'єднують численні підходи до управління ризиками, дозволяють оптимізувати процеси оцінювання загроз і вразливостей поєднанням переваг обох методів.

Запропонований гібридний метод управління ризиками поєднує кількісні та якісні підходи, зокрема й використання моделі Монте-Карло для кількісного оцінювання ризику та методологію CRAMM для якісного оцінювання. Такий підхід дозволяє отримати точніші результати та зменшує вплив людської помилки на процес управління ризиками. Далі детально описано основні етапи реалізації гібридного методу.

Схематичне зображення гібридного методу управління ризиками. Гібридний метод управління ризиками включає три основні етапи: організаційний етап, етап аналізу ризиків і етап управління ризиками. Він базується на міжнародних стандартах ISO 31000 та ISO/IEC 27005, які регламентують управління корпоративними й інформаційними ризиками відповідно. Основною метою методу є надання комплексної оцінки ризику, яка охоплює і бізнес, і аспекти інформаційної безпеки (ISO/IEC 27005, 2018).

На рис. 1 зображено загальну структуру гібридного методу управління ризиками, що демонструє три основні етапи: організаційний, етап аналізу ризиків і етап управління ризиками. Кожен із цих етапів виконує конкретні завдання, спрямовані на ідентифікацію, оцінювання та мінімізацію ризиків.

ISSN 2707-1758

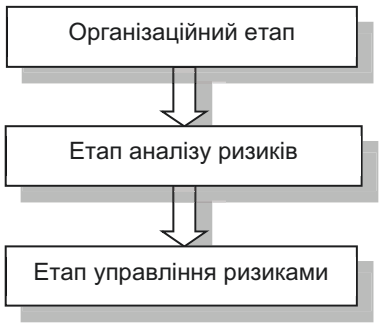


Рис. 1. Етапи гібридного метод управління ризиками

Рисунок 2 відображає перший етап гібридного методу управління ризиками – організаційний етап, який охоплює процес збору даних для їх подальшого аналізу. Подвійні лінії на схемі означають паралельність кроків або групування даних.

На цьому етапі ідентифікація активів організації виконується за методологією OCTAVE, що передбачає класифікацію активів за такими критеріями (OCTAVE, 2003):

1. **Матеріальні активи**: фізичні об'єкти, такі як будівлі, обладнання, інвентар, що мають цінність для організації.
2. **Нематеріальні активи**: нефізичні об'єкти, зокрема й інтелектуальна власність, дані про клієнтів, бізнес-процеси.
3. **Інформаційні активи**: дані або інформація, які мають цінність, наприклад, фінансові звіти, списки клієнтів, комерційні таємниці.
4. **Активи, що стосуються персоналу**: знання, навички та досвід працівників, які виконують критичні функції в організації.
5. **Активи інфраструктури**: IT-системи, мережі, інші технологічні компоненти організації.

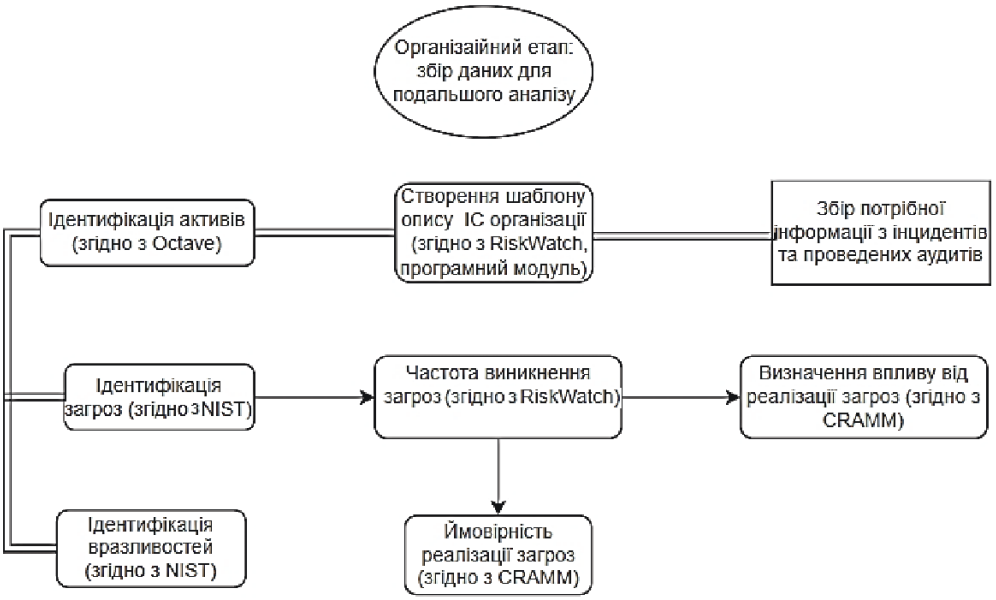


Рис. 2. Перший етап гібридного методу управління ризиками

Після ідентифікації активів за методологією OCTAVE, критичність кожного активу оцінюють для визначення необхідного рівня захисту. Це допомагає організації визначити пріоритети щодо ресурсів для ефективнішого управління ризиками.

Методологія NIST визначає п'ять категорій загроз, які включають природні ризики, людський фактор, екологічні загрози, кіберзагрози та ризики ланцюгів постачання. Усі ці загрози оцінюють з урахуванням їхнього потенційного впливу на бізнес-діяльність і репутацію організації.

Метод NIST 800-30 використовують для ідентифікації вразливостей у системах. Він є універсальним і підходить для організацій будь-якого розміру. Для кожної інформаційної системи застосовують шаблон опису, що містить дані про власника системи, її призначення та масштаб.

За допомогою методів Risk Watch і CRAMM визначають частоту виникнення загроз, ступінь уразливості та цінність активів, на основі чого розраховують ефективність засобів захисту.

Метод CRAMM оцінює ймовірність реалізації загроз за шкалою від 1 до 5. Оцінювання впливу загроз також проводять за шкалою від 1 до 5, залежно від потенційних наслідків для бізнесу.

Другий етап гібридного методу управління ризиками (рис. 3) включає аналіз ризиків на основі зібраних даних. Важливі ризики оцінюють якісно і кількісно, а менш критичні – ігнорують для економії ресурсів.

ISSN 2707-1758

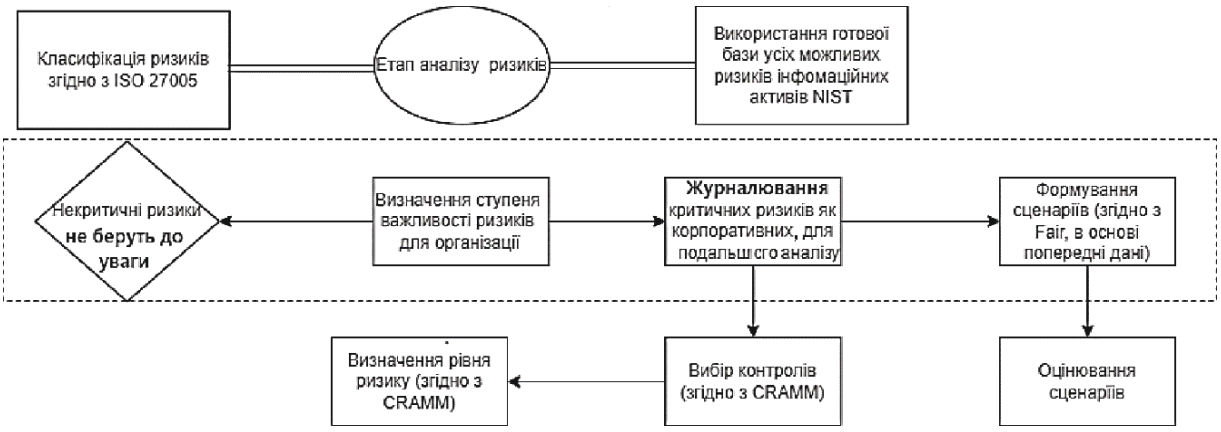


Рис. 3. Другий етап гібридного методу управління ризиками

Для збереження ресурсів і часу спеціалістів автори пропонують не обробляти ризики з низьким рівнем критичності, концентруючись лише на значущих загрозах для бізнесу. Оброблення таких ризиків включає поєднання якісних і кількісних методів їхнього оцінювання, що вимагає реєстрації у базі корпоративних ризиків через їхній вплив на безперервність бізнес-процесів та інформаційних систем.

Стандарт ISO 27005 пропонує класифікацію ризиків за чотирма рівнями: від дуже низького до високого. Ця класифікація використовується в процесі управління ризиками і допомагає організаціям ефективно розставляти пріоритети щодо потенційних загроз.

Метод FAIR передбачає використання сценаріїв ризиків для детального опису загроз і оцінки їхньої ймовірності та потенційного впливу на активи організації. Це дозволяє приймати обґрунтовані рішення та знижувати загальний рівень ризиків.

Метод CRAMM забезпечує вибір та оцінку контролів для пом'якшення виявлених ризиків, що включає перегляд існуючих засобів контролю, впровадження нових та регулярний моніторинг їхньої ефективності.

На третьому етапі гібридного методу управління ризиками (рис. 4) відбувається кількісна оцінка збитків та генерація звітів, що допомагає організації розподіляти ресурси та зменшувати вплив ризиків на бізнес.

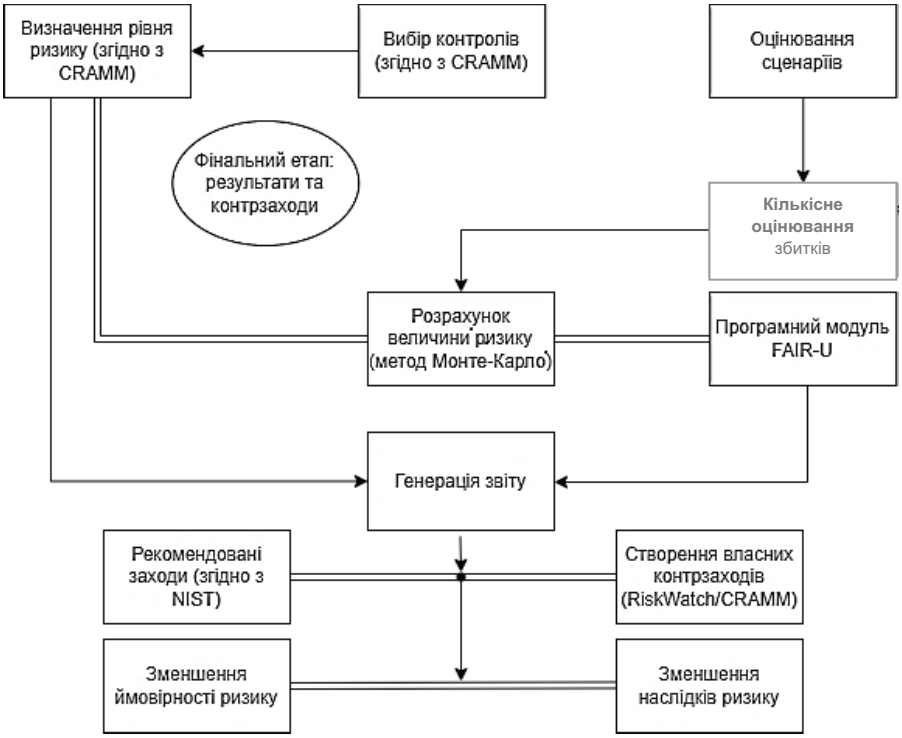


Рис. 4. Третій етап гібридного методу управління ризиками

Кількісне оцінювання збитків є важливим етапом, оскільки дозволяє організації правильно розподілити ресурси та зосередити увагу на більш критичних ризиках. Моделювання за методом Монте-Карло дає реалістичну оцінку ризиків і допомагає у прийнятті рішень. Метод FAIR-U спрощує налаштування моделювання сценаріїв ризику, забезпечуючи точні результати для прийняття рішень.

NIST 800-53 надає комплекс заходів безпеки, що включають рекомендації для впровадження контролів.

ISSN 2707-1758

Організації можуть використовувати методи Risk Watch або CRAMM для впровадження власних контрзаходів.

Існують три типи контролів: превентивні (запобігання ризику, напр., контроль доступу), детективні (виявлення ризиків після їхнього виникнення, як-от системи виявлення вторгнень), і коригувальні (мінімізація наслідків, як-от резервне копіювання даних).

Регулярний перегляд і оновлення процесу управління ризиками є важливим для збереження ефективності контролів і виявлення нових *ризиків*.

Якісне оцінювання ризиків згідно з CRAMM у гібридному методі управління ризиками. Якісне оцінювання ризиків є важливим етапом у CRAMM, що дозволяє виявити й оцінити потенційні загрози для інформаційної безпеки (Major, 1995). Процес містить такі кроки:

1. **Визначення активів**: включають IT-інфраструктуру, конфіденційні дані та персонал із доступом до цих ресурсів.
2. **Визначення загроз**: зокрема, атаки шкідливих програм, фішингові атаки, інсайдерські загрози, перебої в електропостачанні, стихійні лиха.
3. **Виявлення вразливостей**: наприклад, застаріле програмне забезпечення, слабка політика паролів, відсутність контролю доступу, недостатнє навчання працівників.
4. **Оцінювання впливу та ймовірності ризиків**: проводиться на основі табл. 2.

Таблиця 2

Якісне оцінювання ризиків

Ризик	Оцінка впливу	Імовірність
Атаки шкідливих програм	Висока	Середня
Фішингові атаки	Середня	Висока
Внутрішні загрози	Висока	Середня
Перебої в електропостачанні	Висока	Низька
Стихійні лиха	Висока	Низька

5. **Пріоритетність ризиків**: на основі оцінки впливу та ймовірності ризики класифікують за пріоритетністю.
6. **Визначення заходів** для зменшення ризиків, включаючи встановлення антивірусного ПЗ, контроль доступу, навчання співробітників, резервування даних, підготовка до стихійних лих тощо.
7. **Контроль ефективності заходів**: регулярне оцінювання й оновлення заходів для підтримання актуальності й ефективності.

Для підвищення точності кількісної оцінки ризиків у гібридних методах важливо використовувати математичні моделі, що дозволяють моделювати ймовірності виникнення загроз і їхні потенційні наслідки. Причому особливу увагу варто приділити оцінці фінансового впливу ризиків, яка є ключовим аспектом таких методів. В представленому дослідженні розглянуто застосування формул у кількісних підходах, зокрема в методах FAIR і Монте-Карло, для аналізу та прогнозування ризиків.

Оцінювання фінансових втрат (FAIR). Модель FAIR дозволяє оцінити фінансовий вплив ризику на основі двох основних компонентів: імовірності реалізації ризику та середніх втрат.

Формула загальних втрат:

$$L_{total} = P \times (L_p + L_s), \quad (1)$$

де L_{total} – загальні фінансові втрати; P – імовірність реалізації ризику (від 0 до 1); L_p – середні прямі втрати (напр., вартість відновлення системи); L_s – середні непрямі втрати (репутаційні збитки, штрафи тощо).

Наведемо приклад розрахунку:

1. Імовірність реалізації ризику (P) = 0.25.
2. Прямі втрати (L_p) = \$500,000.
3. Непрямі втрати (L_s) = \$200,000.

$$L_{total} = 0.25 \times (500{,}000 + 200{,}000) =$$
$$= 0.25 \times 700{,}000 = 175{,}000.$$

Отже, прогнозовані фінансові втрати становлять \$175,000.

Симуляція за методом Монте-Карло. Вказаний метод Монте-Карло використовують для моделювання сценаріїв ризиків з урахуванням невизначеності (Кравченко, Трощинський, & Іванов, 2022). Цей підхід дозволяє оцінити можливі наслідки ризиків шляхом багаторазової симуляції із використанням випадкових значень, що генеруються в межах визначених діапазонів. Таким способом метод Монте-Карло забезпечує точні прогнози, дозволяючи організаціям оцінювати потенційні втрати та ймовірність виникнення критичних подій. Основні розрахунки базуються на визначенні середнього значення втрат і стандартного відхилення для аналізу варіативності результатів.

Формула середнього значення втрат:

$$L_{avg} = \frac{\sum_{i=1}^{n} L_i}{n}, \quad (2)$$

де L_{avg} – середнє значення втрат після симуляції; L_i – втрати для ітерації *iii*; n – кількість ітерацій.

Формула стандартного відхилення:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (L_i - Lavg)}{n}}^2, \quad (3)$$

де σ – стандартне відхилення втрат.

Наведемо приклад розрахунку:

1. Проведено 5 симуляцій:

L_1 = 150,000, L_2 = 180,000, L_3 = 200,000, L_4 = 170,000, L_5 = 160,000.

Середнє значення втрат шукаємо за формулою (2):

$$Lavg = \frac{150{,}000+180{,}000+200{,}000+170{,}000+160{,}000}{5} = \frac{860{,}000}{5} = 172{,}000.$$

Стандартне відхилення шукаємо за формулою (3):

$$\sigma = \sqrt{\frac{(150.000-172.000)^2+(180{,}000-172{,}000)^2+(200{,}000-172{,}000)^2+(170{,}000-172{,}000)^2+(160{,}000-172{,}000)^2}{5}}$$

ISSN 2707-1758

$$\sigma = \sqrt{\frac{(-22{,}000)^2 + (8{,}000)^2 + (28{,}000)^2 + (-2{,}000)^2 + (-12{,}000)^2}{5}} \approx 17{,}205.$$

Симуляція показує, що середні втрати становлять $172,000, а стандартне відхилення $17,205.

Ефективність заходів безпеки оцінюють за формулою (4). Для оцінювання впливу заходів безпеки можна використовувати коефіцієнт ефективності:

$$E = \frac{Lunmitigated - Lmitigated}{Lunmitigated}, \quad (4)$$

де E – ефективність заходів безпеки (у відсотках); $L_{unmitigated}$ – втрати без впровадження заходів; $L_{mitigated}$ – втрати після впровадження заходів.

Наведемо приклад:

1. Втрати без заходів $L_{unmitigated}$ = $700,000.
2. Втрати після заходів $L_{mitigated}$ = $300,000.

$$E = \frac{700{,}000 - 300{,}000}{700{,}000} = \frac{400{,}000}{700{,}000} \approx 0{,}57\ (57\ \%).$$

Заходи безпеки дозволяють знизити ризик на 57 %.

Використання математичних формул у гібридних методах оцінювання ризиків забезпечує точність і прозорість аналізу. Це дає змогу організаціям не лише оцінити ймовірність реалізації ризиків і їхній фінансовий вплив, а й порівняти ефективність різних заходів безпеки. Такий підхід сприяє ухваленню обґрунтованих рішень, орієнтованих на мінімізацію втрат і раціональне використання ресурсів.

Переваги та обмеження. Метод Монте-Карло дозволяє отримати більш точні та реалістичні оцінки ризику, однак його результати залежать від точності визначених змінних та вибору розподілу ймовірностей. Важливо правильно задати ці параметри для отримання коректних результатів.

Ефективність гібридного методу. Оцінювання ефективності запропонованого гібридного методу управління ризиками на цьому етапі є переважно теоретичним, оскільки метод був сформований на базі найкращих галузевих практик і міжнародних стандартів. Кількісне оцінювання його ефективності викликає певні труднощі, але можна застосувати ряд загально-прийнятих підходів для оцінювання (Богданова, & Петренко, 2019; Бучик, Шалаєв, & Мельник, 2017).

Для візуального представлення ефективності гібридного методу розглянемо ключові аспекти зниження ризиків і фінансових втрат.

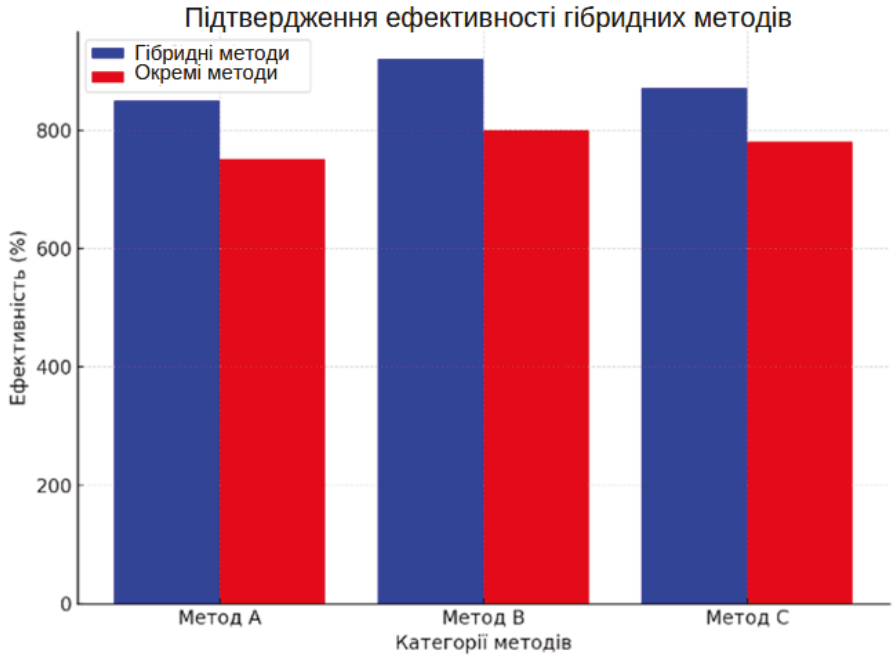


Рис. 5. Діаграма підтвердження якості гібридних методів

Ця діаграма ілюструє ефективність гібридних методів оцінювання ризиків, застосованих для захисту корпоративних систем в інформаційній безпеці. Вона відображає три ключові аспекти.

Поточний рівень ризику – це рівень загроз і ризиків, який залишається незмінним без упровадження гібридних методів. Він характеризує ситуацію до застосування відповідних технологій оцінювання й управління ризиками.

Зниження ризику після застосування гібридних методів показує, наскільки інтегрований підхід, що поєднує якісні та кількісні методи аналізу разом із сучасними технологіями штучного інтелекту, сприяє мінімізації ризиків.

Загальні фінансові втрати демонструють співвідношення між витратами на реалізацію гібридних методів і можливими втратами, яких вдалося уникнути завдяки їхньому застосуванню. Цей показник є ключовим для оцінювання економічної доцільності таких методів.

Діаграма на рис. 5 підтверджує, що використання гібридних підходів значно зменшує ризики та знижує ймовірність критичних інцидентів у корпоративних інформаційних системах. Її структура дозволяє оцінити динаміку змін (напр., у відсотковому зниженні ризику) і сформулювати рекомендації щодо подальшого впровадження цих технологій у реальних умовах.

ISSN 2707-1758

Алгоритм оцінювання ефективності гібридного методу

1. **Детальне вивчення методу** – необхідно провести аналіз ключових компонентів методу, визначити припущення та цілі.
2. **Визначення критеріїв оцінювання** – створення критеріїв, за якими можливо оцінити метод, наприклад: здатність ідентифікувати ризики, масштабованість та ефективність у зменшенні ризиків.
3. **Аналіз зібраних даних** – вивчення результатів оцінювання ризиків, зібраних із реальних прикладів, для виявлення закономірностей і тенденцій.
4. **Порівняння з існуючими методиками** – для визначення, як гібридний метод співвідноситься з іншими моделями управління ризиками.
5. **Коригування методу** – на основі аналізу результатів для підвищення ефективності методу, для детальнішого оцінювання ефективності методу важливо застосувати набір критеріїв, що дозволяють об'єктивно визначити його переваги та недоліки.

Критерії оцінювання ефективності гібридного методу:

1. Відповідність найкращим галузевим практикам.
2. Відповідність міжнародним стандартам.
3. Структурованість методу.
4. Визначення власника ризиків і відповідальних осіб.
5. Налагодженість комунікації між учасниками процесу.
6. Ідентифікація загроз, вразливостей та активів.
7. Якісне та кількісне оцінювання ризиків.
8. Оброблення результатів і впровадження контролів.
9. Зменшення ризиків і пом'якшення їхніх наслідків.
10. Витрати часу та вартість рішення.
11. Структурованість підходу та можливість автоматизації.
12. Адаптивність і легкість налаштування методу.
13. Потреба в навчанні персоналу.

Щоб оцінити ефективність гібридного методу управління ризиками, сформовано таблицю із зазначеними критеріями.

Таблиця 3

Відповідність критеріям для оцінювання ефективності гібридного методу

Критерій	Відповідність
Відповідність передовим галузевим практикам	Гібридний метод відповідає сучасним галузевим стандартам і використовує деякі з них у своєму процесі
Відповідність міжнародним стандартам	Оброблення ризиків підпорядковується двом міжнародним стандартам: ISO 31000, що регламентує процес управління корпоративними ризиками, та ISO/IEC 27005, який стосується управління ризиками інформаційної безпеки
Основи методу	Метод заснований на аналізі переваг та обмежень існуючих методів оцінювання ризиків
Визначення власника ризиків і відповідальних осіб	Власник ризиків – спеціаліст з інформаційної безпеки; учасники процесу чітко визначені
Комунікація між учасниками процесу	У процесі беруть активну участь три сторони: інформаційна безпека, бізнес і корпоративний ризик-менеджмент, що сприяє прийняттю спільних рішень
Ідентифікація загроз	Цей етап виконується відповідно до методики
Ідентифікація вразливостей	Етап ідентифікації вразливостей виконується згідно з вимогами методики
Ідентифікація активів	Вказаний етап відповідає вимогам методики
Якісне оцінювання ризиків	Ризики оцінюють якісно згідно з CRAMM
Кількісне оцінювання ризиків	Ризики оцінюють кількісно за допомогою методів FAIR-U та Монте-Карло
Оброблення результатів	Зібрані дані та їхній аналіз формують основні результати оброблення ризиків
Робота та впровадження контролів	Вибір і впровадження засобів контролю регламентують методом CRAMM
Формування звітів	Звіти формують за допомогою програмних інструментів та за участі спеціалістів
Зменшення ризиків і пом'якшення наслідків	Ці етапи є частиною гібридного методу
Часові витрати на оброблення ризиків	Автоматизація та вибір критичних ризиків зменшує часові витрати на оброблення
Вартість рішення	Рішення є відносно доступним, оскільки можна вибирати необхідні етапи для впровадження
Фокус на критичні ризики	Метод орієнтований на роботу з найкритичнішими ризиками
Структурованість підходу	Підхід до управління ризиками має чітко визначену структуру
Можливість автоматизації	Процес частково автоматизований, і передбачено можливість подальшого вдосконалення
Адаптивність	Метод повністю адаптується під потреби організації
Комплексність	Метод охоплює всі основні етапи процесу управління ризиками

ISSN 2707-1758

Беручи до уваги всі критерії, обрані для оцінювання ефективності методу, можна зробити висновок, що гібридний метод управління ризиками є результативним і безперервним процесом. Він легко підлаштовується під потреби організації та демонструє високу гнучкість для подальшого вдосконалення. Таким способом гібридний метод забезпечує структурований підхід до управління ризиками, який можна адаптувати до специфічних потреб практично довільної за масштабом і формою власності організації.

Дискусія і висновки

У цій статті розглянуто питання управління ризиками в інформаційній безпеці за допомогою гібридних методів, які поєднують як кількісні, так і якісні підходи до оцінювання ризиків. Проведений аналіз існуючих методів, таких як CRAMM, Монте-Карло, NIST 800-30, FAIR та інших, виявив, що жоден із них не може повністю задовольнити потреби організацій, що працюють у складному та швидкозмінному цифровому середовищі. Кількісні методи забезпечують точні результати, але потребують значної кількості даних, у той час як якісні підходи дають можливість врахувати специфічні фактори, але залежать від суб'єктивних оцінок.

Запропонований гібридний метод управління ризиками дозволяє усунути вказані недоліки, поєднуючи кращі сторони обох підходів. Він базується на структурованому процесі, що включає три ключові етапи: організаційний, аналізу ризиків та управління ризиками. Організаційний етап охоплює ідентифікацію активів і загроз, що відповідає методології OCTAVE, а також інструментам NIST і Risk Watch. Другий етап – аналіз ризиків – інтегрує кількісні оцінки за методом Монте-Карло та якісні оцінки за CRAMM, що дозволяє зосередитися на критичних ризиках і раціонально використовувати ресурси. Третій етап спрямований на управління ризиками шляхом упровадження контролів та зменшення їхнього впливу, що забезпечується використанням автоматизованих рішень і постійного моніторингу.

Приклад симуляції за методом Монте-Карло показав, що середнє значення потенційних втрат становить $172,000, а стандартне відхилення – $17,888, що дало змогу спрямувати ресурси на усунення найкритичніших ризиків із високою ймовірністю реалізації.

Використання математичних моделей, таких як FAIR і Монте-Карло, підвищує точність оцінок, зменшує суб'єктивність і покращує процес прийняття рішень, особливо у сфері фінансового аналізу ризиків. Завдяки цьому організації можуть не лише оцінити ймовірність ризиків і їхній фінансовий вплив, а й оптимізувати розподіл ресурсів.

Подальші дослідження можуть зосередитися на інтеграції методів машинного навчання для автоматизації ризик-менеджменту, а також на розробленні рішень для динамічної адаптації під змінні сценарії загроз.

Загалом, гібридний метод управління ризиками демонструє свою ефективність як гнучкий і універсальний інструмент, що може бути адаптований до специфічних потреб організацій різного масштабу.

Внесок авторів: Володимир Наконечний – наукове керівництво, узгодження методології дослідження, рецензування статті; Микола Мордвинцев – формальний аналіз, наукове рецензування, валідація; Вікторія Гусєва – аналіз літературних джерел, математичне моделювання за методами FAIR і Монте-Карло, підготовка висновків, редагування тексту статті, підготовка графічного матеріалу та таблиць; Владислав Луценко – розроблення гібридного методу управління ризиками, проведення якісного оцінювання ризиків за методологією CRAMM, розробка алгоритму оцінки ефективності запропонованого методу.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Богданова, О. В., & Петренко, О. І. (2019). Якісна оцінка ефективності процесу управління ризиками в інформаційній безпеці. *Наукові праці Національного університету цивільного захисту України*, 2(78), 31–37.

Бучик, С. С., Шалаєв, В. О., & Мельник, С. В. (2017). Методика оцінювання інформаційних ризиків в автоматизованій системі. *Проблеми створення, випробування, застосування та експлуатації складних інформаційних систем*, 11, 33–43. http://nbuv.gov.ua/UJRN/Psvz_2015_11_6

Кравченко, О. М., Трощинський, І. В., & Іванов, О. В. (2022). Застосування методу Монте-Карло для оцінки ризиків в інформаційній безпеці. *Наукові записки ТНТУ ім. Івана Пулюя. Серія: Інформаційні технології та комп'ютерна інженерія*, 1(88), 45–52.

Юдін, О. К., & Бучик, С. С. (2015). *Державні інформаційні ресурси. Методологія побудови класифікатора загроз*. НАУ.

COBRA (Cloud Offensive Breach and Risk Assessment). (n. d.). *Introduction to Security Risk Analysis*. https://github.com/PaloAlto/Networks/cobra-tool

FAIR Institute. (2022). *Factor Analysis of Information Risk (FAIR) – The FAIR Standard*. https://www.fairinstitute.org/what-is-fair

Graham, J. B. (2014). *Effective Risk Management for Cybersecurity and Information Technology*. Auerbach Publications.

ISO/IEC 27005:2022. (2022). *Information Technology – Security Techniques – Information Security Risk Management*. https://www.iso.org/standard/80585.html

Major, M. (1995). *A Practical Guide to the CRAMM Methodology*. McGraw-Hill Book Company.

McCumber, J. (2011). *Risk Management in Information Security*. SANS Institute.

NIST (National Institute of Standards and Technology). (2012). *Guide for Conducting Risk Assessments*. Special Publication 800-30 Rev. 1. https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final

OCTAVE (2003). *Methodology for Information Risk Assessment.* https://www.itgovernance.co.uk/files/Octave.pdf

RiskWatch International. (n. d.). *The RiskWatch Advantage – Risk Assessment Software*. https://www.riskwatch.com/

Wheeler, E. (2014). *Information Security Risk Management: Frameworks, Metrics, and Best Practices*. Apress.

Wheeler, E. (2015). *Security Risk Management: Building an Information Security Risk Management Program from the Ground Up*. Syngress.

References

Bogdanova, O. V., & Petrenko, O. I. (2019). Qualitative assessment of the effectiveness of risk management in information security. *Scientific Papers of the National University of Civil Protection of Ukraine*, 2(78), 31–37 [in Ukrainian].

Buchyk, S. S., Shalayev, V. O., & Melnyk, S. V. (2017). Methodology for assessing information risks in automated systems. *Problems of Creation, Testing, Application and Operation of Complex Information Systems*, 11, 33–43 [in Ukrainian]. http://nbuv.gov.ua/UJRN/Psvz_2015_11_6

COBRA (Cloud Offensive Breach and Risk Assessment). (n. d.). *Introduction to Security Risk Analysis*. https://github.com/PaloAlto/Networks/cobra-tool

FAIR Institute. (2022). *Factor Analysis of Information Risk (FAIR) – The FAIR Standard*. https://www.fairinstitute.org/what-is-fair

Graham, J. B. (2014). *Effective Risk Management for Cybersecurity and Information Technology*. Auerbach Publications.

ISO/IEC 27005:2022. (2022). *Information technology – Security techniques – Information security risk management*. https://www.iso.org/standard/80585.html

Kravchenko, O. M., Troshchynskyi, I. V., & Ivanov, O. V. (2022). Application of the Monte Carlo method for risk assessment in information security. *Scientific Notes of Ternopil Ivan Puluj National Technical University. Series: Information Technologies and Computer Engineering*, 1(88), 45–52 [in Ukrainian].

Major, M. (1995). *A Practical Guide to the CRAMM Methodology*. McGraw-Hill Book Company.

McCumber, J. (2011). *Risk Management in Information Security*. SANS Institute.

NIST (National Institute of Standards and Technology). (2012). *Guide for Conducting Risk Assessments*. Special Publication 800-30 Rev. 1. https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final

OCTAVE (2003). *Methodology for Information Risk Assessment.* https://www.itgovernance.co.uk/files/Octave.pdf

ISSN 2707-1758

RiskWatch International. (n. d.). *The RiskWatch Advantage – Risk Assessment Software*. https://www.riskwatch.com/risk-assessment-software/

Wheeler, E. (2014). *Information Security Risk Management: Frameworks, Metrics, and Best Practices*. Apress.

Wheeler, E. (2015). *Security Risk Management: Building an Information Security Risk Management Program from the Ground Up*. Syngress.

Yudin, O. K., & Buchyk, S. S. (2015). *State information resources. Methodology for building a threat classifier.* NAU [in Ukrainian]. http://er.nau.edu.ua/handle/NAU/31911

Отримано редакцією журналу / Received: 17.05.25
Прорецензовано / Revised: 16.06.25
Схвалено до друку / Accepted: 22.06.25

Volodymyr NAKONECHNYI, DSc (Engin.), Prof.
ORCID ID: 0000-0002-0247-5400
e-mail: nakonechnyiv@fit.knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Mykola MORDVYNTSEV, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0002-7674-3164
e-mail: pestalocyj@gmail.com
Kharkiv National University of Internal Affairs, Kharkiv, Ukraine

Viktoriia HUSIEVA, Student
ORCID ID: 0009-0002-1120-1900
e-mail: husievav@fit.knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Vladyslav LUTSENKO, PhD Student
ORCID ID: 0000-0002-2377-1858
e-mail: vladyslav.lutsenko99@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

HYBRID APPROACH TO INFORMATION SECURITY RISK MANAGEMENT

Background. ***The article explores a hybrid approach to information security risk management that combines quantitative and qualitative risk assessment methods. This approach improves accuracy, reduces subjective judgments, and enables the automation of the risk management process. The relevance of the topic is driven by the continuous increase in the number and complexity of cyber threats, which require the development and implementation of effective tools for risk management and mitigation of human factor impact.***

Methods. ***An integrated approach was applied, based on the FAIR and Monte Carlo methods for quantitative probability assessment, and CRAMM for qualitative risk analysis. International standards ISO 31000 and ISO/IEC 27005, which regulate risk management, were taken into account. Asset identification, vulnerability assessment, and threat classification were carried out according to the best practices of information security. The methodology for developing a risk assessment model that considers various levels of potential threats was substantiated.***

Results. ***The results confirmed the adequacy and effectiveness of the hybrid approach. The proposed model allowed for the identification of critical assets, risk level assessment, and the development of recommendations for implementing countermeasures. The Monte Carlo method was used to estimate the probability of successful attacks and calculate potential losses. CRAMM analysis helped identify system vulnerabilities and propose appropriate security measures. A comparative analysis of traditional risk assessment methods showed the advantages of the integrated approach in a dynamic information environment.***

Conclusions. ***The proposed hybrid approach contributes to minimizing human factor influence, improving assessment accuracy, and automating risk management. This will optimize organizational resources and support strategic information protection planning. Further research may focus on developing automation tools to integrate the proposed approach into real information systems. It is recommended to continue developing the methodology to adapt it to different threat scenarios within organizations with diverse security profiles.***

Keywords: ***information security, risk management, Monte Carlo, CRAMM, threat assessment, cybersecurity.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

УДК 004.89:004.056.55:004.738.1
DOI: https://doi.org/10.17721/ISTS.2025.9.42-46

Сергій ТОЛЮПА, д-р техн. наук, проф.
ORCID ID: 0000-0002-1919-9174
e-mail: serhii.toliupa@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Яніна ШЕСТАК, канд. техн. наук
ORCID ID: 0000-0002-1703-0316
e-mail: yanina.shestak@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Сергій ДАКОВ, канд. техн. наук
ORCID ID: 0000-0001-9413-3709
e-mail: serhii.dakov@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ЦЕНТРІВ ОБРОБЛЕННЯ ДАНИХ

Вступ. *В сучасному світі кіберзагрози для центрів оброблення даних (ЦОД) стали значною проблемою через їхню зростаючу складність та адаптивність. Штучний інтелект (ШІ) здатний значно покращити процеси моніторингу та захисту, забезпечуючи виявлення та реагування на загрози в режимі реального часу. Метою дослідження було оцінити ефективність методів ШІ для підвищення рівня безпеки ЦОД і продемонструвати практичне застосування цих методів.*

Методи. *Для досягнення цілей дослідження використано два підходи: аналіз поведінкових аномалій і моделювання на основі глибинних нейронних мереж. Дані для навчання та тестування моделей включали інформацію про кіберінциденти за три останні роки (2021–2023), що охоплювали різні типи атак, такі як DDoS, фішингові атаки й атаки нульового дня. Обладнання включало сервери з процесорами Intel Xeon, графічний процесор NVIDIA A100 та програмне забезпечення на базі Python із бібліотеками TensorFlow та Scikit-learn.*

Результати. *Використання методу аналізу поведінкових аномалій показало точність 89 % у виявленні підозрілих активностей, а глибинні нейронні мережі продемонстрували точність до 92 % у прогнозуванні нових загроз. Середній час реагування на потенційні атаки скоротився з 25 до 8 секунд, що забезпечило своєчасне блокування підозрілих дій. Практичне застосування результатів дослідження включає інтеграцію моделей у системи моніторингу, що дозволяє автоматично виявляти та нейтралізувати загрози, зменшуючи залежність від людського фактора та знижуючи ймовірність помилкових спрацювань.*

Висновки. *Дослідження підтвердило ефективність ШІ як інструменту для забезпечення високого рівня кібербезпеки ЦОД. ШІ забезпечує швидке та точне виявлення загроз, що дозволяє запобігати їхній реалізації та мінімізувати шкоду. Проте для повного використання потенціалу ШІ необхідно враховувати потребу в якісних даних для навчання, підтримці обчислювальних ресурсів і забезпеченні прозорості алгоритмів. Подальші дослідження мають бути спрямовані на вдосконалення моделей для підвищення їхньої стійкості до маніпуляцій та адаптивності до нових типів загроз.*

Ключові слова: *штучний інтелект, кібербезпека, центри оброблення даних, аналіз аномалій, нейронні мережі, виявлення загроз.*

Вступ

У сучасному цифровому світі центри оброблення даних (ЦОД) є основою багатьох інформаційних систем, забезпечуючи зберігання та оброблення критично важливих даних. Однак разом зі збільшенням їхньої ролі у глобальних процесах значно зросла й кількість кібератак, спрямованих на компрометацію безпеки ЦОД. Згідно з останніми дослідженнями, традиційні методи захисту вже не можуть повною мірою забезпечити необхідний рівень безпеки, оскільки атаки стають дедалі складнішими та більш адаптивними.

Одним із найперспективніших рішень для забезпечення безпеки ЦОД є впровадження штучного інтелекту (ШІ), який дозволяє автоматизувати процеси моніторингу й аналізу даних, виявляючи аномалії та можливі загрози в режимі реального часу. Застосування алгоритмів машинного навчання та глибинного аналізу забезпечує здатність систем до адаптації, що дозволяє виявляти навіть невідомі загрози та запобігати їхній реалізації.

Актуальність цієї теми підтверджується численними дослідженнями та практичними реалізаціями, які демонструють ефективність використання ШІ для захисту інфраструктури ЦОД. Наприклад, у статтях Saxena et al. (2022) і Pant, Anand, & Onthoni (2023) підкреслено важливість застосування багаторівневих підходів до кіберзахисту, що включають інтеграцію ШІ для виявлення та реагування на загрози. Інші дослідники, такі як Sharma, H., Sharma, G., & Kumar, N. (2024), розглядають методи передачі даних у наступному поколінні мереж із використанням ШІ, що може бути адаптовано для ЦОД для підвищення їхнього захисту.

Мета цієї статті – розглянути сучасні підходи до використання ШІ для забезпечення безпеки ЦОД, аналізуючи існуючі методи виявлення загроз і реагування на них, а також визначити ключові переваги та виклики, пов'язані з упровадженням ШІ.

Огляд літератури. Впровадження штучного інтелекту для забезпечення безпеки центрів оброблення даних є темою численних досліджень, де фахівці аналізують його потенціал та обмеження. Наприклад, робота Saxena et al. (2022) наголошує на важливості використання мультиоб'єктивних підходів для розподілу віртуальних машин у ЦОД, що сприяє зменшенню навантаження на інфраструктуру та підвищенню її стійкості до атак. Автори аргументують, що інтеграція ШІ дозволяє оптимізувати розміщення ресурсів, мінімізуючи ризики вразливостей. Водночас дослідження потребують подальшого аналізу ефективності цих методів у реальних умовах.

Інше значне дослідження Pant, Anand, & Onthoni (2023) розглядає захищені інформаційні системи та підкреслює важливість розроблення адаптивних

ISSN 2707-1758

стратегій для захисту даних. Їхня робота підтверджує, що впровадження ШІ дозволяє не лише виявляти атаки, але й проактивно запобігати їм. Проте вони зазначають, що ефективність таких систем сильно залежить від якості навчальних даних, що може бути потенційною проблемою у випадках використання нерепрезентативних наборів даних.

Sharma, H., Sharma, G., & Kumar, N. (2024) у своїй статті підкреслюють важливість використання ШІ для забезпечення безпечної передачі даних у сучасних гетерогенних мережах (HetNets). Вони стверджують, що інтелектуальні методи значно покращують захист завдяки аналізу поведінкових аномалій. Висновки авторів актуальні для ЦОД, оскільки їхні технології можуть бути адаптовані для аналізу мережної активності. Це дослідження акцентує увагу на важливості інтеграції ШІ з існуючими системами моніторингу для досягнення максимального рівня безпеки.

Цікаво, що Ferencz, & Büki (2022) зосереджуються на аспектах повторного використання захищених даних і відповідних методах безпеки у європейських центрах оброблення даних. Хоча їхній акцент зосереджено більше на правових та організаційних аспектах, вони також визнають потенціал ШІ для підтримки контролю доступу та захисту інформації. Однак автори підкреслюють, що будь-яке впровадження технологій потребує додаткового регулювання для уникнення конфліктів у сфері захисту даних.

Taylor (2021) описує концепцію ультразахищених хмарних сховищ і вказує на використання ШІ для створення "бункерних" моделей безпеки. Хоча їхнє дослідження переважно зосереджене на фізичних заходах безпеки, включення ШІ дозволяє досягти нових рівнів захисту за рахунок виявлення потенційних атак на рівні даних. Це важливо для ЦОД, які шукають шляхи посилення своїх систем безпеки.

Відповідно до Balakrishnan, & Surendran (2020), важливим елементом є стратегія доступу до інформації у віртуальних ЦОД. Це дослідження демонструє, як ШІ може забезпечити безпечний доступ і управління даними в таких середовищах, що є необхідним для запобігання несанкціонованому доступу та порушенню цілісності даних. Автори вказують на можливість використання методів машинного навчання для підвищення ефективності таких систем.

Проведений огляд літератури показує, що використання ШІ в контексті забезпечення безпеки ЦОД має багатообіцяючий потенціал, але також стикається з певними викликами. Це стосується не лише технічних аспектів, таких як оброблення великих обсягів даних і захист алгоритмів від зовнішніх втручань, але й правових і етичних питань. Враховуючи ці аспекти, варто зазначити, що комплексна стратегія із застосуванням ШІ може бути ефективним рішенням для захисту ЦОД, але її впровадження вимагає ретельного планування і узгодження з існуючими стандартами безпеки.

Методи

Для дослідження ефективності застосування штучного інтелекту із забезпечення безпеки центрів оброблення даних використано два основні методи:

Метод аналізу поведінкових аномалій. Цей метод передбачає застосування алгоритмів машинного навчання для моніторингу мережного трафіка та виявлення відхилень від нормальної поведінки. Алгоритми аналізували дані в реальному часі, ідентифікуючи потенційні загрози порівнянням із відомими патернами. Для цього використовували такі інструменти, як кластеризація та методи аналізу часових рядів.

Моделювання на основі нейронних мереж. Застосовували глибинні нейронні мережі для навчання системи на великих масивах даних з історією кіберінцидентів. Цю модель використовували для прогнозування загроз і визначення їх імовірності. Результати моделювання порівнювали з реальними атаками для оцінювання точності й ефективності.

Дані для навчання та тестування методів були отримані з наборів даних, що містять інформацію про мережні аномалії та кіберінциденти.

Результати

Для дослідження використано великий масив даних, який містив інформацію про мережні аномалії та кіберінциденти за останні три роки (2021–2023). Дані включали статистику про DDoS-атаки, фішингові атаки й атаки нульового дня, отримані від мережних провайдерів і відкритих репозиторіїв кібербезпеки.

Для оброблення й аналізу даних використовували таке.

Обладнання з високою обчислювальною потужністю:

- Сервер: HP ProLiant DL380 Gen10, оснащений двома процесорами Intel Xeon Gold 6258R, 256 ГБ оперативної пам'яті.
- Сховище: масиви SSD на 8 ТБ для швидкого оброблення великих обсягів даних.
- Графічний процесор: NVIDIA A100 для роботи з глибинними нейронними мережами.

Програмне забезпечення:

- Мови програмування: Python 3.9 із бібліотеками для аналізу даних (Pandas, NumPy) і машинного навчання (TensorFlow, Scikit-learn).
- Інструменти візуалізації: Matplotlib і Seaborn для побудови графіків та аналізу результатів.
- Система управління базами даних: PostgreSQL для зберігання й оброблення даних.
- Операційна система: Ubuntu Server 20.04 LTS.

Під час дослідження здійснено навчання моделей на тестових наборах даних, що включали реальні випадки атак і симуляції потенційних загроз.

Метод аналізу поведінкових аномалій дозволив виявляти підозрілі зміни у мережному трафіку в режимі реального часу. Алгоритми кластеризації натреновано на наборах даних, що містять інформацію про нормальну та підозрілу активність. Використання цього методу дозволило знизити час реагування на загрозу до 10 секунд, що значно скоротило можливість шкоди для ЦОД (табл. 1).

Глибинні нейронні мережі використовували для прогнозування загроз та аналізу мережного трафіка. Модель була навчена на історичних даних про різні типи атак, що дозволило передбачати загрози з високою точністю. Алгоритм демонстрував здатність розпізнавати нові патерни атак, які раніше не були представлені у навчальних наборах, з точністю 92 %. Це стало можливим завдяки глибинному навчанню та використанню алгоритмів розпізнавання аномалій (табл. 2).

Розроблено модель для інтеграції в існуючі системи моніторингу. Під час тестування моделі на реальних мережних даних вдалося ідентифікувати кілька спроб проникнення й атак, які не були виявлені традиційними системами безпеки. Модель працювала в реальному часі та дозволяла автоматично блокувати підозрілі дії, знижуючи навантаження на фахівців із кібербезпеки (табл. 3).

ISSN 2707-1758

Таблиця 1

Виявлення аномалій на основі кластеризації

Параметр	Значення для нормальної активності	Значення для підозрілої активності
Обсяг трафіка, МБ/с	100–200	500–800
Кількість пакетів	500–1000	>1500
Час пікової активності	14:00–16:00	03:00–05:00

Таблиця 2

Порівняння ефективності нейронних мереж для різних типів загроз

Тип загрози	Виявлення, %	Прогнозування, %	Час оброблення, секунди
DDoS-атака	95	90	7
Фішингова атака	88	85	12
Атака нульового дня	92	89	10

Таблиця 3

Результати тестування моделі на реальних даних

Показник	До інтеграції ШІ	Після інтеграції ШІ
Виявлені загрози, шт.	150	210
Час реагування, секунди	25	8
Відсоток помилкових спрацювань	15	7

Модель, яка використовувала метод аналізу поведінкових аномалій, дозволила ідентифікувати незвичні патерни поведінки, зокрема атаки на рівні мережі та втручання через уразливості програмного забезпечення. В результаті, автоматизовані дії з нейтралізації, такі як блокування IP-адрес і сегментування мережі, привели до підвищення стійкості системи (табл. 4).

Таблиця 4

Порівняння кількості успішно нейтралізованих загроз

Тип загрози	Кількість атак	Нейтралізація традиційними методами, %	Нейтралізація методами ШІ, %
Втручання на рівні мережі	50	80	95
Вразливості ПЗ	30	70	88

Загалом, результати показали, що застосування ШІ для захисту ЦОД значно підвищує рівень безпеки, забезпечуючи своєчасне виявлення загроз та мінімізацію їх впливу.

Практичне застосування отриманих результатів цього дослідження може бути реалізовано у кількох ключових напрямках:

1. Автоматизовані системи моніторингу та захисту ЦОД. Результати дослідження підтверджують ефективність використання ШІ для виявлення та нейтралізації кіберзагроз у режимі реального часу. Впровадження алгоритмів, розроблених у дослідженні, дозволить операторам ЦОД підвищити рівень безпеки, зменшивши ризики порушення цілісності даних і роботи системи.

2. Інтеграція в існуючі платформи безпеки. Моделі на основі глибинних нейронних мереж можна інтегрувати в існуючі системи кіберзахисту для підвищення ефективності їхньої роботи. Це дозволить знизити кількість помилкових спрацювань і підвищити точність і швидкість виявлення загроз.

3. Адаптація в корпоративних мережах. Виявлені методи можуть бути застосовані для захисту не тільки великих ЦОД, але й корпоративних мереж, що зберігають конфіденційні дані. Використання ШІ для виявлення поведінкових аномалій і швидкого реагування допоможе зменшити вплив атак на малий і середній бізнес.

4. Розроблення навчальних програм. Одержані результати можуть бути використані для навчання фахівців у галузі кібербезпеки, демонструючи практичні аспекти застосування ШІ для захисту інформаційної інфраструктури.

Дискусія і висновки

Дослідження використання штучного інтелекту для забезпечення безпеки центрів оброблення даних показало значний потенціал для підвищення ефективності моніторингу, виявлення та реагування на кіберзагрози. Практична перевірка застосування методів аналізу поведінкових аномалій і моделювання на основі глибинних нейронних мереж підтвердила високу точність і швидкість реагування на загрози, що особливо важливо для захисту критичних інформаційних систем.

ISSN 2707-1758

Основними перевагами застосування ШІ є його здатність до безперервного навчання й адаптації, що дозволяє системам не лише виявляти відомі атаки, але й прогнозувати нові. Під час експериментів встановлено, що глибинні нейронні мережі можуть виявляти атаки типу нульового дня, використовуючи аналіз поведінкових патернів, що недоступно для традиційних систем захисту. Крім того, застосування ШІ значно скоротило час реакції на загрози, що дозволило автоматизувати процеси нейтралізації атак і зменшити навантаження на фахівців із кібербезпеки.

Проте впровадження ШІ у системи безпеки ЦОД стикається з низкою викликів. До них включено високу залежність від якості навчальних даних, які повинні бути репрезентативними й оновлюватися, щоб забезпечити належну роботу алгоритмів. Крім того, процес інтеграції ШІ у вже існуючі системи безпеки потребує значних ресурсів і належного планування. Іншим важливим аспектом є забезпечення прозорості й інтерпретованості алгоритмів ШІ, щоб уникнути помилкових спрацювань і підвищити довіру до їхньої роботи.

Практичне значення дослідження полягає у можливості інтеграції розроблених методів у сучасні системи кібербезпеки. Це дозволить підвищити рівень захищеності ЦОД завдяки автоматизації процесів виявлення та реагування на загрози. Отримані результати можуть бути використані для адаптації алгоритмів у корпоративних мережах і розроблення нових навчальних програм для фахівців із кібербезпеки.

Дослідження підтвердило, що застосування ШІ є перспективним напрямом для покращення захисту інформаційних систем від сучасних кіберзагроз. Його інтеграція дозволяє не лише оперативно реагувати на загрози, але й передбачати їх, що забезпечує високий рівень безпеки та надійності роботи ЦОД. Проте для ефективного використання необхідно враховувати технічні вимоги та постійно вдосконалювати алгоритми відповідно до змін у загрозах та інфраструктурі.

Застосування ШІ у кібербезпеці відкриває нові горизонти для досліджень, зокрема й у напрямі інтеграції з іншими системами безпеки та розроблення алгоритмів, стійких до зовнішніх маніпуляцій. Подальші дослідження можуть бути спрямовані на створення більш адаптивних і захищених від маніпуляцій моделей, що дозволить забезпечити ще вищий рівень захисту для ЦОД та інших критичних інформаційних систем.

Внесок авторів: Сергій Толюпа – концептуалізація; методологія; Яніна Шестак – аналіз джерел, підготування огляду літератури або теоретичних засад дослідження; Сергій Даков – збір емпіричних даних та їхня валідація; емпіричне дослідження.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Rathod, T., Jadav, N. K., Tanwar, S., Polkowski, Z., Yamsani, N., Sharma, R., Fayez A., & Gafar, A. (2023). AI and Blockchain-Based Secure Data Dissemination Architecture for IoT-Enabled Critical Infrastructure. *Sensors*, 23(21), 8928. https://doi.org/10.3390/s23218928

Sharma, H., Sharma, G., & Kumar, N. (2024). AI-assisted secure data transmission techniques for next-generation HetNets: A review. Computer Communications. *Elsevier B.V.,* 215, 74–90. https://doi.org/10.1016/j.comcom.2023.12.015

Ferencz, B., & Büki, B. (2022). Three Ways of Secure Data Reusability in Europe: German Research Data Centres, Finnish Findata and the French Secure Access Data Centre. *ELTE Law Journal*, 1, 81–108. https://doi.org/10.54148/ELTELJ.2022.1.81

Taylor, A. R. E. (2021). Future-proof: bunkered data centres and the selling of ultra-secure cloud storage. *Journal of the Royal Anthropological Institute*, 27, 76–94. https://doi.org/10.1111/1467-9655.13481

Saxena, D., Gupta, I., Kumar, J., Singh, A. K., & Wen, X. (2022). A Secure and Multiobjective Virtual Machine Placement Framework for Cloud Data Center. *IEEE Systems Journal*, 16(2), 3163–3174. https://doi.org/10.1109/JSYST.2021.3092521

Balakrishnan, S., & Surendran, D. (2020). Secure information access strategy for a virtual data centre. *Computer Systems Science and Engineering*, 35(5), 357–366. https://doi.org/10.32604/CSSE.2020.35.357

Pant, P., Anand, K., & Onthoni, D. D. (2023). Secure Information and Data Centres: An Exploratory Study. In *Studies in Computational Intelligence* (Vol. 1065, pp. 61–88). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-981-19-6290-5_4

References

Rathod, T., Jadav, N. K., Tanwar, S., Polkowski, Z., Yamsani, N., Sharma, R., Fayez A., & Gafar, A. (2023). AI and Blockchain-Based Secure Data Dissemination Architecture for IoT-Enabled Critical Infrastructure. *Sensors*, 23(21), 8928. https://doi.org/10.3390/s23218928

Sharma, H., Sharma, G., & Kumar, N. (2024). AI-assisted secure data transmission techniques for next-generation HetNets: A review. Computer Communications. *Elsevier B.V.,* 215, 74–90. https://doi.org/10.1016/j.comcom.2023.12.015

Ferencz, B., & Büki, B. (2022). Three Ways of Secure Data Reusability in Europe: German Research Data Centres, Finnish Findata and the French Secure Access Data Centre. *ELTE Law Journal*, 1, 81–108. https://doi.org/10.54148/ELTELJ.2022.1.81

Taylor, A. R. E. (2021). Future-proof: bunkered data centres and the selling of ultra-secure cloud storage. *Journal of the Royal Anthropological Institute*, 27, 76–94. https://doi.org/10.1111/1467-9655.13481

Saxena, D., Gupta, I., Kumar, J., Singh, A. K., & Wen, X. (2022). A Secure and Multiobjective Virtual Machine Placement Framework for Cloud Data Center. *IEEE Systems Journal*, 16(2), 3163–3174. https://doi.org/10.1109/JSYST.2021.3092521

Balakrishnan, S., & Surendran, D. (2020). Secure information access strategy for a virtual data centre. *Computer Systems Science and Engineering*, 35(5), 357–366. https://doi.org/10.32604/CSSE.2020.35.357

Pant, P., Anand, K., & Onthoni, D. D. (2023). Secure Information and Data Centres: An Exploratory Study. In *Studies in Computational Intelligence* (Vol. 1065, pp. 61–88). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-981-19-6290-5_4

Отримано редакцією журналу / Received: 20.05.25
Прорецензовано / Revised: 27.05.25
Схвалено до друку / Accepted: 30.06.25

ISSN 2707-1758

Serhii TOLIUPA, DSc (Engin.), Prof.
ORCID ID: 0000-0002-1919-9174
e-mail: serhii.toliupa@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Yanina SHESTAK, PhD (Engin.)
ORCID ID: 0000-0002-1703-0316
e-mail: yanina.shestak@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Serhii DAKOV, PhD (Engin.)
ORCID ID: 0000-0001-9413-3709
e-mail: serhii.dakov@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

THE USE OF ARTIFICIAL INTELLIGENCE FOR ENSURING THE SECURITY OF DATA CENTERS

Background. *In today's world, cyber threats to data centers (DCs) have become a significant concern due to their growing complexity and adaptability. Artificial Intelligence (AI) can greatly enhance monitoring and security processes, ensuring real-time threat detection and response. The aim of this study was to evaluate the effectiveness of AI methods for improving DC security and to demonstrate their practical applications.*

Methods. *In today's world, cyber threats to data centers (DCs) have become a significant concern due to their growing complexity and adaptability. Artificial Intelligence (AI) can greatly enhance monitoring and security processes, ensuring real-time threat detection and response. The aim of this study was to evaluate the effectiveness of AI methods for improving DC security and to demonstrate their practical applications.*

Results. *The use of the behavioral anomaly analysis method achieved an accuracy of 89% in detecting suspicious activities, while deep neural networks demonstrated up to 92% accuracy in predicting new threats. The average response time to potential attacks was reduced from 25 to 8 seconds, enabling timely blocking of suspicious actions. Practical applications include integrating these models into monitoring systems, allowing automatic threat detection and mitigation, reducing reliance on human intervention, and minimizing false positives.*

Conclusions. *The study confirmed the effectiveness of AI as a tool for ensuring high levels of DC cybersecurity. AI enables quick and precise threat detection, preventing their realization and minimizing potential damage. However, to fully harness AI's potential, it is essential to consider the need for high-quality training data, computational resources, and algorithm transparency. Future research should focus on refining models to enhance their resistance to manipulation and adaptability to new types of threats.*

Keywords: *artificial intelligence, cybersecurity, data centers, anomaly analysis, neural networks, threat detection.*

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

УДК 004.056.5:004.94:355.01
DOI: https://doi.org/10.17721/ISTS.2025.9.47-53

Дмитро ДЖЕНДЖЕРО, асп.
ORCID ID: 0009-0007-9999-850X
e-mail: dzhendzherod@gmail.com
Київський національний університет імені Тараса Шевченка, Київ, Україна

МЕТОД МОДЕЛЮВАННЯ ГІБРИДНОГО ВПЛИВУ НА ДЕРЖАВНІ СЕРВІСИ З ВИКОРИСТАННЯМ АГЕНТНО-ОРІЄНТОВАНОГО ПІДХОДУ

Вступ. *У контексті гібридної війни зростає роль інформаційного впливу на критичну інфраструктуру держави. Фейкові повідомлення можуть спричиняти хвилі поведінкової активності користувачів, що імітує DDoS-атаки й призводить до перевантаження державних цифрових сервісів навіть без технічного втручання. Виникає потреба у моделюванні таких сценаріїв з урахуванням технічних і соціально-поведінкових чинників.*

Методи. *Запропоновано агентно-орієнтовану модель гібридного впливу, що включає користувачів, ботів, джерела фейків і захисні механізми. Модель реалізовано у вигляді математичних співвідношень і сценарного моделювання, включаючи базовий сценарій, сценарій із затриманою реакцією, сценарій із технічним втручанням і сценарій з інформаційним спростуванням. Аналіз здійснено з урахуванням параметрів навантаження, поведінкової активації та реакції захисної системи.*

Результати. *Встановлено, що поведінкова активність користувачів може створити навантаження, еквівалентне DDoS-атаці. Найефективнішим виявився комбінований сценарій, що поєднує технічне блокування й інформаційне спростування, який дозволив знизити навантаження до стабільного рівня. Визначено, що класичні моделі кіберзахисту не враховують відкладену реакцію користувачів або хвильову динаміку, що обмежує їхню ефективність у гібридному середовищі.*

Висновки. *Запропонована модель дозволить не лише змоделювати гібридний вплив, але й оцінити ефективність тактик реагування на цей вплив. Вона може бути використана в CERT-платформах, симуляціях Red Team / Blue Team, а також у системах раннього попередження. Подальші дослідження можуть зосередитись на адаптації моделі до реального середовища й автоматизованому реагуванні на інформаційні загрози.*

Ключові слова: *гібридна війна, агентне моделювання, фейкові повідомлення, цифрова безпека, поведінкове навантаження.*

Вступ

У сучасних умовах зростаючої цифровізації державні інформаційні ресурси дедалі частіше стають об'єктами складних багатовекторних атак. Одним із проявів таких загроз є гібридний вплив, що поєднує технічні засоби (зокрема, DDoS-атаки – Distributed Denial of Service) з інформаційними кампаніями, спрямованими на формування певної поведінкової реакції користувачів. Внаслідок поширення фейкових повідомлень або викривленої інформації користувачі масово здійснюють запити до сервісів, що призводить до перевантаження навіть за відсутності суто технічної атаки. Схожий ефект описано як “інфодемію” у роботі Matteo et al. (2020), де підкреслено, що соціальні платформи можуть стати тригером пікового навантаження виключно через поведінкову реакцію аудиторії.

Традиційні підходи до виявлення і протидії DDoS-атакам передбачають реагування на трафік, який генерується ботами або автоматизованими системами. Однак у гібридному середовищі значна частина навантаження створюється легітимними користувачами, які реагують на дезінформацію, поширену через месенджери, соціальні мережі чи анонімні канали. Це створює серйозні виклики для систем захисту, оскільки поведінкова активність не є злочинною за суттю, але може мати руйнівні наслідки для стабільності інфраструктури.

Актуальність дослідження посилюється з огляду на реальні інциденти, які відбувалися під час активних фаз гібридної війни в Україні. Наприклад, фейкові повідомлення про “злам” державних цифрових сервісів 2022 р. призводили до хвильових навантажень на інфраструктуру, спричинених масовим входом, змінами паролів або надсиланням запитів підтримки. Це свідчить про необхідність моделювання таких процесів не лише на рівні мережного трафіка, а і з урахуванням поведінки користувачів. Швидкість і масштаб поширення фейкової інформації є викликом не лише для технічного, а і для соціального контролю за інформаційними потоками (David et al., 2018).

В таких умовах виникає потреба у моделюванні гібридного впливу з урахуванням ролей окремих агентів: користувачів, джерел фейків, ботів і захисних систем. Агентно-орієнтоване моделювання (ABM – Agent-Based Modeling) дає змогу реалізувати таку багатошарову систему. Цей підхід дозволяє враховувати індивідуальну реакцію агентів, часову затримку в активності, ефект від спростування фейків і можливість комбінованого навантаження.

Метою цієї роботи є розроблення й апробація агентно-орієнтованої моделі виявлення і стримування гібридного впливу, зокрема й поведінкової активності користувачів, викликаної інформаційними атаками. Запропонована модель поєднує класичні компоненти аналізу навантаження з реакцією на інформаційні стимули та дозволяє формувати сценарії, в яких навіть невелика кількість фейкових повідомлень може призвести до системного збою. Це створює умови, за яких без технічного втручання цифрові системи можуть зазнавати суттєвого навантаження внаслідок неконтрольованої активності користувачів.

Огляд літератури. Одним із критичних факторів у контексті гібридного впливу є динаміка та тип реакції користувачів на інформаційні повідомлення. У дослідженні Glenski, Weninger, & Volkova (2018) запропоновано нейронну модель, що класифікує реакції користувачів на новини в соціальних мережах за дев'ятьма типами, зокрема такими: запитання, уточнення, вдячність, емоційна оцінка, заперечення або гумор. Така деталізація дозволила авторам виявити відмінності у сприйнятті достовірних і дезінформаційних джерел, що також впливає на швидкість та інтенсивність реакцій.

Аналіз понад 10 млн твітів і 6 млн коментарів Reddit показав, що реакції на недостовірні новини часто

ISSN 2707-1758

мають іншу природу – наприклад, у відповідь на фейкові новини користувачі частіше демонструють вдячність або задають уточнювальні запитання, тоді як достовірні джерела викликають глибші аналітичні відповіді чи емоційне схвалення.

Зазначене дослідження підтверджує, що моделювання поведінки користувачів у контексті гібридних загроз не повинно зводитися до бінарної моделі "активовано / не активовано". Натомість, доцільним є використання функціональних імовірнісних моделей, які враховують різноманітність реакцій – саме це і було реалізовано у запропонованій агентно-орієнтованій моделі. Shu et al. (2019) підкреслюють, що ефективне виявлення фейкової інформації на соціальних платформах неможливе без урахування трьох взаємопов'язаних аспектів: змісту, реакцій користувачів і характеристик джерел.

У дослідженні Dhiman Shah, Thirunarayan (2024) запропоновано гібридну модель GBERT (GPT-BERT), яка поєднує архітектури BERT (Bidirectional Encoder Representations from Transformers) і GPT (Generative Pre-trained Transformer) для підвищення точності виявлення фейкових новин. Модель поєднує дві найсучасніші технології: BERT відповідає за глибокий контекстуальний аналіз тексту, а GPT – за виявлення довгострокових семантичних зв'язків і логічної узгодженості повідомлень. Отримане поєднання протестовано на двох реальних наборах даних із Twitter і Reddit, де продемонструвало підвищену точність класифікації фейкових повідомлень порівняно з традиційними архітектурами. Такий підхід підтверджує доцільність поєднання контент-орієнтованого аналізу з мовними моделями у разі виявлення дезінформації – саме це лягло в основу структури поведінкової оцінки в нашій моделі.

Як показують результати дослідження Tafur, & Sarkar (2023), на формування думки користувачів щодо достовірності інформації впливають не лише зміст повідомлення чи джерело публікації, а й сигнали, які надходять від автоматизованих систем верифікації. У проведеному експерименті із 40 учасниками виявлено, що навіть за наявності помилок із боку системи класифікації фейкових новин, користувачі продовжували покладатися на її висновки у понад 57 % випадків. Така поведінка свідчить про високий рівень довіри до алгоритмів і демонструє важливу роль соціального контексту, когнітивних упереджень і настанов у процесі прийняття рішень. Ці висновки важливі для побудови агентно-орієнтованих моделей, у яких реакція користувача моделюється не лише як відповідь на контент, а і як результат взаємодії зі складною інформаційною екосистемою.

Агентно-орієнтовані моделі виявилися ефективними для моделювання складних загроз у кіберпросторі. У роботі Kotenko Stepashkin, & Doynikova (2008) запропоновано архітектуру симуляційного середовища для аналізу атак ботнетів і механізмів їхнього стримування. Автори представили динамічну взаємодію між агентами-атаками (ботами) та агентами-захисниками, що моделювали координацію у межах багаторівневих команд. У фокусі дослідження – протистояння DDoS-атакам із використанням сценаріїв самонавчання, фільтрації та обміну сигнатурами між агентами. Такий підхід близький до реалізованої в нашій моделі логіки: поєднання поведінкових агентів, системи блокування та реагування у складному інформаційному середовищі. Це свідчить про сталість підходу до використання агентних структур у моделюванні кіберзагроз, зокрема і в разі врахування поведінки ботмереж та адаптивних захисних стратегій.

Методи

У моделі розглядають сукупність агентів, кожен з яких відіграє окрему роль у сценарії гібридного впливу. Кожен агент має набір поведінкових правил і здатний змінювати стан системи. Основні типи агентів наведено в табл. 1.

Таблиця 1

Основні типи агентів у моделі

Агент	Позначення	Поведінка
Користувач	Ac	Реагує на фейкову інформацію, генерує запити до сервісу
Telegram-канал	At	Поширює фейкові повідомлення
Бот/DDoS-агент	Ab	Імітує масові запити на перевантаження
Державний сервіс	As	Обробляє запити, має обмежену пропускну здатність
Захисна система	Az	Реагує на навантаження, блокує джерела трафіка

Імовірність активації користувача розраховують як

$$P_c = \alpha \cdot D_t + \beta \cdot V_t,$$

де P_c – імовірність того, що користувач здійснить дію (перехід, запит), D_t – ступінь довіри до джерела, $V(t)$ – інтенсивність поширення повідомлення.

Розглянемо формулу

$$\alpha, \beta \in [0; 1], \quad \alpha + \beta = 1.$$

Ця формула дозволяє змоделювати ймовірність того, що користувач повірить у фейкову інформацію та здійснить активну дію (напр., перехід на сервіс). Параметри D_t та V_t відповідають за рівень довіри до джерела повідомлення та його видимість у соціальному середовищі відповідно. Значення α та β задають вагу кожного чинника. Таке представлення дозволяє моделювати поведінкову залежність користувача в умовах інформаційного тиску.

Загальне навантаження на сервіс визначають як

$$L(t) = N_u \cdot p(t) \cdot r_u + N_b \cdot r_b,$$

де N_u – кількість користувачів, N_b – кількість ботів, $p(t)$ – імовірність активації користувачів у момент часу (t), r_u, r_b – інтенсивність запитів від користувача та бота відповідно.

Критичне навантаження, за якого сервіс відмовляє:

$$L(t) > L_{\text{крит}} \Rightarrow \text{сервіс у стані відмови.}$$

ISSN 2707-1758

Це рівняння моделює загальне навантаження на державний сервіс у конкретний момент часу. Воно враховує як запити від звичайних користувачів (параметри N_u, r_u, $p(t)$), так і від ботів, які виконують DDoS-атаки (N_b, r_b). Під час перевищення критичного порогу навантаження $L_{\text{крит}}$, система може перейти в нестабільний стан або припинити оброблення запитів. Такий підхід дозволяє виявляти порогові умови відмови в межах змодельованого сценарію.

Модель дає змогу відтворити сценарії, за яких внаслідок активності користувачів відбувається різке зростання запитів до сервісу. У табл. 2 представлено один із таких сценаріїв: у початковій фазі (0–19 хв) система функціонує стабільно, однак після активізації фейку (20–30 хв) навантаження зростає, що свідчить про ризик перевантаження навіть без технічної атаки з боку ботів.

Таблиця 2

Сценарій зміни навантаження

Час, хвилини	Активовані користувачі, %	Запити від користувачів	Запити від ботів	Загальне навантаження	Стан сервісу
0–19	12	1200	2000	3200	Стабільна робота
20+	25	2500	2000	4500	Підвищене навантаження
30+	30	3000	2000	5000	Підвищене навантаження

Наступний сценарій моделює ситуацію, у якій після початкового навантаження на сервіс відбувається активація захисної системи. Агент Az виявляє зростання трафіка та починає блокування частини джерел, що підозрюються у генерації фейкових запитів. У моделі враховується як вплив бота, так і реакція користувачів на фейкове повідомлення. Таблиця 3 ілюструє, як змінюється рівень загального навантаження та стан сервісу після втручання системи захисту.

Таблиця 3

Сценарій із реакцією захисної системи

Час, хвилини	Активовані користувачі, %	Запити від користувачів	Запити від ботів	Заблоковані джерела	Загальне навантаження	Стан сервісу
0–19	12	1200	2000	0	3200	Стабільна робота
20+	25	2500	2000	500	4000	Підвищене навантаження
30+	30	3000	2000	1000	4000	Підвищене навантаження

У базовій моделі користувачі реагують на фейкову інформацію миттєво після її отримання. Проте на практиці реакція часто є відкладеною: деякі користувачі спершу спостерігають, очікують підтвердження з інших джерел або активізуються лише після повторного контакту з фейком. З огляду на цей ефект, у модель додано сценарій із часовою затримкою активації.

Імовірність переходу користувача до активних дій описується логістичною функцією

$$p(t) = \frac{1}{1 + e^{-k(t-\tau)}},$$

де t – час затримки, а k – коефіцієнт швидкості реагування.

У сценарії прикладу активність користувачів поступово зростає впродовж перших 24 хв: від 10 % до 25 %. Це створює умови для поступового накопичення навантаження, яке досягає критичного рівня не відразу, а лише в кінці інтервалу. Такий сценарій дозволяє моделювати ефект уповільненого піку навантаження та необхідність більш ранньої реакції системи. Детальну динаміку навантаження в цьому сценарії представлено у табл. 4.

Таблиця 4

Сценарій затриманої активації

Час, хвилини	Активовані користувачі, %	Запити від користувачів	Запити від ботів	Загальне навантаження	Стан сервісу
0–9	10	1000	2000	3000	Стабільна робота
10–14	15	1500	2000	3500	Стабільна робота
15–19	20	2000	2000	4000	Підвищене навантаження
20–24	25	2500	2000	4500	Підвищене навантаження

ISSN 2707-1758

Ще одним сценарієм, реалізованим у моделі, є інформаційне втручання з боку держави після виявлення пікового навантаження. Офіційне спростування може бути реалізоване через повідомлення на державному порталі, офіційні сторінки у соціальних мережах або push-сповіщення у мобільному додатку. Цей сценарій враховує зміну функції активації користувачів у часі. До 20-ї хвилини ймовірність $p(t)$ зберігається на рівні 0.3, після чого знижується до 0.1 у результаті втручання. Це моделюється як кускова функція:

$$p(t) = 0.3, \text{ якщо } t < 20\,,$$
$$0.1, \text{ якщо } t \geq 20\,.$$

У моделі також враховано дію агента Az, який реалізує технічне блокування підозрілих джерел після моменту втручання. Таблиця 5 ілюструє, як змінюється навантаження: спершу воно досягає порога перевантаження, але згодом стабілізується завдяки одночасному зниженню поведінкової активності користувачів і технічному блокуванню джерел.

Зазначений підхід дозволяє враховувати не лише технічні механізми захисту, але й вплив інформаційного середовища. За даними ENISA (2023), оперативне інформування громадян і контрнаративи з боку офіційних джерел можуть суттєво знизити поведінкову реакцію на шкідливі інформаційні кампанії. Відповідно, інтеграція подібної логіки у технічні моделі є обґрунтованим кроком для сценарного аналізу гібридних загроз.

Таблиця 5

Зміна активності користувачів після спростування

Час, хвилини	Активовані користувачі, %	Запити від користувачів	Запити від ботів	Заблоковані джерела	Загальне навантаження	Стан сервісу
0–9	30	3000	2000	0	5000	Стабільна робота
10–14	30	3000	2000	0	5000	Підвищене навантаження
15–19	30	3000	2000	0	5000	Підвищене навантаження
20–24	10	1000	2000	500	2500	Стабілізація

Результати

Сценарне моделювання дозволило оцінити, як різні варіанти реагування користувачів і системи захисту впливають на навантаження державного інформаційного сервісу. Загалом змодельовано чотири сценарії: базовий без втручання, базовий із втручанням захисної системи, сценарій із затриманою реакцією користувачів і сценарій з інформаційним спростуванням.

У базовому сценарії (табл. 2) користувачі миттєво реагують на фейкове повідомлення, що призводить до поступового зростання навантаження. Навіть якщо DDoS-трафік залишається незмінним (2000 запитів), збільшення частки активованих користувачів до 30 % створює додаткове навантаження у 3000 запитів, що в сукупності становить 5000 – критичний поріг для сервісу. Це свідчить про те, що навіть без технічної атаки система може зазнати відмови внаслідок поведінкової активності легітимних користувачів.

У сценарії з активною захисною системою (табл. 3), після виявлення перевищення навантаження агент Az блокує частину джерел запитів. Блокування 1000 одиниць трафіка дозволяє знизити загальне навантаження до 4000, що стабілізує роботу сервісу. Це підтверджує, що навіть обмежене технічне реагування здатне стабілізувати систему, особливо якщо воно здійснюється до досягнення критичного рівня.

У сценарії із затриманою реакцією (табл. 4) активація користувачів відбувається не миттєво, а поступово – від 10 до 30 % упродовж 30 хв. Це веде до відкладеного накопичення навантаження: початкові фази є відносно безпечними, але у фінальній фазі сукупна кількість запитів досягає того самого рівня 5000, що й у базовому сценарії. Водночас повільне зростання навантаження ускладнює виявлення загрози для систем, які орієнтовані на миттєві DDoS-сплески. Така поведінка вимагає гнучкішого реагування захисних систем на поведінкову динаміку.

У сценарії з інформаційним втручанням (табл. 5) реакція користувачів також досягає пікових значень до 20-ї хвилини, однак подальше публічне спростування знижує $p(t)$ до 0.1. Одночасно із цим агент Az блокує частину джерел, що спричиняє швидке зменшення навантаження – з 5000 до 3000 запитів. У результаті система стабілізується навіть без потреби в глибокому технічному втручанні. Такий сценарій підтверджує практичну ефективність комбінації інформаційної протидії та точкового блокування для захисту від гібридних атак.

Порівняльний аналіз усіх чотирьох сценаріїв дозволяє зробити висновок, що найефективнішим із погляду стримування навантаження є комбінований підхід – технічне блокування джерел разом з інформаційним втручанням (табл. 5). Саме в цьому випадку вдалося знизити кількість запитів до 3000, тоді як у базовому сценарії цей показник сягав критичного рівня 5000. Навпаки, сценарій із затриманою реакцією (табл. 4) продемонстрував, що навіть без ботатаки, поступове зростання $p(t)$ може призвести до подібного навантаження. Це доводить, що не лише технічна, але й поведінкова динаміка має ключове значення для стабільності сервісу. Водночас сценарій із частковим втручанням (табл. 3) забезпечує лише часткове зниження – до 4000 запитів, що є межовим. Відповідно, саме мультикомпонентні сценарії демонструють найкращі результати з погляду підтримання стабільної роботи сервісу.

Отже, результати демонструють, що агентно-орієнтована модель дозволяє не лише імітувати поведінкові й технічні загрози, а й протестувати ефективність різних тактик реагування. Усі сценарії підтверджують, що швидкість і форма відповіді системи захисту мають критичне значення для запобігання відмові сервісу.

Обговорення. Отримані результати свідчать про критичний вплив поведінкової активності користувачів на навантаження цифрових сервісів публічного сектору. У всіх сценаріях (табл. 2–5) підтверджено, що навіть без

ISSN 2707-1758

змін у DDoS-компоненті моделі (кількість ботів лишалася сталою), реакція користувачів на фейкові повідомлення здатна вивести систему за межі її працездатності. Це демонструє необхідність урахування поведінкових механізмів в оцінюванні кіберзагроз.

Зокрема, сценарії із затриманою реакцією (табл. 4) й інформаційним втручанням (табл. 5) демонструють складнішу динаміку навантаження. У першому випадку – система перебуває в зоні ризику через кумулятивне зростання запитів, у другому – стабілізації досягають лише за умови одночасного впливу інформаційного спростування та блокування джерел. Ці сценарії підтверджують, що класичні DDoS-захисти, орієнтовані на миттєві сплески, можуть бути не ефективними у випадках поведінкових хвиль або комбінованих атак.

Агентно-орієнтований підхід дозволив імітувати взаємодію різних елементів – користувачів, атакуючих, каналів поширення та систем захисту – у єдиній моделі. Завдяки цьому змодельовано не лише технічні, а й інформаційно-поведінкові впливи. Таке поєднання є особливо актуальним у контексті гібридної війни, де атака може відбуватися без жодного коду чи вірусу – лише через маніпуляцію сприйняттям.

Реалістичність змодельованих сценаріїв підтверджується на практиці. У 2022 р. в українському інформаційному просторі поширився фейк про нібито злам додатку “Дія” та витікання даних. Заклики негайно перевірити облікові записи спричинили різкий сплеск активності, який не був пов'язаний із реальним технічним оновленням. Лише після публічного спростування з боку держави навантаження нормалізувалося. Цей кейс демонструє, що навіть без атак ботів система може зазнати DDoS-подібного навантаження виключно через поведінку користувачів.

На відміну від класичних моделей DDoS-аналізу, що фокусуються виключно на технічних метриках (напр., кількість запитів, обсяг трафіка або швидкість генерації навантаження), запропонована модель враховує динаміку поведінки користувачів як окремий чинник впливу. Це дозволяє виявити сценарії, які є малопомітними в класичній аналітиці, але можуть становити не меншу загрозу для стабільності системи. Зокрема, сценарій із затриманою реакцією демонструє, що навіть за відсутності технічної атаки, поступове накопичення активності може призвести до перевантаження.

У практичному контексті це означає, що системи моніторингу, орієнтовані лише на пікові навантаження, можуть виявитися неефективними за наявності поведінкових хвиль. Агентно-орієнтована модель, навпаки, дозволяє адаптувати стратегії захисту до різних форм дестабілізації – від миттєвих фальстартів до довготривалих інформаційних кампаній.

Подібний підхід реалізовано в гібридній моделі CSI (Ruchansky, Seo, & Liu, 2017), яка інтегрує контент повідомлення, реакції користувачів і характеристики джерел. Це підтверджує ефективність багатофакторного моделювання для виявлення фейкової активності без необхідності графового аналізу.

Крім дослідницького потенціалу, модель може бути інтегрована в прикладні інструменти оцінювання ризиків і навантаження. Зокрема, вона може використовуватись в аналітичних модулях платформ типу CERT-UA для симуляцій поведінкової активності користувачів під час фейкових інформаційних кампаній. Іншим можливим напрямом є використання моделі як компонента в системах раннього попередження про аномальне навантаження – з урахуванням не лише технічних показників, а й інформаційного контексту. Крім того, модель може стати основою для сценаріїв Red Team / Blue Team навчань у сфері кібербезпеки, де важливо протестувати і технічну стійкість систем, і комунікаційну реакцію на загрозу.

Попри переваги агентно-орієнтованого підходу, запропонована модель має певні обмеження, що варто враховувати при її інтерпретації та подальшому застосуванні. По-перше, у моделюванні використано узагальнені параметри $p(t)$, які наближено відображають поведінку користувачів, але не враховують міжіндивідуальні відмінності, наприклад, вплив демографічних чи психологічних характеристик. По-друге, модель передбачає фіксовану кількість ботагентів і не враховує можливість їхнього саморозширення або адаптації, що властиво сучасним ботнетам. Крім того, інформаційне середовище змодельовано як умовно однорідне, без диференціації за типами каналів або рівнем довіри. Подальші дослідження можуть зосередитись на розширенні моделі за рахунок факторів соціального впливу, багатоканального поширення та реалізації агентів з адаптивною логікою поведінки.

Одним із факторів, який потенційно підсилює вплив фейкової інформації, є соціальний резонанс – ефект, коли користувач не реагує на повідомлення у першому контакті, але активізується після того, як аналогічну інформацію побачить у кількох джерелах. Цей ефект важко моделювати у класичних технічних системах, але його можна відтворити через багатофазну поведінку агентів. Повторне зіткнення з повідомленням, посилене авторитетом джерела або соціальним тиском, може істотно змінити траєкторію поведінки. У роботі Starbird, Arif, Wilson (2019) підкреслено, що дезінформація поширюється не лише як технічний вплив, а і як колективна діяльність аудиторії, яка неусвідомлено підтримує життєвий цикл шкідливого контенту.

У контексті моделі, що розроблена у цьому дослідженні, подібна логіка може бути реалізована завдяки розширенню функції активації $p(t)$ залежно від кількості повторів або "відлуння" повідомлення. Такий підхід дозволяє наблизити симуляцію до реального інформаційного середовища, де не лише контент, але і частота контакту з ним визначає ймовірність дії. Це є потенційним напрямом подальшого розвитку моделі – з урахуванням циклів повторення та міжагентного поширення інформації.

Крім симуляційного аналізу, запропоновану модель можна адаптувати для використання у системах моніторингу в режимі реального часу. Один із перспективних напрямів – адаптація логіки агентної моделі для систем попередження інформаційних аномалій. Це особливо актуально у випадках, коли поширення дезінформації не супроводжується класичними технічними індикаторами (напр., DDoS або мережними скануваннями), але створює високий рівень навантаження через реакцію аудиторії.

Аналіз динаміки поведінки користувачів, зокрема і різких змін у $p(t)$, дозволяє розпізнавати хвильові аномалії, які можуть указувати на запуск скоординованої фейкової кампанії. У таких випадках система може ініціювати автоматизоване спростування або блокування поширювачів – навіть до того, як

ISSN 2707-1758

з'являться технічні наслідки. Отже, модель може функціонувати не лише як інструмент для планування і навчання, але і як практичний компонент у системах кіберзахисту, де швидкість реакції має вирішальне значення. Подальші дослідження можуть бути спрямовані на визначення оптимальних порогових значень $p(t)$ для запуску таких механізмів залежно від типу інформаційної платформи та характеристик аудиторії.

Крім того, модель може бути використана для виявлення потенційно вразливих інформаційних сегментів наприклад, груп користувачів із високим $p(t)$, які найімовірніше піддадуться впливу фейкових повідомлень. Це відкриває можливості для превентивного інформування, спрямованого на посилення кіберстійкості цільових аудиторій.

Запропонована модель дозволяє не лише відтворити загрозу, але й перевірити ефективність різних тактик реагування. Це робить її корисною не лише для наукового аналізу, а й для прикладного використання – наприклад, у межах Red Team / Blue Team навчань, для оцінювання ризиків у CERT-платформах або підготовки до інформаційних кампаній. Модель також може бути розширена: з урахуванням ефекту масового повторення, затримок у поширенні або впливу ключових лідерів думок. Такі модифікації створять основу для адаптивних систем прогнозування поведінки користувачів під час криз.

Дискусія і висновки

Запропонована агентно-орієнтована модель виявлення та стримування гібридного впливу дозволяє комплексно моделювати взаємодію користувачів, ботагентів, джерел дезінформації та захисних систем. На відміну від класичних DDoS-моделей, вона враховує поведінкову динаміку й інформаційні ефекти, зокрема й затримку реакції користувачів і вплив офіційного спростування.

Сценарне моделювання показало, що навіть легітимна поведінка користувачів, спровокована фейком, здатна створити критичне навантаження на державний сервіс. Водночас своєчасне інформаційне втручання або точкове блокування джерел дозволяють стабілізувати ситуацію без масштабного технічного реагування. Це підкреслює важливість гібридного підходу до кіберзахисту – поєднання технічних і комунікаційних засобів реагування.

Модель узгоджується із сучасними підходами до виявлення фейкової активності: концепції багатофакторного аналізу, реалізовані в CSI, поведінкове розгалуження реакцій користувачів (Glenski et al.). Заявлене вище, а також вплив довіри до верифікаційних систем підтверджують доцільність інтегрованого моделювання поведінки, контенту та технічної динаміки.

Практична цінність моделі полягає в можливості її використання для аналізу навантаження в системах CERT, симуляцій Red Team / Blue Team, а також в оцінюванні ризиків під час дезінформаційних кампаній. Її можна адаптувати до різних конфігурацій: як для окремих сервісів (напр., електронних реєстрів), так і для масштабніших цифрових платформ.

У перспективі модель може бути доповнена модулями довіри, соціального поширення, впливу лідерів думок та алгоритмами прогнозування реакцій у часі. Це розширить її функціональність і дозволить використовувати як компонент в адаптивних системах інформаційного захисту.

Отже, ключовим елементом у формуванні ризику перевантаження виявляється не сам факт атаки, а форма сприйняття та реакція аудиторії. Навіть мінімальна технічна активність, поєднана з панікою або дезінформуванням, може мати наслідки, еквівалентні повномасштабній DDoS-атаці. Це підкреслює необхідність урахування поведінкових чинників до моделей ризику в інформаційній безпеці.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Matteo, C., Walter, Q., Alessandro, G., Valensise, Carlo, M., Emanuele, B., Schmidt, Ana, L., Paola, Z., Zollo F. & Scala, A. (2020). The COVID-19 social media infodemic. *Scientific Reports*, 10, 16598. https://doi.org/10.1038/s41598-020-73510-5

Dhiman, D., Shah, R., Thirunarayan, K. (2024). Detecting fake news using a hybrid GBERT model. *Procedia Computer Science,* 217, 1427–1436. https://doi.org/10.1016/j.procs.2022.12.341

ENISA. (2023). *Threat Landscape for Disinformation Campaigns.* European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-for-disinformation-campaigns

Glenski, M., Weninger, T., & Volkova, S. (2018). Identifying and understanding user reactions to deceptive and trusted social news sources. In K. Bharat, R. Jim, & G. Kathleen (Eds.). *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 1063–1072). Association for Computing Machinery. https://doi.org/10.1145/3269206.3271774

Kotenko, I., Stepashkin, M., & Doynikova, E. (2008). Agent-based modeling and simulation of botnets and botnet defense. In A. Smith, & B. Johnson (Eds.). *Proceedings of the 2008 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing* (pp. 71–78). IEEE Press. https://doi.org/10.1109/PDP.2008.71

David M. J., Lazer, Matthew A., Baum, Yochai, B., Adam J. B., Kelly M. Greenhill, F. M., Miriam J. Metzger, B., N., Gordon P., & Jonathan L., Z. (2018). The science of fake news. *Science*, 359(6380). https://doi.org/10.1126/science.aao2998

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A hybrid deep model for fake news detection. In J. Doe, & A. Smith (Eds.). *Proceedings of the 2017 ACM Conference on Information and Knowledge Management* (pp. 797–806). Association for Computing Machinery. https://doi.org/10.1145/3132847.3132877

Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H. (2019). Fake news detection on social media: A data mining perspective. *SIGKDD Explorations*, 19(1), 22–36. https://doi.org/10.1145/3137597.3137600

Starbird, K., Arif, A., Wilson, T. (2019). Disinformation as collaborative work: Surfacing the participatory nature of strategic information operations. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW). https://doi.org/10.1145/3359229

References

Matteo, C., Walter, Q., Alessandro, G., Valensise, Carlo, M., Emanuele, B., Schmidt, Ana, L., Paola, Z., Zollo F. & Scala, A. (2020). The COVID-19 social media infodemic. *Scientific Reports*, 10, 16598. https://doi.org/10.1038/s41598-020-73510-5

Dhiman, D., Shah, R., Thirunarayan, K. (2024). Detecting fake news using a hybrid GBERT model. *Procedia Computer Science,* 217, 1427–1436. https://doi.org/10.1016/j.procs.2022.12.341

ENISA. (2023). *Threat Landscape for Disinformation Campaigns.* European Union Agency for Cybersecurity. https://www.enisa.europa.eu/publications/enisa-threat-landscape-for-disinformation-campaigns

Glenski, M., Weninger, T., & Volkova, S. (2018). Identifying and understanding user reactions to deceptive and trusted social news sources. In K. Bharat, R. Jim, & G. Kathleen (Eds.). *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 1063–1072). Association for Computing Machinery. https://doi.org/10.1145/3269206.3271774

Kotenko, I., Stepashkin, M., & Doynikova, E. (2008). Agent-based modeling and simulation of botnets and botnet defense. In A. Smith, & B. Johnson (Eds.). *Proceedings of the 2008 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing* (pp. 71–78). IEEE Press. https://doi.org/10.1109/PDP.2008.71

David M. J., Lazer, Matthew A., Baum, Yochai, B., Adam J. B., Kelly M. Greenhill, F. M., Miriam J. Metzger, B., N., Gordon P., & Jonathan L., Z. (2018). The science of fake news. *Science*, 359(6380). https://doi.org/10.1126/science.aao2998

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A hybrid deep model for fake news detection. In J. Doe, & A. Smith (Eds.). *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*

ISSN 2707-1758

(pp. 797–806). Association for Computing Machinery. https://doi.org/10.1145/3132847.3132877

Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H. (2019). Fake news detection on social media: A data mining perspective. *SIGKDD Explorations*, 19(1), 22–36. https://doi.org/10.1145/3137597.3137600

Starbird, K., Arif, A., Wilson, T. (2019). Disinformation as collaborative work: Surfacing the participatory nature of strategic information operations. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW). https://doi.org/10.1145/3359229

Отримано редакцією журналу / Received: 01.06.25
Прорецензовано / Revised: 27.06.25
Схвалено до друку / Accepted: 30.06.25

Dmytro DZHENDZHERO, PhD Student
ORCID ID: 0009-0006-7394-4995
e-mail: dzhendzherod@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

A METHOD FOR MODELING HYBRID INFLUENCE ON GOVERNMENT SERVICES USING AN AGENT-BASED APPROACH

Background. ***In the context of hybrid warfare, the role of information influence on critical infrastructure is growing. Fake news can trigger waves of user behavioral activity that mimic DDoS attacks, overloading government digital services even without direct technical interference. This creates the need to model such scenarios considering both technical and socio-behavioral factors.***

Methods. ***An agent-based model of hybrid influence is proposed, including users, bots, fake news sources, and protective mechanisms. The model is implemented using mathematical expressions and scenario simulation, including a baseline scenario, delayed response scenario, technical intervention scenario, and official rebuttal scenario. The analysis considers system load parameters, behavioral activation probability, and the system's response.***

Results. ***It was found that user behavior alone can generate critical system load equivalent to a DDoS attack. The most effective scenario combined technical blocking and information rebuttal, reducing load to a stable level. Classic cybersecurity models often fail to account for delayed reactions or wave-like behavioral patterns, limiting their effectiveness in hybrid environments.***

Conclusions. ***The proposed model enables not only simulation of hybrid influence but also evaluation of response tactics. It can be applied in CERT platforms, Red Team / Blue Team exercises, and early-warning systems. Further research may focus on adapting the model to real-world environments and developing automated response mechanisms to information threats.***

Keywords: ***hybrid warfare, agent-based modeling, fake news, digital security, behavioral load.***

Автор заявляє про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

UDC 004.056.54:519.7
DOI: https://doi.org/10.17721/ISTS.2025.9.54-60

Yevhenii PONOMARENKO, PhD Student
ORCID ID: 0009-0000-2647-3381
e-mail: allagrir@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Oleksandr LAPTIEV, DSc (Engin.), Assoc. Prof., Senior Researcher
ORCID ID: 0000-0002-4194-402X
e-mail: olaptiev@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

MATHEMATICAL MODEL OF STEGANOGRAPHY USING SUBOPTIMAL DECISIONS IN DATA COMPRESSION ALGORITHMS

Background. *Protecting access to information in the digital age requires the development of digital tools to encrypt and hide the information from everyone who is not meant to be able to access it. While encryption is great at preventing unauthorized people from accessing the information, it lets people know that there is something hidden behind the transformation. Steganography instead allows us to obscure the fact of information concealment in the first place. Unfortunately, common types of steganography rely on altering the source signal, leaving subtle footprints of data manipulation that can be traced and detected. The goal of this research paper is to provide a model that has the potential of preserving the source signal in its entirety, instead relying on changing the way the source signal is represented in digital media in a way that allows us to encode secret data into the output stream.*

Methods. *A theoretical analysis of steganography approaches on data compression algorithms was conducted. Methods of preserving source data stream were investigated.*

Results. *A new model has been developed that uses decision-making processes in data compression algorithms to encode steganographic data in a resulting data stream. In lossless data compression schemes, it is possible to achieve perfect reproduction of source data stream, making detection through analysis of underlying signal encoded in data compression algorithms useless.*

Conclusions. *The need for information security has been increasing over time, with tensions between countries resulting in new armed conflicts around the world. The ability to embed and covertly send data through steganography may provide a competitive economic, political, and/or military edge. The developed model can be applied to further develop specific methods of steganographic data encoding that is resilient to analysis and detection by existing approaches that rely on statistical analysis of underlying signal stream.*

Keywords: *steganography, data compression algorithms, signal processing, data security, decision trees, entropy encoding, arithmetic encoding.*

Background

With the advent of digital computers, a need to store information about the real world in digital form emerged. This information takes a lengthy and complex route in order to turn from physical phenomena of a light bouncing off of the surface of a physical object or pressure waves travelling through the air into a string of numbers, zeroes and ones, that can be stored in computer's memory, and then it needs additional work to be turned from those back into physical light, sound, or other form of real-life, analog phenomena – or even objects. Photosensitive diodes turn photons of light into electrical signals that we can register and process, allowing us to take a photograph of the real world. A transducer turns pressure waves into digital signals, allowing us to record sound.

Of course, without a way to represent this information in a digital, binary form, there is no way we can store this information on computer systems. For photographic signals, we rasterize visual data, storing color information in discrete, individual cells called pixels. This means that we quantize the infinite precision of the real world in two ways: spacial and spectral. Rasterized pixels would be the spacial quantization, while the color data of those pixels – being limited to a finite number of bits we can use to store color information in the pixel – would be spectral. For sound, the quantization would be temporal and amplitude: the former splits the infinitely precise stretches of time into distinct (and usually equal) durations of time, during which a single intensity of the sound signal is assumed. This signal intensity, just like with pictures, could only be encoded with a limited number of bits, leading to amplitude quantization. Video signals, therefore, are usually nothing more than a sequence of pictures presented at equal intervals of time between one another, synchronized with a sound signal coming from one or more sources in physical space. There is a temporal quantization happening between frames, with values of dozens of frames per second being common.

All these encoding solutions are made with decoding and reproduction back into physical phenomena in mind. A digital encoding of a photo has no use if we cannot present it back into a grid of colors that we can perceive in the real world with our human eyes. A sound recording is useless if we cannot listen back to it. The process of both capturing analog signals and capturing them into digital form, and reproducing these digital signals back into physical realm is imperfect, since the components we use to both transform light, sound, and other sensations into electrical signals, and perform the inverse of that, are subject to distortions, noise, lack of performance, bias, and other flaws – though the quality of devices to both capture and reproduce those signals has been improving since their inception, and continues to this day (Galal, 2016; Rumsey, & Mccormick, 2013, pp. 201–256).

Storing information in digital form, however, poses a challenge of providing enough storage capacity for a given media. Storing color data of a single pixel in a 24-bit image takes up three bytes of storage. Multiply that by multiple megapixels that a commonly used 1080p display would use, and you arrive at approximately six megabytes of raw picture data. Videos require thirty of these frames per second, which quickly turns into a gigabyte of storage capacity being taken up by 5–6 seconds of raw video stream. This explosion of storage capacity that is necessary for storing modern pieces of media means that we must somehow reduce the amount of data our media object requires to store in digital form. In other words, we need to compress digital data.

Modern schemes of digital compression can reach file size reduction of dozens of times, if not more (Barina, 2021; Öztürk, & Mesut, 2021, pp. 15–20) while preserving most (if not all) of the detail of the original, uncompressed

ISSN 2707-1758

data stream. To achieve high compression ratios, compression algorithms adapt their approach for various parts of the source media. For example, when compressing sound, an algorithm might dedicate more bandwidth to a more intense part of a song with lots of different instruments all playing at once, and less bandwidth to sections with silence or quiet sounds. In other words, they make decisions based on the source data stream. Encoding schemes have a limited number of ways to decide how to compress a certain primitive structure of the data stream, like a block of pixels or a set of audio samples. Commonly, they tend to select the best ways of encoding the data that they expect would take the least number of bits. In the end, these compression algorithms produce a smaller, more compact representation of source media, potentially with some losses in perceptual fidelity that the algorithm authors considered acceptable for a given compression ratio. Importantly, these compressed representations can be decompressed and turned back into a raw stream of data that can be presented to the user in the form of a picture, sound, or video.

With established methods of representing – and compressing – analog media in digital formats, we can shift our focus to another need: the need to covertly transmit information. There are two ways we can achieve this: encryption and steganography. Encryption is the process of converting readable information into unreadable information to prevent unauthorized access. Steganography is the process of embedding secret data within another object in a way that conceals the presence of secret data from an unsuspecting observer. Encryption is great at preventing people from accessing information they should not have access to, but encrypted data on its own is obvious, anyone can see there is something there, painting a target on it. After all, why would somebody encrypt something that is not worth protecting? Steganography, on the other hand, does not, on its own, prevent others from accessing the data, but instead attempts to hide it in plain sight, preventing people from knowing it is even there unless they know about it.

There are many different approaches to steganography, and they depend on the cover media used. Common types of cover media include pictures, video, audio, and text (Anas, Ridzuan, & Pitchay, 2025). Broadly, these techniques tend to either modify the source data directly (i.e., manipulating pixel color or sound intensity values by changing their least significant bits (LSB)) or indirectly through manipulating values in the frequency domain (Apau et al., 2024). This leads to a material change to the underlying representation of media; in other words, after transformation, the raw data that is transmitted to the screen, played back through the speakers, or printed out on a sheet of paper, is different from the original data of the cover media. Though there exist adaptive techniques that utilize machine learning, artificial intelligence, blockchains, and other tools to prevent existing steganalysis and statistical analysis techniques from being able to detect steganography (Apau et al., 2024, p. 5), they still change the apparent representation of carrier media. This leaves a gap in current research, as it is difficult to find a steganography method that would leave the apparent representation unchanged.

The objective of this research, therefore, is to develop a new model of steganographic encoding of data that does not change the apparent representation of some, if not all, media objects.

Methods

In this article, we performed theoretical analysis and systematization of data compression algorithms and ways of exploiting their decision-making processes to encode a secondary, secret message within the encoded cover media such that apparent result is not changed. Methods of preserving the apparent representation of carrier media were investigated.

Results

There are many ways to represent the same data. Data compression algorithms exploit this idea, attempting to find a specific representation of a given set of data that takes up as little storage space as possible. Of course, there are many trade-offs: processing difficulty (how long does it take to compress/decompress a file?), compression ratio (how much smaller the file has become after compression compared to the original output?), legal constraints (are there any patents or licensing issues that prevent us from being able to use this algorithm?), memory requirements (how much RAM do we need to process this algorithm?), and many more. There are other considerations that influence data compression algorithm design. For example, are there any special properties for the underlying data we can exploit to achieve better efficiency? With time, new data structures are discovered, new mathematical properties are found, and new ways of representing the same set of data are found, resulting in innovation in data compression algorithms. One thing that is worth keeping in mind, however, is that even within the constraints of a given data compression/decompression format, there are many ways of representing that data. Compression programs try to find the best one among many, but there are other ways to compress the data that will allow the decompressor to decode that representation back into the same original raw data.

What if we manipulated the way compression algorithms worked to produce another, equally valid string of bits that would decode into the same picture, sound, video, text, or anything else? Better yet, what if we did so in a way that allowed us to encode additional information, allowing us to embed secret information into the compressed result that we could then transmit without any changes to the source media? That is the goal of this research.

The overall flow of many compression algorithms takes three basic steps: take the source data object, define certain transformations on it, change the way it should presented through using a certain data structure; perform decisions as to how the source object should be converted into this new representation; and finally, convert it into a binary form that can be stored on a digital storage medium. For example, with pictures, instead of operating on raw pixel data, groups of pixels could be analyzed in an attempt to find larger structural blocks, information about which could be conveyed not through the raw pixel data, but with a relationship between a certain overall expectation and the difference between expectation and reality (W3C, 2025; International Organization for Standardization, 1994; Google, 2025). As could be seen in Fig. 1, on one of the compression stages in WebP image format, the compression algorithm performs various predictions on a blocks of pixels of fixed size and then chooses the one that has the best match with actual data (Google, 2025).

Though the original compression algorithm aims to produce an output that takes up the least amount of space, it does not have to. If we instead force the compression algorithm to take different, suboptimal decisions, we could encode additional data into the resulting compressed file.

ISSN 2707-1758

Fig. 2 shows a generalized view of proposed suboptimal compression decision hijacking steganography approach. For a given compression format, a set of decisions is outlined. During compression stage, a steganographic encoder performs the compression of data as normal until a decision point is reached. Then, instead of selecting an option that would lead to biggest reductions in file size, a different option may be selected, thus encoding a certain amount of secret data into the resulting file.

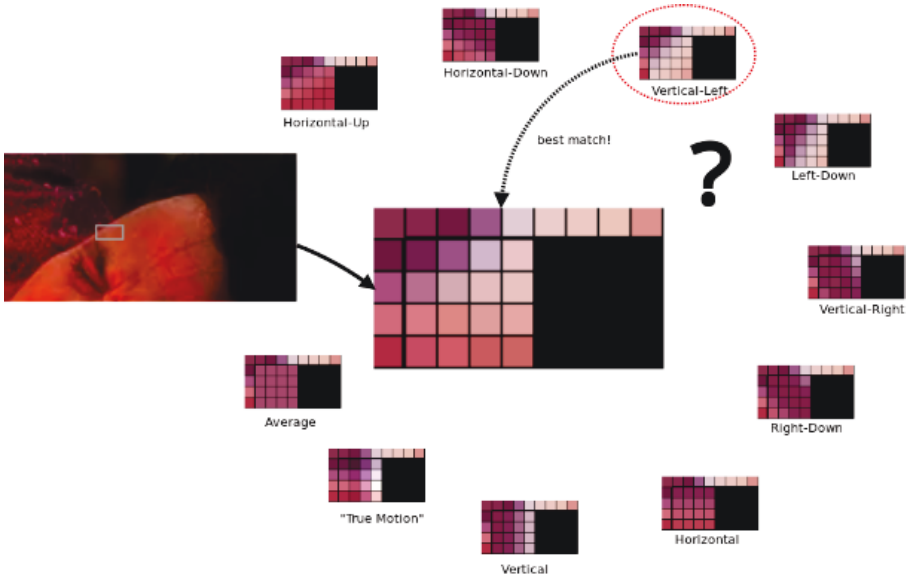


Fig. 1. Example of prediction modes in WebP

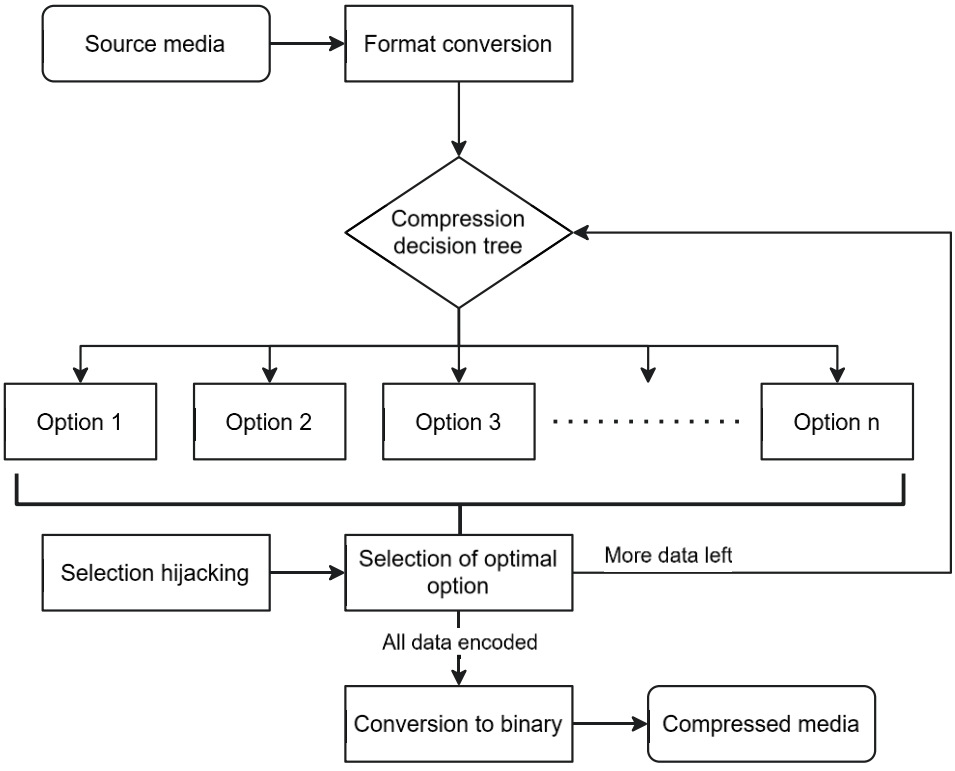


Fig. 2. Compression hijacking diagram

Selection hijacking engines (SHjEs) can be designed to adapt to specific needs. The most naive approach would be to hijack all decisions and utilize the entire decision space available to encode the cyphertext as quickly as possible. While it is a viable approach, it may result with significant distortions to the beginning of the source file, especially if compression is lossy. Additionally, the apparent dip in compression ratio between the start and the end of the file may allow for some steganographic analysis. A much better approach would be to spread the hijacked decision points around the entire length of the file. This would require either multiple compression passes to

ISSN 2707-1758

perform precisely, or a decent heuristic approach that would predict the amount of decision points and their capacity within a couple of percentage points from ground truth. Additionally, SHjE can limit its decision space to powers of two, allowing it to encode bytes directly instead of resorting to techniques like arithmetic coding (Witten, Neal, & Cleary, 1977).

Nonetheless, arithmetic coding, or other similar entropy encoding techniques, could be used in cases where maximal capacity for encoding secret data is necessary. What is especially useful in entropy encoding is the ability to use mixed-radix numeral systems, allowing us to minimize losses incurred by limiting ourselves to just binary codes. The reason it is possible to use mixed-radix numeral systems is because the compression algorithms tend to have a certain and well-defined set of options at specific decision points. In other words, the number of decisions at a specific decision point utilized to encode secret data stream defines corresponding radix in the entropy encoding solution. Fig. 3 provides an example visual illustration of steganographic encoding in action. At certain points in a compression algorithm, a decision with multiple options needs to be made. SHjE decides to use specific decision points to encode the data it needs. Because selection space is well-defined, the decoder would know the radix of this hijacked selection, allowing it to interpret it appropriately.

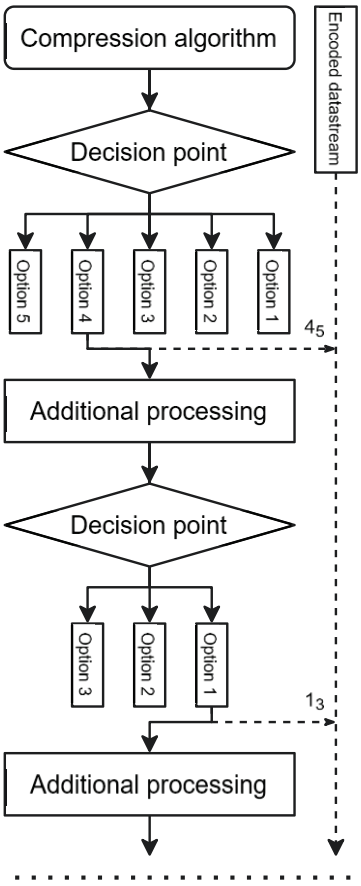


Fig. 3. Encoding of data through selection hijacking

Decoding secret data stream from a steganographic media boils down to performing the same steps in reverse: perform the decoding process; figure out the decision space that was available during the encoding step; and then derive a numerical value that corresponds to an option that was chosen during encoding.

There are two main components of this new model: SHjE itself and an embedding engine that integrates it into a given compression algorithm. While the same SHjE could be used within different data formats and compression algorithms, it requires modifying them to provide hooks and interfaces for the SHjE to operate on. While specifics of these implementations are outside of the scope of this research article (and therefore avenues for further research), they will need to rely upon specific data format and compression algorithm details to define valid decision points that could be used to encode secret data.

Mathematical model. To begin expressing this model in mathematical terms, we need to introduce some definitions. Let us define the following:

- $R = (y_1, y_2, y_3, \dots, y_n)$ – raw media object of length n, where y_i are media primitives (i.e., pixel brightness, sound intensity at a given point in time, etc.)
- e_i – i-th encoding decision with capacity c_i.
- $\mathbb{E}$ – encoding decision space for a given data compression format.
- $E(R) = (e_1, e_2, e_3, \dots, e_n)$ – encoding decision chain for a given raw media object R, $\forall i \in \overline{1,n}: e_i \in \mathbb{E}, \forall E(R) \in \mathbb{E}^*$.
- $e_i^{k_i}$ – resolved encoding decision with resolution $k_i: 1 \le k_i \le c_i$.
- $C_E(R) = \left(e_1^{k_1}, e_2^{k_2}, e_3^{k_3}, \dots, e_m^{k_m}\right)$ – a compression of R, which is a tuple of resolved encoding decisions which represent the raw source object.
- $\hat{C}_{E,S}(R) = \left(e_1^{\hat{k}_1}, e_2^{\hat{k}_2}, e_3^{\hat{k}_3}, \dots, e_m^{\hat{k}_m}\right)$ – a hijacked compression of R that carries a secret message S.
- $D\left(C_E(R)\right) = \bar{R}$ – a decompression of $C_E(R)$ which retrieves a decompressed object $\bar{R}$.

A compression $C_E(R)$ can be represented as a sequence of bits $B\left(C_E(R)\right) = (b_1, b_2, b_3, \dots, b_l)$ of length $\left|B\left(C_E(R)\right)\right| = l$. An optimal compression function $C_{E,optimal}(R)$ derives such a sequence of encoding decisions e_i and their resolutions k_i that its length is as small as possible. In other words, optimal compression satisfies the condition:

$$\min\left|B\left(C_E(R)\right)\right| = \left|B\left(C_{E,optimal}(R)\right)\right|. \quad (1)$$

It is important to note, however, that for a given lossless compression (i.e., the one for which $D\left(C_E(R)\right) = R$ always holds true for any $C_E(R)$), there is more than one way to compress a given raw media object R. In other words, there are many compressions $C(R)$ that can be decompressed back to R. We can use a subset of encoding decisions $\hat{E}(R) \subseteq E(R) \in \mathbb{E}^*$ as a carrier for our secret message S. To do this, we can take each encoding decision $\hat{e}_i \in \hat{E}(R)$ and, instead of letting the compression algorithm resolve it, give it to a SHjE, which will use it to encode a part of a secret message.

To dig deeper into a SHjE, we first need to establish an arithmetic coding system. Let us assume we have the following:

- x_l – an l-digit mixed-radix number with a finite number of digits.
- d_i – an i-th digit with a radix $r_i \in \mathbb{N}$, $d_i \in \overline{0,(r_i-1)}$.
- $\mathcal{D}(x_l) = (d_1, d_2, \dots, d_l)$ – a string of digits for an input number x_l.
- w_i^j – a weight assigned to the j-th value for i-th digit, $j \in \overline{1,r_i}$, $w_i^j \in [0,1]$.

ISSN 2707-1758

- $W_i = \{w_i^1, w_i^2, \dots, w_i^{r_i}\}$ – a set of weights assigned to each value an i-th digit could be, where:

$$\sum_{k=1}^{r_i} w_i^k = 1. \quad (1)$$

- $\mathcal{W}(x_l) = \mathcal{W}_{x_l} = \{W_1, W_2, \dots, W_l\}$ – a collection of sets of weights for an input number x_l.
- $A\big(\mathcal{D}(x_l), \mathcal{W}(x_l)\big) = [a, b)$ – an arithmetic coding of an input number x_l, $0 \le a < b < 1$.

The arithmetic coding system works by progressively dividing a given range $[a, b)$ into smaller regions $[a, b_1)$, $[b_1, b_2)$, …, $[b_{r_i-1}, b)$ (assuming $b_0 = a$, $b_{r_i} = b$), where the length of each region is proportional to the weights W_i:

$$b_n - b_{n-1} = (b - a)w_i^n. \quad (2)$$

With that in mind, an arithmetic coding of a number x_l could be defined recursively. Let $x_h = \lfloor x_l \rfloor_h$ be a number comprised of the first h digits of x_l, where $h \in \overline{1, l}$. Therefore:

$$A\big(\mathcal{D}(x_h), \mathcal{W}(x_h)\big) = [a_h, b_h),\ a_0 = 0, b_0 = 1. \quad (4)$$

$$a_h = a_{h-1} + (b_{h-1} - a_{h-1}) \sum_{i=0}^{d_h - 1} w_h^i,\ b_h = a_h + (b_{h-1} - a_{h-1}) w_h^{d_h}. \quad (5)$$

Equation (4) establishes the base case for recursion, and equation (5) defines the bounds a_h and b_h for a number x_h.

While this approach allows us to encode any number, we have no way of knowing how long that number is. To solve this problem, a stop digit can be introduced. This would increase the radix of every digit by one and require adding a new weight $w_i^{r_i+1} \in (0, 1)$. To satisfy the condition set by equation (1), the full set of weights W_i will need to be rebalanced for each digit to produce a new set of weights $\overline{W}_h$:

$$\forall i \in \overline{1, r_i}: \overline{w}_h^i = w_h^i \times \left(1 - w_h^{r_i+1}\right). \quad (6)$$

With that settled, a SHjE is a function that performs arithmetic coding through manipulating encoding resolutions $\hat{k}_i$ of encoding decisions $\hat{e}_i$ in a subset of an encoding decision chain for a given media object $\hat{E}$ provided by the embedding engine. More specifically, we use $\hat{e}_i$ as digits with corresponding radices c_i. The SHjE is also responsible for deciding upon a set of weights W_i for each digit. After that, SHjE can produce arithmetic coding T of a target value that will embed the secret data S into the hijacked compression $\hat{C}_{E,S}(R)$:

$$T = A\big(\hat{C}_{E,S}(R), \mathcal{W}_T\big) = [a_T, b_T) \in [a_S, b_S) = A\big(\mathcal{D}(S), \mathcal{W}(S)\big), \quad (7)$$

where $\mathcal{W}_T$ – weight set collection for a hijacked compression $\hat{C}_{E,S}(R)$ defined internally as one of the parameters of a SHjE, $A\big(\mathcal{D}(S), \mathcal{W}(S)\big)$ – arithmetic coding of secret data S.

As such, the hijacked compression $\hat{C}_{E,S}(R)$ is, in some sense, a byproduct of an attempt to arithmetically code our secret value using an unconventional mixed radix numbering system that consists of encoding decisions of a compression algorithm.

Note that it is not necessary to have a stop digit in the SHjE, since as soon as it encodes a value that is fully inside one of the stop digits of arithmetically coded secret data S, it can stop its processing.

Decoding the secret value S encoded in a hijacked compression $\hat{C}_{E,S}(R)$ is therefore a simple task of letting the embedding engine define the same subset of an encoding decision chain $\hat{E}$, providing the resolved encoded decisions $e_i^{k_i}$ to the SHjE, and then using them to recreate the value of T, and from that the value of S, assuming the radices of $\mathcal{D}(S)$ and the weight set collection $\mathcal{W}(S)$ is known ahead of time (either by being agreed upon by the parties using this model to exchange steganographic information, or being a part of the SHjE configuration).

To summarize, the selection hijacking engine has the following parameters:

- $\mathcal{W}_T$ – weight set collection used for creating a target arithmetic coding.
- $\mathcal{W}(S)$ – weight set collection for secret data S.
- $r_{i,S}$ – radices of digits of secret data S.
- Whereas the embedding engine has the following parameters:
- Mapping $E(R) \to \hat{E}(R)$.
- r_i – radices of digits of a hijacked encoding, $1 \le r_i \le c_i$.
- Ordering of resolutions $\hat{k}_i$ for all encoding decisions $\hat{e}_i$.

As such, the embedding engine may decide to artificially restrict decision space for the SHjE in cases where taking a particular decision may result in undesirable side effects like particularly large file size increases. Additionally, it is responsible for mapping the numeric representation of resolutions $\hat{k}_i$ into actual decisions an encoder it embeds itself into needs to take.

Comparison with existing approaches. Definitions of this research article could be used to describe most of the existing steganographic algorithms as well. While their construction varies in complexity, most of them boil down to manipulating the raw media object R, performing a transformation that turns it into an $\hat{R}$, the changes in which compared to the original are imperceptible to the human senses while providing it a certain property that is useful for covertly embedding a representation of secret data S. Least significant bits (LSB) methods operate on raw bits directly. Transform domain techniques like Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), etc. also operate on raw media object data. Same with phase coding, masking, filtering, and other transformative forms of steganography.

Selection hijacking method therefore has a unique advantage that separates it from previous models: it does not change the raw source media object but merely changes the way in which it is compressed and stored on disk. In a lossless compression algorithm, the hijacked compression leaves the raw source media completely intact, since, by the definition of a lossless compression algorithm, $D\big(C_E(R)\big) = D\left(\hat{C}_{E,S}(R)\right) = R$.

This model does have disadvantages, however. Any deviations from the optimal compression $C_{E,optimal}(R)$ will result in enlargement of the bit sequence $B\left(\hat{C}_{E,S}(R)\right)$. The compression space may not necessarily be monotonic, there may be local minima, and theoretically it is possible that a given hijacked encoding decision $e_i^{\hat{k}_i}$ may be able to decrease the overall length of the resulting bit sequence, it

ISSN 2707-1758

is expected to be an exception, not a rule. This would mean that the resulting compression $\hat{C}_{E,S}(R)$ will result in a larger file size compared to the optimal compression $C_{E,optimal}(R)$ achieved by the original compression algorithm. At the same time, it is worth noting that the resulting file contains more entropy than just the entropy of the source media object R, since it also contains a string of secret data S. Since the source media object R is left intact, it would mean that the entropy of a file with steganographically embedded data using this model is larger than the one without it. Simply put, this model results in larger file sizes.

Moreover, the storage capacity for secret data is limited by the number of decisions available to the SHjE. Depending on the compression algorithm used, it may result in orders of magnitude less storage capacity compared to the resulting file size. Since no methods based on this model have been developed yet, and no experimental data has been provided, it is difficult to estimate the exact impact on storage capacity it would allow.

Discussion and conclusions

With various data representation and storage schemes theorized, a problem arises: SHjE synchronization. To be able to correctly decode secret data stream encoded with this method, SHjEs on both ends of the process – encoding and decoding – need to be able to arrive at the same conclusions in each step of the way.

One solution is to share the implementation details and SHjE parameters beforehand. While it is a viable approach, it requires communicating these engine parameters beforehand through another, external channel. Once these engine parameters are locked in and hardcoded into the program that would perform the encoding/decoding, it would be impossible to change them without communicating through these external channels, whether for getting a new copy of the software with new hardcoded values, or for a new set of configuration options the end user would be able to input manually after receiving the necessary SHjE configuration.

A more user-friendly, though more challenging for implementors way of solving engine synchronization problems would be to embed SHjE configuration in the embedded secret data stream itself in the form of a predefined header. As long as both parties support the same base set of engine configurations, the encoding party can easily change engine parameters individually per media object, or even on the fly. Developing precise methods of engine synchronization is one of the possibilities for future research.

Secondly, since the hijacked selections will lead to lower compression performance, this will lead to a measurable and noticeable increase in storage capacity required for storing the compressed (encoded) media objects. This is an inherent property of this model. Because the resulting encoded media object theoretically retains all its original data, encoding additional secret data stream into it will necessitate an increase in file size. While it is a feature of this design, it is neither good nor bad. It is simply a design choice that can either be extremely useful to some users, or unnecessary to others.

This model would work well on lossless compression algorithms since, no matter what option is chosen by the SHjE, the implementation details and design of the compression algorithm used would eventually lead it to completely and identically representing the entire source media object in its processed (encoded) form. However, this may not be the case for lossy compression algorithms. By their nature, they discard certain information from the resulting data representation, and the decisions performed by the encoding algorithm influence which datapoints specifically get discarded. Hijacking those decisions would mean discarding different data from the resulting compressed media object. Depending on the design of the SHjE, this might lead to an encoder SHjE having different input data compared to the decoder SHjE, preventing them from being able to synchronize properly. Further research is required to adapt this model into specific methods that would address these concerns and make it possible to encode steganographic data in lossy media objects.

Furthermore, this model's performance is limited by the compression algorithm it is implemented on. To encode the secret data stream, it needs to not only perform the native compression steps but also spend computational resources into hijacking those decisions. Since those decisions are suboptimal, it is entirely likely that the compression algorithm would also need to spend more computational resources “cleaning up” after taking that suboptimal decision, further limiting its performance. Currently, it is only speculation, and further research is necessary to observe the impact of selection hijacking on compression speed and efficiency.

Additionally, without further SHjE tuning, with secret data streams that are shorter than the potential embedding capacity of a given media object, all the suboptimal compression decisions would be bunched up at the front of the file. This would allow for a potential statistical analysis attack that would allow adversaries to potentially detect this kind of manipulation. It is important to find ways of spreading out the hijacked decisions in a consistent way that is indistinguishable from randomness.

Finally, this model is well-suited for a wide variety of compression algorithms, but its implementation into actual methods requires specializing it for a particular compression algorithm, or even a particular implementation of encoders and decoders. While the theory might be broad, practical implementations need to be the focus of further research to make sure this model can be effectively leveraged in the future.

In conclusion, a new model of encoding steganographic data into media objects without changing their apparent representation was developed and proposed. This model is capable of being used on any kind of media if it has a compression algorithm that needs to perform decisions to optimize its performance. Further research is required to develop specific methods of implementing this model for specific data formats and compression algorithm implementations to bring this model from the realm of theory into the realm of practical application.

Authors' contribution: Yevhenii Ponomarenko – conceptualization, data curation, methodology, formal analysis, analysis of sources, writing – original draft; Oleksandr Laptiev – data curation, supervision, methodology, writing – review & editing.

Sources of funding. This study did not receive any grant from a funding institution in the public, commercial or non-commercial sectors.

References

Anas, T., Ridzuan, F., & Pitchay, S. A. (2025). Cover Selection in Steganography: A Systematic Literature Review. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 52(2), 107–129. https://doi.org/10.37934/araset.52.2.107129

Apau, R., Asante, M., Twum, F., Ben Hayfron-Acquah, J., & Peasah, K. O. (2024). Image steganography techniques for resisting statistical steganalysis attacks: A systematic literature review. *PLOS ONE*, 19(9), 1–47. https://doi.org/10.1371/journal.pone.0308807

ISSN 2707-1758

Barina, D. (2021). Comparison of Lossless Image Formats. *Computer Science Research Notes*, 3101, 339–342. https://doi.org/10.24132/CSRN.2021.3101.38

Galal, A. M. (2016). An analytical study on the modern history of digital photography. *International Design Journal*, 6(2), 203–215. https://www.faa-design.com/files/6/18/6-2-amr.pdf

Google. (2025, April 08). *Compression Techniques. WebP.* Google for Developers. https://developers.google.com/speed/webp/docs/compression

International Organization for Standardization. (1994). *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines* (ISO/IEC 10918-1:1994). https://www.iso.org/standard/18902.html

Öztürk, E., & Mesut, A. (2021). Performance evaluation of JPEG standards, WebP and PNG in terms of compression ratio and time for lossless encoding. In M. Yılmaz, & S. Kaya (Eds.). *Proceedings of the 2021 6th International Conference on Computer Science and Engineering (UBMK)* (n.pp.). IEEE. https://doi.org/10.1109/UBMK52708.2021.9558922

Rumsey, F., & Mccormick, T. (2013). *Sound and Recording* (6th ed.). New York: Focal Press. https://doi.org/10.4324/9780080953960

W3C. (2025, May 15). *Portable Network Graphics (PNG) Specification* (3rd ed.). W3C. https://www.w3.org/TR/png-3/

Witten, I. H., Neal, R. M., & Cleary, J. G. (1977, May). Arithmetic coding for data compression. *Communications of the ACM,* 30(6), 520–540. https://doi.org/10.1145/214762.214771

Отримано редакцією журналу / Received: 16.05.25
Прорецензовано / Revised: 16.05.25
Схвалено до друку / Accepted: 30.06.25

Євгеній ПОНОМАРЕНКО, асп.
ORCID ID: 0009-0000-2647-3381
e-mail: ponomarenkoye@fit.knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Олександр ЛАПТЄВ, д-р техн. наук, доц., ст. наук. співроб.
ORCID ID: 0000-0002-4194-402X
e-mail: olaptiev@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

МАТЕМАТИЧНА МОДЕЛЬ СТЕГАНОГРАФІЇ, ЩО ВИКОРИСТОВУЄ НЕОПТИМАЛЬНІ РІШЕННЯ В АЛГОРИТМАХ СТИСНЕННЯ ДАНИХ

Вступ. *Захист доступу до інформації в цифрову епоху вимагає розроблення цифрових інструментів для шифрування та приховування інформації від усіх, хто не повинен мати до неї доступ. Хоча шифрування чудово запобігає доступу до інформації неавторизованими особами, воно дозволяє людям знати, що за перетворенням щось приховано. Натомість, стеганографія дає змогу приховати сам факт приховування інформації. На жаль, поширені типи стеганографії покладаються на зміну вихідного сигналу, залишаючи малопомітні сліди маніпуляцій із даними, які можна відстежити та виявити. Метою цієї статті є надання моделі, яка має потенціал для збереження вихідного сигналу в повному обсязі, замість цього покладаючись на зміну способу представлення вихідного сигналу в цифрових носіях у такий спосіб, щоб мати можливість кодувати секретні дані у вихідний потік.*

Методи. *Проведено теоретичний аналіз стеганографічних підходів до алгоритмів стиснення даних. Досліджено методи збереження вихідного потоку даних.*

Результати. *Розроблено нову модель, яка використовує процеси прийняття рішень в алгоритмах стиснення даних для кодування стеганографічних даних у результуючому потоці даних. У схемах стиснення даних без втрат можливо досягти ідеального відтворення вихідного потоку даних, що унеможливлює виявлення шляхом аналізу основного сигналу, закодованого в алгоритмах стиснення даних.*

Висновки. *Потреба в інформаційній безпеці із часом зростає, а напруженість між країнами призводить до нових збройних конфліктів по всьому світу. Можливість вбудовувати та приховано надсилати дані за допомогою стеганографії може забезпечити конкурентну економічну, політичну та/або військову перевагу. Розроблену модель можна застосувати для подальшого розроблення конкретних методів стеганографічного кодування даних, стійких до аналізу та виявлення за допомогою існуючих підходів, що спираються на статистичний аналіз потоку базового сигналу.*

Ключові слова: *стеганографія, алгоритми стиснення даних, оброблення сигналів, захист інформації, дерева прийняття рішень, ентропійне кодування, арифметичне кодування.*

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

УДК 004.056.55:004.738.5
DOI: https://doi.org/10.17721/ISTS.2025.9.61-73

Іван ОПІРСЬКИЙ, д-р техн. наук, проф.
ORCID ID: 0000-0002-8461-8996
e-mail: ivan.r.opirskyi@lpnu.ua
Національний університет "Львівська політехніка", Львів, Україна

Сергій КОРЕНЬ, студ.
ORCID ID: 0009-0003-9239-136X
e-mail: serhii.koren.kb.2023@lpnu.ua
Національний університет "Львівська політехніка", Львів, Україна

Антон ГУНДЕРИЧ, студ.
ORCID ID: 0009-0005-0207-581X
e-mail: anton.hunderych.kb.2023@lpnu.ua
Національний університет "Львівська політехніка", Львів, Україна

ІМПЛЕМЕНТАЦІЯ ЗАХИСТУ ВЕБДОДАТКІВ НА NODE.JS: ОСНОВНІ ЗАГРОЗИ ТА МЕТОДИ БЕЗПЕКИ

Вступ. *Інтенсивний розвиток вебтехнологій і зростання популярності вебзастосунків, розроблених на платформі Node.js, відкривають нові можливості для цифрового бізнесу, водночас посилюючи ризики кіберзагроз. Однією з ключових проблем сучасної кібербезпеки є захист таких систем від несанкціонованого доступу, порушення цілісності даних і забезпечення їхньої стабільної роботи. У контексті зростання кількості атак на вебресурси особливої актуальності набувають засоби захисту, які можуть бути інтегровані без значного зниження продуктивності. Застосування багаторівневих механізмів, зокрема і контроль доступу, валідація введення та параметризація запитів до бази даних, формують основу сучасного підходу до безпечного розроблення.*

Методи. *Проведено систематичний аналіз сучасних методів забезпечення безпеки вебзастосунків, зокрема і в контексті Node.js. Використано методи теоретичного моделювання, порівняльного аналізу, практичного тестування систем захисту й аналізу ефективності різних стратегій. Особливу увагу приділено інтеграції механізмів захисту, таких як обмеження кількості запитів (rate limiting), оброблення параметризованих SQL-запитів, фільтрація користувацького введення та застосування принципу найменших привілеїв. Для кожного з методів оцінено рівень продуктивного навантаження, точність виявлення атак і сумісність з архітектурою Node.js.*

Результати. *Аналіз показав, що найефективнішим підходом до захисту вебзастосунків є поєднання кількох взаємодоповнювальних стратегій. Наприклад, використання параметризованих запитів суттєво знижує ризик SQL-ін'єкцій, тоді як контроль доступу до критичних ресурсів унеможливлює несанкціоновану модифікацію даних. Доведено, що комбінування rate limiting і фільтрації введення значно підвищує стійкість застосунку до атак типу brute force і script injection. Водночас такі заходи не створюють істотного навантаження на систему, що дозволяє впроваджувати їх у реальних умовах. Визначено оптимальні конфігурації захисних механізмів залежно від рівня загроз і функціональних вимог до вебзастосунку.*

Висновки. *Захист вебзастосунків, побудованих на Node.js, вимагає системного і комплексного підходу. Комбінування кількох методів захисту дозволяє досягти високого рівня безпеки без зниження ефективності роботи застосунку. Результати дослідження можуть бути використані для побудови інтегрованих систем виявлення та запобігання кіберзагрозам з урахуванням архітектурних особливостей Node.js. Крім технічних аспектів, підкреслено важливість упровадження політик безпеки, що охоплюють як технологічні, так і організаційні компоненти захисту. Системне впровадження таких підходів забезпечить стійкість застосунків навіть в умовах зростання складності кіберзагроз.*

Ключові слова: *Node.js, безпека вебзастосунків, SQL-ін'єкції, контроль доступу, обмеження запитів, фільтрація введення, багаторівневий захист, конфіденційність даних, програмна вразливість, кібербезпека.*

Вступ

У контексті глобальної цифровізації вебдодатки виконують центральну роль у впровадженні нових бізнес-моделей, автоматизації процесів і забезпеченні доступу до інформаційних сервісів для мільйонів користувачів. Платформа Node.js, яка побудована на рушії V8 і заснована на подійному циклі (event loop), забезпечує високу продуктивність і масштабованість, що робить її привабливою для створення як стартапів, так і корпоративних рішень. Однак разом із розповсюдженням вебдодатків зростає й кількість кіберінцидентів. За даними PT Security, 17 % усіх зареєстрованих кібератак спрямовано саме на вебінтерфейси, де експлуатуються вразливості в коді та конфігурації додатків. У 2024 р. фінансові втрати від кіберзлочинів досягли рекордних $16,6 млрд – це на 33 % більше, ніж 2023 р., хоча кількість інцидентів трохи зменшилася до 860 000 звернень, що свідчить про зростання складності атак і їхньої ефективності. Крім того, упродовж 2024 р. зафіксовано 40 077 нових вразливостей у програмному забезпеченні (CVE), а загрози через API-інтерфейси виросли на 681 %, що вказує на необхідність комплексних підходів до захисту як клієнтської, так і серверної частин Node.js-додатків. Отже, у світлі високої популярності Node.js і дедалі складніших кіберзагроз дослідження ефективних механізмів захисту вебдодатків на цій платформі має і теоретичне, і практичне значення для забезпечення конфіденційності, цілісності та доступності інформаційних систем.

У межах дослідження кіберзагроз, що мають критичний вплив на вебдодатки, обґрунтовано обрано одну з найпоширеніших і найнебезпечніших категорій атак – SQL-ін'єкції. Рішення зосередитися саме на цьому типі зумовлено його високою частотою появи в інцидентах інформаційної безпеки, широким спектром наслідків для цілісності даних, а також постійним домінуванням у міжнародних звітах і наукових публікаціях. Згідно з OWASP Top 10 (OWASP Foundation, 2025), ін'єкції залишаються серед трійки найнебезпечніших загроз для вебдодатків, поступаючись лише проблемам автентифікації.

Node.js є одним із найпопулярніших середовищ для розроблення сучасних вебдодатків завдяки своїй масштабованості, продуктивності й активній підтримці спільноти. Проте його асинхронна модель з

ISSN 2707-1758

обробленням запитів у межах єдиного потоку відкриває вразливості, які можуть бути критичними у разі неефективного управління введенням або за відсутності належної перевірки запитів. Наприклад, у Snyk (2025) зазначають, що SQL-ін'єкція залишається серйозною загрозою для додатків на Node.js, оскільки неконтрольоване введення без валідації або параметризації запитів дозволяє зловмисникам впроваджувати шкідливі SQL-команди, що призводить до розкриття конфіденційних даних або зміни бази даних. Саме ці аспекти підкреслюють актуальність аналізу SQL-ін'єкцій як із погляду архітектурних вразливостей, так і з позиції безпеки критичних сервісів на платформі Node.js.

Аналітичні звіти та наукові дослідження останніх років засвідчують високу вразливість вебдодатків, побудованих на Node.js, до атак типу SQL-ін'єкції. Згідно з даними Synk Open Source Security Report (2024), близько 27 % вразливостей у Node.js-додатках стосуються оброблення введення, що створює сприятливе середовище для ін'єкційних атак, особливо за відсутності параметризованих запитів (Snyk, 2024). У звіті IBM X-Force Threat Intelligence Index за 2024 р. зазначено, що SQL-ін'єкції стали причиною 41 % інцидентів, пов'язаних із порушенням конфіденційності у вебдодатках (IBM, n. d.).

Отже, вивчення SQL-ін'єкцій у контексті Node.js дозволяє оцінити найкритичніші фактори ризику для вебдодатків і надати рекомендації щодо побудови ефективної моделі безпеки. Це дослідження має практичну та методологічну цінність, оскільки охоплює аналіз поточних тенденцій, вразливостей і захисних підходів у найбільш уживаних середовищах веброзроблень.

Метою цього дослідження є аналіз основних загроз безпеці вебдодатків, розроблених на платформі Node.js, і визначення ефективних методів їхнього захисту. З огляду на поширеність атак, таких як SQL-ін'єкції, XSS, CSRF, атаки "людина посередині" (Man-in-the-Middle), експлуатація веселкових таблиць, віддалене виконання коду, необхідно розробити комплексні заходи для їхньої нейтралізації.

Для досягнення цієї мети передбачено розв'язання таких завдань:

- Провести аналіз існуючих загроз безпеці вебдодатків на основі Node.js.
- Дослідити методи атаки, що найчастіше використовуються для компрометації даних.
- Визначити основні принципи та механізми захисту вебдодатків.
- Розглянути способи підвищення рівня захищеності комунікацій.
- Запропонувати рекомендації щодо посилення безпеки вебдодатків з урахуванням сучасних стандартів кібербезпеки.

***Огляд літератури*.** Протягом останніх років спостерігається значне зростання кіберзагроз, що впливають на безпеку вебдодатків. Зокрема, у 2023 р. кількість атак на прикладному рівні HTTP зросла на 93 % порівняно з попереднім роком. Це свідчить про підвищену активність зловмисників, які використовують нові методи для обходу традиційних засобів захисту.

Дослідження Edgescan показало, що 19,47 % виявлених вразливостей у 2023 р. були класифіковані як високого або критичного рівня (Edgescan, 2024). Це підкреслює необхідність упровадження ефективних стратегій управління вразливостями та постійного моніторингу безпеки вебдодатків. Варто зазначити, що дослідження Cyber Security Threats and Vulnerabilities: A Systematic Mapping Study (2022) провело систематичне картування наукових праць, спрямованих на вивчення загроз і вразливостей у кіберпросторі (Awan, & Khan, 2020). Автори класифікували існуючі дослідження за тематиками, методами та напрямами, що дозволило виявити прогалини у знаннях і визначити пріоритетні області для майбутніх досліджень.

Звернемо увагу на дослідження A Survey of Security Vulnerabilities and Protective Strategies (2021), яке зосереджується на аналізі поширених вразливостей у програмному забезпеченні та мережах, а також на ефективності різних стратегій захисту (Zhang et al., 2023). Автори підкреслюють важливість проактивного підходу до безпеки, включаючи регулярне оновлення систем, навчання персоналу та впровадження багаторівневих механізмів захисту.

У дослідженні Ishaq, & Fareed (2023), представленому як "Mitigation Techniques for Cyber Attacks: A Systematic Mapping Study", проведено систематичне картування методів протидії кіберзагрозам з акцентом на пріоритетні уразливості, такі як зловмисне ПЗ, фішинг та ін'єкційні атаки (Ishaq, & Fareed, 2023). Авторами визначено основні категорії методів захисту – від пасивних (блокування та фільтрація) до активних. Вони також зазначають, що значна частина досліджень зосереджується на загальних механізмах запобігання, але потребує професійної адаптації до специфічних середовищ, як-от вебдодатків на Node.js. Це підкреслює нагальність упровадження спеціальних стратегій, таких як багаторівнева фільтрація запитів та оптимізовані схеми оброблення введення.

Крім того, аналітична публікація Intigriti (2024) зосереджується на викликах, пов'язаних із новими технологіями – штучним інтелектом, Інтернетом речей і блокчейном, та їхньому впливу на кібербезпеку (Intigriti, 2024). Автори наголошують, що швидке впровадження цих технологій значно розширює площину атаки, а також вимагає адаптивних стратегій, таких як машинне навчання для виявлення аномалій і динамічні методи фільтрації трафіка. У контексті Node.js-додатків це дозволяє поєднувати контекстний аналіз поведінки запитів з оптимізованими обмеженнями входу, що значно підвищує стійкість застосунків до сучасних кіберзагроз.

У контексті фінансових технологій (FinTech) дослідження Cybersecurity Threats in FinTech: A Systematic Review (2023) виявило 11 основних кіберзагроз, серед яких: фішинг, атаки на ланцюги постачання та вразливості в API (Jabeen, Li, & Kim, 2024). Автори пропонують 9 стратегій захисту, включаючи багатофакторну автентифікацію, шифрування та моніторинг аномалій, що є особливо актуальними для вебдодатків на Node.js, які часто використовують у фінансових сервісах.

Варто відмітити дослідження Modern Hardware Security: A Review of Attacks and Countermeasures (2025), яке аналізує вразливості на апаратному рівні, такі як атаки через кеш-пам'ять та електромагнітні випромінювання (Zhang, & Lee, 2025). Автори підкреслюють необхідність врахування апаратних аспектів безпеки у розробленні вебдодатків, особливо в умовах зростання використання хмарних технологій та IoT.

Незважаючи на прогрес у вивченні способів захисту й аналізу їх, потрібно ще виконати багато роботи для

ISSN 2707-1758

повноцінного вивчення цієї теми. Подальші дослідження повинні зосередитися на інтеграції існуючих стратегій захисту, розробленні нових методів виявлення та реагування на загрози, а також на вдосконаленні навчальних програм для підвищення обізнаності користувачів щодо кібербезпеки.

Методи

В роботі використано методи аналізу, які включають систематичне картування наукових публікацій із кібербезпеки, теоретичне моделювання механізмів SQL-ін'єкцій, порівняльний аналіз ефективності захисних методів, емпіричне тестування систем безпеки й оцінювання ризиків впливу вразливостей, що дозволило досягти комплексного розуміння принципів захисту вебдодатків на платформі Node.js від основних кіберзагроз.

Результати

SQL-ін'єкція (SQL Injection) – це один із найпоширеніших типів атак на вебдодатки, що виникає, коли зловмисник має змогу вписати шкідливий SQL-код у запит до бази даних через незахищене або неналежно захищене поле для введення користувача. У результаті цього атака дозволяє несанкціонований доступ до БД.

У вебдодатках, розроблених на платформі Node.js, SQL-ін'єкції можуть виникати за використання SQL-баз даних, особливо у випадках, коли розробники формують запити вручну, без застосування ORM.

Принцип дії SQL-ін'єкції полягає в тому, що зловмисник вставляє фрагмент SQL-коду в текстове поле або параметр URL, який без належної валідації потрапляє у фінальний SQL-запит. Це дозволяє змінити логіку виконання запиту.

Процес (рис. 1) зазвичай починають з того, що користувач заповнює поле для введення. Якщо бекенддодаток не використовує механізмів очищення або підготовлених запитів (prepared statements). Зловмисник може скористатися цим і ввести запит, щоб змінити логіку роботи SQL. У результаті замість пошуку конкретного запиту, запит поверне всі записи, бо певна умова може бути завжди істинною. Це дозволяє обійти автентифікацію або витягти конфіденційну інформацію з бази даних.

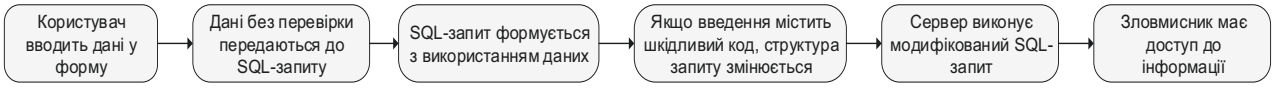


Рис. 1. Процес виконання SQL-ін'єкції

SQL-ін'єкція виникає тоді, коли введення користувача без перевірки інтегрується в SQL-запит. Наприклад, розглянемо такий запит:

*SELECT * FROM users WHERE username = = 'введене_користувачем';*

може стати вразливим, якщо користувач введе:

' OR '1'='1.

Результат:

*SELECT * FROM users WHERE username = '' OR '1'='1';*

що повертає всі рядки таблиці.

Враховуючи фундаментальний механізм формування вразливого запиту та здатність зловмисника через несанкціоновану інтерполяцію впливати на виконання SQL-інструкцій, будь-який вхідний канал, через який дані надходять до СУБД, може стати об'єктом атаки. Систематична класифікація цих каналів дозволяє побудувати ефективні стратегії захисту, зокрема й застосування підготовлених виразів (prepared statements) і суворого розділення коду і даних. У подальшому буде здійснено детальний аналіз основних джерел користувацького введення, що слугують точками входу для SQL-ін'єкцій, включно з вебформами, URL-параметрами, HTTP-заголовками та запитами через API.

Дослідження джерел виникнення атаки. SQL-ін'єкція може виникати будь-де, де дані користувача потрапляють у SQL-запит без попередньої валідації чи чіткого розділення даних і коду. У контексті вебдодатків на Node.js особливу увагу слід приділити таким каналам надходження даних:

- Вебформи. Поля форм є одним із найпоширеніших джерел SQL-ін'єкцій, оскільки вони безпосередньо приймають довільний текст від користувача. Якщо, наприклад, валідація обмежується лише перевіркою на порожнечу або довжину, а введене значення безпосередньо конкатенується із SQL-рядком, зловмисник може впровадити довільні SQL-оператори. У численних дослідженнях зазначено, що понад 70 % виявлених SQL-вразливостей належать саме до полів форм (поля логіна, реєстрації, пошуку, коментарів тощо) (OWASP Cheat Sheet Series, 2025).
- URL-параметри (query parameters). Дані з рядка запиту (наприклад, ?id=123) часто використовують для формування умов WHERE в SQL. Якщо значення параметра не проходить через функцію безпечного розмежування (parameter binding), то замість числового ідентифікатора може бути передано частину SQL-коду. Дослідження OWASP рекомендують автоматизоване тестування всіх параметрів URL на наявність SQL-ін'єкцій, оскільки цей канал часто залишається поза увагою розробників (OWASP Testing for SQL injection, 2025).
- HTTP-заголовки. Більшість розробників фільтрує лише ті дані, що надходять у тілі запиту або в параметрах URL, ігноруючи заголовки (User-Agent, Referer, Cookie тощо). Проте ці поля легко модифікувати на стороні клієнта, і якщо вони логуються або використовуються для формування динамічних запитів, то можуть стати вектором атаки. Наприклад, під час логування User-Agent у БД без екранування літералів, зловмисник здатен увести SQL-код безпосередньо в заголовок запиту.
- WebSocket-повідомлення й API-запити (JSON, XML, SOAP). Сучасні Node.js-додатки активно використовують WebSocket або REST/GraphQL API для обміну даними у форматах JSON чи XML. Хоча ці протоколи відрізняються від традиційних форм та URL-параметрів, вони так само приймають дані від клієнта. Якщо такі повідомлення обробляють без строгих механізмів розмежування SQL-параметрів, вони становлять потенційну точку входу для ін'єкції. OWASP радить охоплювати автоматичними сканерами не лише GET/POST-параметри, а й усі JSON/XML-запити, щоб виявити приховані вектори атаки (Halfond, Viegas, & Orso, 2006).

ISSN 2707-1758

Для глибшого розуміння механізмів SQL-ін'єкцій доцільно звернутися до їхньої класифікації, що базується на ознаках взаємодії між цільовим застосунком і зловмисником.

Класифікація типів SQL-ін'єкцій. Основними ознаками, які використовують для класифікації, є:

- канал передачі даних між клієнтом і сервером (той самий канал чи інший);
- наявність або відсутність прямих відповідей від сервера;
- можливість отримання зворотного зв'язку про результат виконання запиту;
- використання побічних каналів комунікації, таких як DNS або HTTP. Відповідно до цих критеріїв класифікації, SQL-ін'єкції умовно поділяють на такі типи:
  - In-band SQL-ін'єкція. Це найпростіший і найрозповсюдженіший тип, за якого результат ін'єкції повертається тим самим каналом, через який була передана атака. Такий тип атаки легше реалізується і забезпечує прямий зворотний зв'язок. Прикладами є: Classic SQLi (отримання всіх записів із таблиці); Error-based SQLi (отримання інформації з повідомлень про помилки SQL).
  - Blind SQL-ін'єкція. Виникає тоді, коли сервер не повертає явних помилок або даних, проте зловмисник може робити припущення на основі відмінностей у поведінці вебзастосунку (напр., тривалості відповіді, переадресації тощо). Вона поділяється на:
    - Boolean-based Blind SQLi (умова істинна / неістинна);
    - Time-based Blind SQLi (запити з примусовою затримкою виконання, якщо умова істинна).
    - Out-of-band SQL-ін'єкція. Застосовують у разі, коли ні in-band, ні blind ін'єкції неефективні. Зловмисник створює умови, за яких СУБД надсилає результати атаки на зовнішній сервер, контрольований атакувальником (напр., через DNS-запити або HTTP-виклики). Такий тип потребує специфічних умов – зокрема і доступу СУБД до зовнішніх мереж.

Ця класифікація є важливою з практичного погляду, оскільки кожен тип SQL-ін'єкції вимагає специфічних методів виявлення та захисту. Розуміння типу ін'єкції дозволяє розробникам і фахівцям із безпеки обрати найефективніші інструменти аналізу та протидії загрозі.

SQL-ін'єкції становлять серйозну загрозу не лише технічному рівню інформаційної безпеки, а й функціональній цілісності бізнес-процесів. У технічному вимірі зловмисники можуть:

- отримати несанкціонований доступ до конфіденційних даних (паролів, банківських реквізитів, персональних даних);
- модифікувати або видаляти дані, порушуючи логіку та достовірність системи;
- отримати права адміністратора, що дає змогу змінювати налаштування або встановлювати шкідливе програмне забезпечення;
- організувати атаку на інші частини інфраструктури через скомпрометований застосунок;
- ініціювати витік сесій, що призводить до викрадення облікових записів або доступу до фінансових ресурсів користувачів.

Крім того, SQL-ін'єкція може спричинити надмірне навантаження на сервер, що знижує його продуктивність, а в окремих випадках – призводить до повної недоступності системи.

З організаційного погляду, наслідки можуть включати:

- втрату довіри користувачів;
- репутаційні втрати компанії;
- фінансові збитки внаслідок штрафів, судових позовів або компенсацій;
- невиконання вимог стандартів безпеки, таких як PCI DSS або GDPR.

Отже, з урахуванням класифікації типів SQL-ін'єкцій та оцінювання їхньої шкоди – як технічної, так і організаційної – необхідно перейти до аналізу методів захисту.

Аналіз методів захисту від SQL injection. У зв'язку із зазначеними вище загрозами та їхніми потенційно руйнівними наслідками, особливої актуальності набуває впровадження ефективних і водночас практично реалізованих механізмів захисту вебзастосунків. У подальшому аналізі увага зосереджуватиметься на підходах, ефективність яких підтверджено як у теоретичній літературі, так і в умовах реального виробничого середовища. Зазначені методи отримали широке визнання серед розробників і фахівців з інформаційної безпеки завдяки їхній практичності, адаптивності та відповідності галузевим стандартам.

Методи, які розглядатимемо в подальшому, обрано з огляду на кілька ключових чинників:

- простота інтеграції та мінімальні втручання в існуючу кодову базу: кожен із підходів може бути впроваджений поступово, що є критично важливим для команд, які працюють за методологіями безперервної інтеграції та доставки (CI/CD), і дає змогу оперативно усувати виявлені вразливості;
- широка підтримка в екосистемі Node.js: розглянуті засоби підтримуються більшістю сучасних фреймворків і бібліотек, що знижує поріг входження, витрати на навчання персоналу та спрощує їхнє впровадження в типові розробницькі процеси (напр., функції escape у драйверах MySQL, бібліотека validator.js, можливості ORM тощо);
- забезпечення захисту на кількох рівнях взаємодії із вхідними даними: застосування поєднання методів – від фільтрації вхідної інформації (Escaping, Input Validation, White-listing) до структурної ізоляції коду й даних (Prepared Statements) та обмеження рівня доступу (Least Privilege) – дає змогу досягти багаторівневої безпеки;
- відповідність найкращим практикам OWASP і міжнародним стандартам: кожен із розглянутих підходів рекомендований авторитетними безпековими організаціями, зокрема й у межах OWASP Top 10, що свідчить про їхню надійність, обґрунтованість і доведену ефективність у запобіганні SQL-ін'єкціям;
- масштабованість і гнучкість упровадження: комбінація обраних методів дозволяє адаптувати стратегії безпеки до різних масштабів і архітектур програмного забезпечення, підтримуючи оптимальний баланс між продуктивністю та рівнем захисту.

Беручи до уваги вказані аргументи, доцільно розпочати аналіз з одного з найбільш інтуїтивно зрозумілих і простих у реалізації методів – Escaping.

Escaping. Escaping є одним із базових методів запобігання SQL-ін'єкціям і передбачає попереднє перетворення вхідних користувацьких даних таким способом, щоб спеціальні символи або оператори SQL втрачали своє функціональне значення та трактувалися як звичайна частина рядка, а не як елементи

ISSN 2707-1758

SQL-синтаксису. Це унеможливлює несанкціоновану зміну структури SQL-запиту з боку користувача.

До вставки користувацьких значень у SQL-запит, спеціальні функції або бібліотеки здійснюють перевірку вхідних параметрів на наявність символів, що мають спеціальне значення в SQL (напр., лапки, крапки з комою, коментарі), і замінюють їх безпечними еквівалентами. У контексті PostgreSQL (рис. 2) одним із ключових аспектів є належне екранування одинарних лапок ('), оскільки саме вони найчастіше використовуються для "розриву" синтаксичної структури запиту.

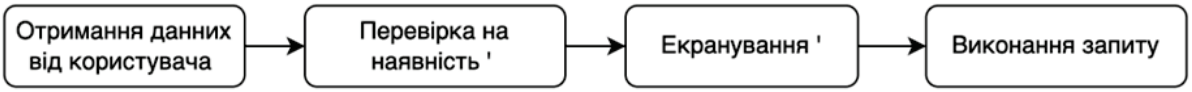


Рис. 2. Механізм виконання у контексті PostgreSQL

Наприклад:

Початковий символ: '

Після екранування: ''

У практичному прикладі, якщо SQL-запит має вигляд:

*SELECT * FROM user WHERE name = ?*

Підставити замість ? - *test'; DROP TABLE user --* SQL отримає

*SELECT * FROM user WHERE name =*
= 'test'; DROP TABLE user--'

У такому вигляді зловмисний код буде виконано, що призведе до втрати даних. Натомість, застосування escaping трансформує ' у '', і результатом стане:

*SELECT * FROM user WHERE name =*
= 'test''; DROP TABLE user--'

У цьому випадку увесь шкідливий фрагмент буде інтерпретовано як частина рядка, а не як SQL-інструкція.

Переваги використання Escaping:

- простота впровадження: реалізація не потребує складних змін у коді або архітектурі програми;
- гнучкість: метод може бути використаний у різних контекстах і з різними типами баз даних.

Недоліки використання escaping:

- незахищеність інших типів параметрів: Escaping ефективний переважно для рядкових значень, тоді як інші типи даних (напр., числа чи булеві значення) потребують додаткового оброблення або перевірки;
- реалізація на стороні прикладного коду: потребує уважності з боку розробників, що створює ризик людського фактора – наприклад, забування застосувати Escaping в окремих випадках або некоректне використання функцій.

У зв'язку із цим постає потреба у додаткових механізмах, які дозволяють здійснювати первинну фільтрацію даних ще до їхньої передачі до SQL-двигуна. Одним із таких універсальних і широко застосовуваних підходів є валідація вхідних даних (Input Validation), що забезпечує формальне підтвердження відповідності параметрів установленим критеріям безпеки та типізації.

Реалізацію функції, яка імплементує алгоритм Escaping (рис. 3).

```
// Функція Escaping: екранує одинарні лапки (для
PostgreSQL)
function escapeSQL(str) {
  if (typeof str !== 'string') {
    return str;
  }
  return str.replace(/'/g, "''");
}

const userInput = "admin' OR 1 = 1 --";
const escapedInput = escapeSQL(userInput);
console.log("Escaped Input:", escapedInput);
// Результат: 'admin'' OR 1=1 -'
```

Рис. 3. Лістинг функції Escaping

Використання input Validation. Input Validation – це метод захисту, що полягає у перевірці вхідних даних перед їх подальшим обробленням системою. Основною метою цього підходу є запобігання введенню неочікуваних, потенційно шкідливих або некоректних значень, які можуть призвести до порушення бізнес-логіки або створити вектори атак.

Процес виконання показано на рис. 4. Input Validation реалізують завдяки застосуванню набору правил до кожного параметра, який надходить від користувача. Ці правила можуть включати:

- Формат: перевірка структури введеного значення (напр., email повинен відповідати шаблону name@example.com).
- Тип: визначення того, чи відповідає тип даних очікуваному (рядок, ціле число, дата тощо).
- Довжина: встановлення мінімальної та максимальної довжини для рядків.
- Діапазон: обмеження числових значень певними межами (напр., вік від 0 до 130).
- Контекст: відповідність даних конкретному призначенню або умовам бізнес-логіки (напр., правильна структура доменного імені в email).

У типовій ситуації, наприклад під час оброблення форми автентифікації, система отримує значення полів email і password. Далі кожен параметр проходить через набір валідаційних правил: регулярні вирази перевіряють

ISSN 2707-1758

формат email; тип і довжину оцінюють окремо, додаткові обмеження накладають залежно від контексту.

Для прикладу, перевірка email може виглядати так:

const emailRegex =

= /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

Цей регулярний вираз гарантує, що введене значення матиме валідну структуру адреси, унеможливлюючи ін'єкції SQL шляхом обмеження допустимих символів. Подібно, для поля password можуть бути визначені обмеження щодо довжини (напр., 8–30 символів) і символів, заборонених до використання (лапки, крапка з комою, подвійне тире тощо).

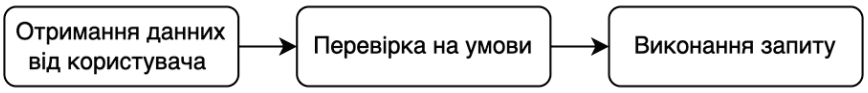


Рис. 4. Процес виконання розібраного алгоритму

Переваги Input Validation:

- Універсальність – дозволяє захиститися не лише від SQL-ін'єкцій, а й від багатьох інших типів атак, таких як XSS, Command Injection, Path Traversal.
- Забезпечення цілісності даних – суттєво знижує ймовірність потрапляння аномальних або "сміттєвих" даних у систему.
- Простота впровадження – базові механізми перевірки легко інтегруються в логіку вебзастосунку.

Недоліки Input Validation:

- Обмежена ефективність як єдиний засіб захисту – метод не змінює структуру SQL-запиту, тому не є достатнім для повної нейтралізації SQL-ін'єкцій.
- Залежність від строгості правил – ефективність залежить від якості, вичерпності та контекстної релевантності встановлених обмежень; недоопрацьовані або занадто загальні правила залишають можливість для обхідних сценаріїв.

Реалізацію функції, яка імплементує алгоритми Input Validation, показано на рис. 5.

Для підвищення ефективності перевірки вхідних даних і зменшення ризику обходу захисту, метод Input Validation доцільно поєднувати з іншими підходами. Одним із таких є White-listing – метод, що полягає у суворому визначенні допустимих значень або шаблонів, які система може прийняти. На відміну від загальної перевірки на правильність формату, цей підхід базується на принципі "дозволено тільки те, що явно дозволено", що забезпечує вищий рівень безпеки під час оброблення критичних даних.

```
// Функція Input Validation для email:
// Перевіряє, чи є рядок валідним email за допомогою
регулярного виразу,
// а також чи не перевищує максимально допустиму довжину
(наприклад, 254 символи)
function validateEmail(email) {
  if (typeof email !== 'string') {
    return false;
  }
  // Основний регулярний вираз для email
  const emailRegex =
/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
  const maxLength = 254;
  return emailRegex.test(email) && email.length <=
maxLength;
}

const emailInput = "example@gmail.c";
if (validateEmail(emailInput)) {
  console.log("Email валідний");
} else {
  console.log("Email невалідний");
}
// Результат: "Email невалідний"
```

Рис. 5. Лістинг Input Validation

Використання white-listing. White-listing (перевірка за білим списком) – це метод контролю вхідних даних, який дозволяє брати до оброблення виключно ті значення, що явно дозволені й попередньо визначені розробником або системним аналітиком. Цей підхід є суворішою формою валідації (input validation), оскільки не допускає жодних відхилень від переліку допустимих значень, незалежно від контексту або формальної коректності введених даних.

На відміну від звичайної перевірки даних, яка часто дозволяє широке коло допустимих значень за формою (напр., відповідність регулярному виразу), White-listing (рис. 6) повністю виключає будь-яке значення, яке не входить до заздалегідь установленого набору. Такий підхід особливо ефективний у випадках, коли система очікує отримати конкретні, обмежені за кількістю варіанти – як-от ролі користувача, типи операцій, коди статусів, параметри сортування тощо.

Якщо система, наприклад, дозволяє лише певні ролі користувачів, як-от "admin", "user", "moderator", то будь-який інший ввід – незалежно від синтаксичної правильності або схожості – автоматично відкидається. Це не лише підвищує безпеку, але й забезпечує цілісність даних і логіку системи. Подібний підхід також унеможливлює впровадження ін'єкційного коду в критичні параметри, навіть у випадках, коли попередня фільтрація або escaping були помилково пропущені.

ISSN 2707-1758

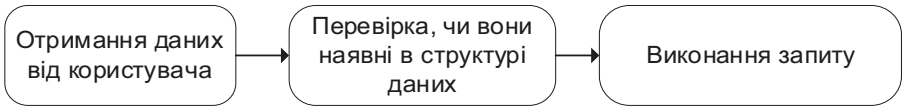


Рис. 6. Процес виконання White-listing

Переваги White-listing:

- Суворий контроль введених даних: дозволяє повністю виключити можливість оброблення непередбачуваних або шкідливих значень, що значно знижує ризик SQL-ін'єкцій, XSS та інших атак, які експлуатують довільний ввід користувача.
- Висока передбачуваність поведінки системи: забезпечує стабільність і логічну узгодженість оброблення даних, що особливо важливо для критично важливих або безпекових застосунків.
- Простота логіки перевірки: у випадках фіксованого набору допустимих значень перевірка є прямолінійною, її легко формалізувати у специфікаціях і перевірити під час аудиту.

Недоліки White-listing:

Обмежена гнучкість: метод ефективний лише для тих вхідних даних, які можна заздалегідь перерахувати. У випадках із довільними рядками, динамічними параметрами або складною бізнес-логікою (напр., пошукові запити, коментарі, адреси електронної пошти) застосування білого списку стає непридатним або вимагає додаткового супроводу.

Підвищені вимоги до проєктування: реалізація White-listing потребує детального аналізу всіх можливих вхідних значень та уважного проєктування переліків допустимого вводу. У разі помилок на цьому етапі може бути втрачена функціональність або створені перешкоди для легітимних користувачів.

White-listing демонструє високу ефективність як компонент комплексної стратегії захисту вебдодатків. Однак його слід використовувати не ізольовано, а в комбінації з іншими методами, такими як escaping, prepared statements і валідація формату, що дозволяє забезпечити багаторівневий захист від ін'єкцій та інших атак, пов'язаних з обробленням користувацьких даних (рис. 7).

```
// Функція White-listing:
// Перевіряє, чи входить задане значення до списку
дозволених значень.
function validateWithWhitelist(value, allowedValues) {
  return allowedValues.includes(value);
}

const allowedRoles = ["admin", "user", "moderator"];
const roleInput = "admin";
if (validateWithWhitelist(roleInput, allowedRoles)) {
  console.log("Роль з білого списку");
} else {
  console.log("Роль не належить до дозволених");
}
// Результат: "Роль з білого списку"
```

Рис. 7. Лістинг White-listing

Використання Prepared Statements. Prepared Statements (підготовлені запити) є одним із найбільш ефективних і надійних методів захисту від SQL Injection у вебдодатках. Цей підхід полягає у попередньому компілюванні SQL-запиту в системі управління базами даних до фактичного виконання з передачею параметрів у відокремленому вигляді. Відокремлення логіки запиту від даних дозволяє уникнути ін'єкцій, оскільки навіть потенційно шкідливі значення інтерпретуються СУБД як звичайні дані, а не як частина SQL-інструкцій.

На відміну від динамічних SQL-запитів, у яких структура формується завдяки безпосередньому включенню даних у текст інструкції, підготовлені запити передбачають чітке розмежування між кодом запиту та його параметрами. Спочатку формується шаблон SQL-інструкції з так званими *плейсхолдерами* (англ. placeholders або bind variables), які позначають місця для майбутніх значень. Цей шаблон надсилається до СУБД, де компілюється один раз і може бути використаний багаторазово з різними наборами параметрів.

Етапи виконання:

1) Формування шаблона (рис. 8). Застосунок формує SQL-запит із плейсхолдерами замість фактичних значень.

Наприклад:

*SELECT * FROM users WHERE email =*
= ? AND password = ?

Тут знаки питання *?* є змінними місцями, що будуть заміщені фактичними значеннями під час виконання.

2) Компіляція запиту в СУБД (рис. 8). Система управління базами даних отримує шаблон і здійснює його синтаксичний аналіз, оптимізацію та компіляцію. На цьому етапі СУБД визначає план виконання запиту, що може бути збережений у пам'яті для подальшого повторного використання.

3) Передача параметрів та виконання (рис. 9). Коли користувач надає фактичні значення (напр., email і пароль), вони передаються до СУБД окремо від шаблона. СУБД підставляє ці значення у відповідні місця без зміни структури запиту, після чого здійснює його виконання.

ISSN 2707-1758

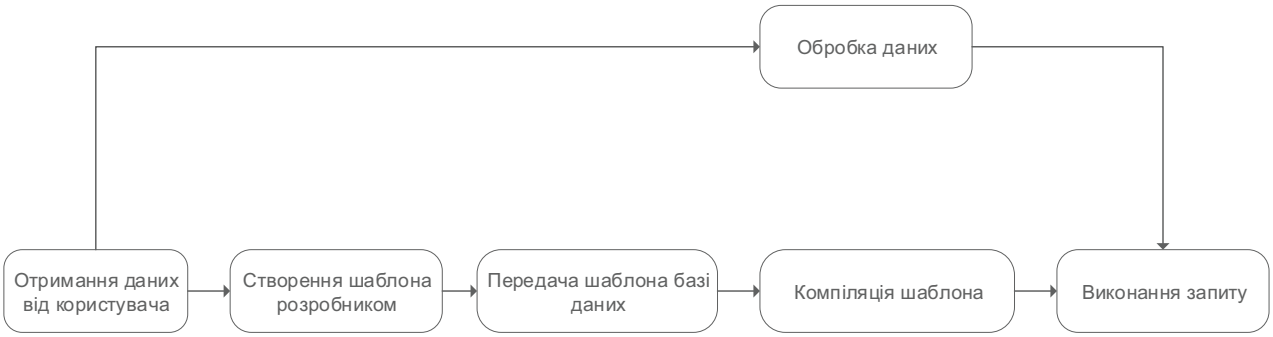


Рис. 8. Процес формування і компілювання шаблона з передачею у нього даних

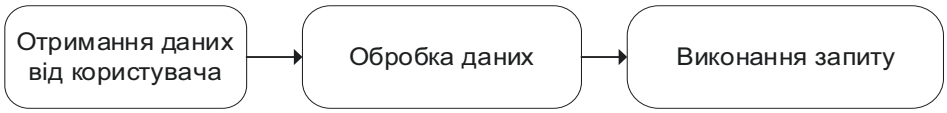


Рис. 9. Налаштування виконання Prepared Statements шаблона

Такий підхід дозволяє багаторазово використовувати той самий шаблон запиту з різними наборами значень, що суттєво підвищує продуктивність системи.

Переваги Prepared Statements:

Безпека даних. Основною перевагою підготовлених запитів є їхня здатність повністю ізолювати логіку SQL-запиту від вхідних даних. Завдяки цьому навіть, якщо користувач намагається передати ін'єкційний код, наприклад *' OR '1'='1*, СУБД сприйме його як звичайний рядок, а не як частину логіки запиту. Отже, можливість SQL-ін'єкції повністю усувається на структурному рівні.

Ефективність виконання.Оскільки компіляція запиту відбувається лише один раз, подальші виклики з різними параметрами не потребують повторного аналізу, що скорочує витрати ресурсів СУБД і зменшує час відповіді.

Масштабованість і гнучкість. Prepared Statements підтримуються більшістю сучасних СУБД (MySQL, PostgreSQL, SQLite, Oracle тощо) і можуть застосовуватися до будь-яких типів SQL-операцій: *SELECT*, *INSERT*, *UPDATE*, *DELETE*. Вони також дозволяють виконувати запити в циклі або пакетному режимі, що зручно для оброблення великих обсягів даних.

Обмеження Prepared Statements:

Попри свої переваги, підготовлені запити не можуть бути єдиним інструментом захисту. Їх використання обмежується лише на рівні бази даних. Для комплексного захисту вебдодатка вони мають поєднуватися з іншими методами, зокрема:

- валідацією вхідних даних (Input Validation);
- фільтрацією за білим списком (White-listing);
- escaping.

Крім того, не всі частини SQL-запиту підтримують параметризацію. Наприклад, назви таблиць або колонок не можна підставити у вигляді плейсхолдерів. У таких випадках слід бути особливо обережними та перевіряти введені значення додатковими методами.

Хоча описані технічні методи ефективно запобігають безпосередньому впровадженню шкідливого коду, сам по собі захист від SQL Injection не вичерпується фільтрацією чи параметризацією запитів. Важливо також враховувати архітектурні принципи безпеки, серед яких один із ключових – принцип найменших привілеїв (Principle of Least Privilege).

Least Privilege. Принцип найменших привілеїв є фундаментальним концептом у галузі інформаційної безпеки, що передбачає надання кожному суб'єкту системи – користувачу, сервісу, процесу або компоненту – лише тих прав доступу, які є строго необхідними для виконання його функцій. У контексті безпеки вебдодатків, реалізація цього принципу допомагає істотно обмежити потенційні наслідки як випадкових помилок користувачів, так і цілеспрямованих атак, зокрема й SQL Injection.

Основною метою впровадження принципу найменших привілеїв є зменшення поверхні атаки системи. Якщо певний обліковий запис або процес буде скомпрометований внаслідок атаки, наприклад, шляхом SQL-ін'єкції, то зловмисник отримає мінімальні повноваження, обмежені лише тим, що було доступно цій одиниці системи. Це означає, що навіть у разі успішного втручання, шкода буде локалізована і не зможе поширитися на критичні частини інформаційної інфраструктури, зокрема і на зміну конфігурацій бази даних, видалення таблиць чи ескалацію привілеїв (рис. 10).

Реалізація принципу найменших привілеїв у вебдодатках охоплює кілька рівнів:

На рівні бази даних: кожен додаток або модуль має використовувати обліковий запис із правами, обмеженими до необхідного мінімуму. Наприклад, публічний інтерфейс не повинен мати права на *DROP*, *ALTER*, або *GRANT*.

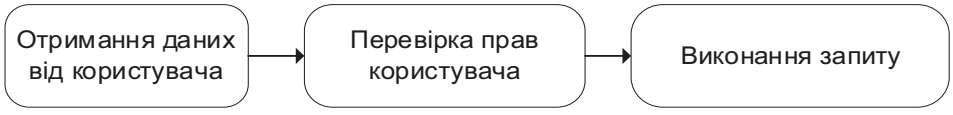


Рис. 10. Принцип роботи Prepared Statements

ISSN 2707-1758

На рівні операційної системи: вебсервер та інші служби не повинні виконуватись від імені адміністративного користувача (root), а мають запускатися з обмеженим контекстом без зайвих прав доступу до файлової системи або мережних ресурсів.

На рівні бізнес-логіки додатка: доступ користувачів до функціоналу додатка має визначатися на основі ролей (Role-Based Access Control), де кожна роль має обмежений набір дій, що дозволені їй відповідно до службових обов'язків.

Переваги Least Privilege:

- Зменшення ймовірності ескалації атаки. У разі, якщо атакувальнику вдасться скомпрометувати систему, його можливості будуть суттєво обмежені. Наприклад, при виконанні SQL-ін'єкції через форму входу зловмисник зможе отримати лише доступ до перегляду даних, але не зможе видалити таблиці або змінити привілеї інших користувачів.
- Запобігання людським помилкам. Обмеження прав доступу також мінімізує ризик випадкових дій із боку адміністраторів або користувачів. Якщо користувач не має доступу до критичних ресурсів, він не зможе ненавмисно викликати їхнє пошкодження або втрату.
- Спрощення аналізу інцидентів. Менший обсяг доступних ресурсів для кожного облікового запису або модуля дозволяє швидше ідентифікувати джерело проблеми або витікання, що значно покращує розслідування інцидентів безпеки.

Недоліки Least Privilege:

- Не запобігає атакам безпосередньо. Варто підкреслити, що сам по собі принцип найменших привілеїв не є механізмом активного захисту. Його основна функція полягає у мінімізації наслідків після потенційного компрометаційного інциденту. Отже, він не замінює інші методи захисту, зокрема й escaping, валідацію введення чи використання Prepared Statements.
- Складність у налаштуванні на великих системах. Детальна градація доступу та підтримка багаторівневого контролю може потребувати значних витрат часу та ресурсів, особливо у великих, динамічних середовищах. Помилки в конфігурації прав доступу також можуть призвести до відмови в обслуговуванні або витікання даних.

Після розгляду сильних і слабких сторін кожного окремого методу захисту доцільно перейти до їх комплексного порівняння.

Порівняльний аналіз: ключові метрики та їхнє оцінювання. Оцінювання ефективності засобів захисту від SQL-ін'єкцій вимагає комплексного підходу на основі формалізованих метрик, що дозволяють порівнювати різні методи за критеріями точності, продуктивності, інтеграційної складності та залежності від людського чинника. Однією з базових метрик є рівень запобігання атакам (Prevention Rate), що демонструє ефективність механізму виявлення і блокування ін'єкцій. За результатами дослідження Halfond (Halfond, Viegas, & Orso, 2006), використання підготовлених запитів дозволяє блокувати понад 95 % атак у контрольованих умовах. Іншою важливою характеристикою є продуктивнісні накладні витрати (Performance Overhead). Як показано у дослідженні Wang, & Lee (2022), впровадження параметризованих запитів у високонавантажених вебдодатках призводить до збільшення затримки виконання запитів на 2–5 %, що може бути критичним у системах реального часу. Рівень хибно позитивних спрацьовувань (False Positive Rate) відображає вірогідність того, що легітимний запит буде неправильно ідентифікований як шкідливий. Емпіричні дані, наведені в роботі García, & Müller (2022), свідчать, що сучасні WAF-системи демонструють середній показник FP на рівні 8–12 %, що вимагає ретельного налаштування політик фільтрації. Складність інтеграції (Integration Effort) оцінюється за кількістю додаткових рядків коду, які необхідно додати до проєкту, а також за кількістю зовнішніх залежностей. За оцінкою Martinez (2022), впровадження середньостатистичного middleware-захисту у Node.js вимагає додатково близько 350 рядків коду та до трьох нових пакетів. Зауважимо, що витрати на супровід і обслуговування (Maintenance Effort), як вказано Patel, Kumar, & Gupta (2023), у середньому складають від 6 до 10 годин на місяць для підтримки оновлень і адаптації захисних рішень до нових типів атак. Ще однією важливою метрикою є охоплення типових сценаріїв загроз (Coverage). Згідно з аналітичним звітом OWASP (OWASP Foundation, 2023; OWASP Top Ten Project: Coverage analysis report, 2023), лише ті системи, що включають багатоетапні перевірки й ізоляцію даних, здатні виявляти понад 90 % сценаріїв з OWASP Top 10, зокрема й складні SQL-ін'єкції. Залежність від людського чинника (Human Factor Dependence) є критичною для оцінювання довгострокової ефективності. За даними Brown, & White (2022) у середньому розробник витрачає 10–15 годин на початкове навчання роботи з інструментами безпеки, а подальші аудити потребують 2–4 години на квартал. Останнім аспектом аналізу є сумісність (Compatibility) з іншими засобами захисту, зокрема WAF, CDN і засобами статичного аналізу коду. Як продемонстровано у дослідженні Chan, & Gupta (2021), лише 78 % протестованих рішень виявилися сумісними в усіх сценаріях комплексного захисту, що вимагає узгодження між інструментами в межах загальної архітектури безпеки.

Для забезпечення цілісності аналізу доцільно структурувати згадані вище метрики й емпіричні дані в табличному форматі (табл. 1), що полегшить порівняння методів за ключовими показниками та гарантовано підвищить зрозумілість отриманих результатів. Ці дані, отримані на підставі вивчення робіт: (Halfond, Viegas, & Orso, 2006; OWASP Foundation, 2023; García, & Müller, 2022; Martinez, 2022; Patel, Kumar, & Gupta, 2023;. Brown, & White, 2022; Chan, & Gupta, 2021), що є базисом для подальшого порівняльного аналізу методів захисту від SQL-ін'єкцій.

Розглянувши та порівнявши ключові підходи до захисту вебдодатків від SQL-ін'єкцій – зокрема escaping, валідацію введення, використання білих списків (white-listing), підготовлені запити (prepared statements) та принцип найменших привілеїв (least privilege) – можна дійти висновку, що ефективна стратегія протидії таким атакам полягає не лише в окремому застосуванні зазначених технік, а у їх поєднанні в межах загальної архітектури безпеки.

У практичному середовищі розробники не реалізують ці підходи з нуля, а покладаються на готові інструменти, бібліотеки та фреймворки, які забезпечують відповідні захисні механізми за замовчуванням або через конфігурацію. Ці засоби не лише спрощують процес розроблення, але й суттєво зменшують імовірність помилок безпеки, пов'язаних із ручною реалізацією захисту.

ISSN 2707-1758

Рекомендації з посилення захисту від SQL-ін'єкцій. Для забезпечення всебічного захисту вебдодатка на базі Node.js слід реалізувати два взаємопов'язані процеси: безпечне оброблення кожного запиту (рис. 11) та безперервне тестування від моменту фіксації змін у репозиторії до розгортання у промисловому середовищі (рис. 12).

Таблиця 1

Метрики й емпіричні дані методів протидії SQL-ін'єкцій

Метод	Prevention Rate (%)	Perf. Overhead (%)	False Pos. Rate (%)	Coverage OWASP (%)	LOC для інтеграції	Maint. Effort (год/тиждень)	Human Dep. (год/квартал)	Compatibility (1–5)
Escaping	65	2	5	70	20 LOC	2	2	4
Input Validation	75	3	10	80	50 LOC	4	4	5
White-listing	85	5	2	90	30 LOC	3	3	3
Prepared Statements	99	3	1	100	15 LOC	1	1	5
Least Privilege	60	1	0	60	40 LOC	5	5	4

Перш за все, після надходження HTTP-запиту сервер спрямовує дані до модуля валідації та санітизації, де white-listing і екранування спеціальних символів усувають потенційно шкідливі фрагменти введення. У разі успішного фільтрування формується параметризований SQL-вираз із плейсхолдерами, у який підставляють лише очищені значення, що виключає можливість маніпуляції логікою запиту. Виконання такого запиту відбувається під обліковим записом із мінімальними привілеями, – цей обмежений набір прав ізолює систему від наслідків потенційного зловмисного коду. У разі помилки внутрішня інформація від СУБД перехоплюється сервером і замінюється на узагальнений код відповіді без технічних деталей, що запобігає витіканню даних про схему бази.

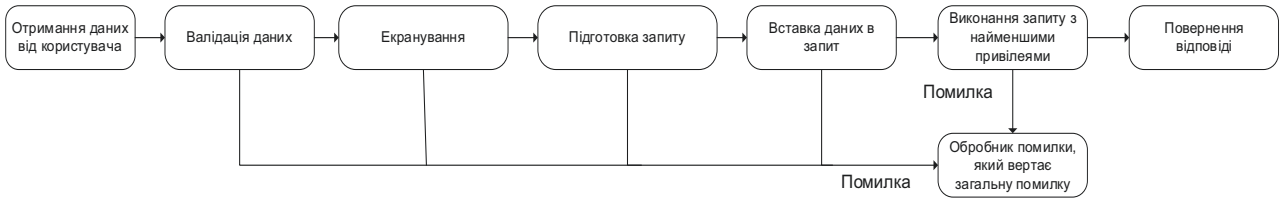


Рис. 11. Процес оброблення запиту

По-друге, кожен коміт у репозиторій ініціює конвеєр CI/CD, у межах якого першочергово виконується статичний аналіз коду (SAST) для виявлення вразливостей на рівні коду. У разі відсутності критичних зауважень CI будує Docker-образ і автоматично розгортає його в тестовому середовищі, де запускається динамічний аналіз (DAST) за допомогою OWASP ZAP для перевірки реакції API на типовий SQL-ін'єкційний трафік. Після успішного проходження цього етапу здійснюється автоматизований пентест (напр., із Burp Suite), і лише у випадку відсутності суттєвих вразливостей образ маркується як готовий до промислового середовища. Деплой у виробниче середовище супроводжується інтеграцією з APM-системами, що дозволяє в режимі реального часу відстежувати продуктивність і виявляти аномалії.

Рис. 12. Процес розгортання програмного забезпечення тестуванням на вразливості

Інтеграція цих двох процесів – безпечного оброблення кожного запиту та безперервного тестування в межах CI/CD – в єдину політику безпеки із чіткою документацією і регулярними навчаннями для розробників створює стійку, масштабовану архітектуру, яка значно знижує ймовірність успішних SQL-ін'єкцій і відповідних економічних і репутаційних втрат.

Для ілюстрації практичної реалізації цих рекомендацій у сучасному розробленні застосунків звернемося до готових інструментів:

Zod – це бібліотека для перевірки вхідних (Input Validation) даних у JavaScript і TypeScript. Вона дозволяє заздалегідь описати, якою має бути структура даних, які типи значень допускаються, які мають бути обмеження для рядків, чисел чи дат.

Перед обробленням чи збереженням даних Zod дозволяє автоматично перевірити:

- чи відповідає формат введеного тексту очікуваному;
- чи є дані правильного типу;
- чи дотримано обмежень по довжині або діапазону.

Як результат, небажані та небезпечні значення не потрапляють далі у бізнес-логіку або базу даних. Цей підхід значно знижує ймовірність SQL-ін'єкцій і забезпечує чистоту даних.

ISSN 2707-1758

ORM (Object-Relational Mapping) – це інструменти, які дозволяють працювати з базами даних через об'єкти та методи, а не через написання сирих SQL-запитів.

Цей механізм автоматично екранує спеціальні символи, включаючи лапки та крапку з комою (Escaping) і гарантує, що дані завжди обробляються як значення, а не як частина SQL-коду (Prepared Statements).

Зразок імплементації безпечного оброблення запиту на прикладі простої реєстрації з використанням сучасних бібліотек показано на рис. 13.

```
// server.js

import Fastify from 'fastify';
import { z } from 'zod';
import { DataSource } from 'typeorm';
import { Entity, PrimaryGeneratedColumn, Column } from
'typeorm';
import bcrypt from 'bcrypt';

// Опис сутності User для TypeORM
@Entity()
class User {
  @PrimaryGeneratedColumn()
  id;

  @Column({ unique: true })
  email;

  @Column()
  password;
}

// Ініціалізація DataSource (PostgreSQL)
const AppDataSource = new DataSource({
  type: 'postgres',
  host: 'localhost',
  port: 5432,
  username: 'node_app_user',      // обліковий запис із
мінімальними привілеями
  password: 'securePassword!',
  database: 'node_app_db',
  entities: [User],
  synchronize: false,
});
```

```
// Ініціалізація Fastify-сервера
const fastify = Fastify({ logger: true });

// Схема валідації для реєстрації
const registerSchema = z.object({
  email: z.string().email(),
  password: z.string().min(8).max(30),
});

// Плагін для підключення до БД
fastify.register(async () => {
  try {
    await AppDataSource.initialize();
    fastify.log.info('DataSource initialized');
  } catch (err) {
    fastify.log.error('Error during DataSource
initialization', err);
    process.exit(1);
  }
});

// Ендпоінт реєстрації користувача
fastify.post('/register', async (request, reply) => {
  // Валідація вхідних даних
  const result = registerSchema.safeParse(request.body);
  if (!result.success) {
    throw fastify.httpErrors.badRequest('Некоректні
вхідні дані');
  }
  const { email, password } = result.data;

  try {
    // Хешування пароля
    const saltRounds = 10;
    const hashed = await bcrypt.hash(password,
saltRounds);
```

```
    // Збереження користувача через TypeORM
    const repo = AppDataSource.getRepository(User);
    const user = repo.create({ email, password: hashed
});
    const saved = await repo.save(user);

    return reply.code(201).send({ id: saved.id });
  } catch (err) {
    fastify.log.error(err);
    throw fastify.httpErrors.badRequest('Bad data');
  }
});

// Запуск сервера
const start = async () => {
  try {
    await fastify.listen({ port: 3000 });
    fastify.log.info(`Server listening on
${fastify.server.address().port}`);
  } catch (err) {
    fastify.log.error(err);
    process.exit(1);
  }
};
start();
```

Рис. 13. Зразок імплементації безпечного оброблення запиту на прикладі простої реєстрації

Дискусія і висновки

1. У ході дослідження здійснено всебічний аналіз актуальних загроз для вебдодатків, побудованих на платформі Node.js. Виявлено, що одними з найнебезпечніших залишаються SQL-ін'єкції, атаки на рівні API, XSS, CSRF, а також атаки через HTTP-заголовки. Причинами їхнього виникнення є як архітектурні особливості асинхронного оброблення запитів у Node.js, так і недотримання принципів безпечного програмування, використання застарілих залежностей, відсутність валідації введення. З огляду на зростання кількості атак, зокрема і з боку автоматизованих ботмереж і складних сканерів, проблема набуває ще більшої актуальності в умовах цифрової трансформації.

2. У процесі дослідження систематизовано найпоширеніші методи атак, що застосовуються для компрометації даних, та охарактеризовано джерела їхнього виникнення. Основну увагу приділено SQL-ін'єкціям, які виникають за недостатнього контролю вхідних даних із боку користувача. Продемонстровано, як атаки можуть реалізовуватись через вебформи, URL-параметри, API-запити, HTTP-заголовки, а також WebSocket-повідомлення. Детальна класифікація SQL-ін'єкцій (in-band, blind, out-of-band) дозволила встановити, що різні типи атак потребують специфічних засобів виявлення та нейтралізації, що ускладнює завдання для розробників.

3. Визначено й охарактеризовано ключові принципи і механізми захисту вебдодатків, реалізованих на Node.js. До таких механізмів віднесено: екранування спеціальних символів (escaping), валідацію введених даних (input validation), перевірку за білим списком (white-listing), використання підготовлених запитів (prepared statements), а також принцип найменших привілеїв (least privilege). Кожен із методів проаналізовано з позицій ефективності у запобіганні атакам, простоти впровадження, продуктивності, рівня залежності від людського фактора та сумісності з іншими засобами захисту. Встановлено, що поєднання кількох методів дає змогу досягнути багаторівневого та стійкого захисту.

4. Досліджено способи підвищення захищеності системної взаємодії та оброблення запитів у Node.js-додатках. Запропоновано впровадження бібліотек для типізованої валідації (напр., Zod), використання ORM для забезпечення відокремлення даних і запитів, застосування рольової моделі доступу (RBAC), а також чітке обмеження прав облікових записів і процесів відповідно до принципу найменших привілеїв. Показано, що реалізація цих заходів дає змогу суттєво зменшити поверхню атаки та підвищити загальну стійкість вебдодатків до компрометації.

5. Сформульовано комплекс практичних рекомендацій із побудови захищених вебдодатків, що відповідають сучасним стандартам кібербезпеки. Зокрема запропоновано: реалізовувати валідацію та санітизацію на кожному етапі оброблення запиту; використовувати підготовлені запити та middleware для захисту бази даних; налаштовувати розмежування доступу на рівні ролей і сервісів; здійснювати безперервне тестування безпеки в межах CI/CD-процесів із застосуванням SAST, DAST та автоматизованих пентестів. Окремо наголошено на доцільності впровадження контролю версій залежностей і регулярного оновлення компонентів.

6. Підкреслено важливість системного, інтегрованого підходу до забезпечення безпеки, який виходить за межі суто технічних рішень. Ефективна стратегія захисту передбачає не лише використання захисних бібліотек та інструментів, але й розроблення внутрішніх політик безпеки, навчання розробників, проведення регулярних аудитів, а також контроль за відповідністю вебдодатків стандартам типу OWASP

ISSN 2707-1758

Top 10, ISO/IEC 27001, PCI DSS. Лише комплексний і динамічний підхід дозволяє забезпечити надійність і безперервність функціонування сучасних інформаційних систем в умовах еволюції кіберзагроз.

Внесок авторів: Іван Опірський – концептуалізація дослідження; методологія; загальне керівництво проєктом; аналіз сучасних загроз кібербезпеки; формулювання мети та завдань дослідження; розроблення теоретичної бази дослідження SQL-ін'єкцій; огляд літератури та аналіз останніх досліджень; формулювання загальних висновків і перспектив подальших досліджень. Сергій Корень – збір і систематизація емпіричних даних про методи захисту; детальний аналіз принципів дії SQL-ін'єкцій; дослідження джерел виникнення атак; класифікація типів SQL-ін'єкцій; розроблення та тестування практичних прикладів коду; валідація запропонованих методів захисту; підготовка порівняльної таблиці метрик. Антон Гундерич – аналіз методів захисту від SQL-ін'єкцій (Escaping, Input Validation, White-listing, Prepared Statements, Least Privilege); порівняльний аналіз ефективності захисних механізмів; розроблення практичних рекомендацій із посилення захисту; створення схем і діаграм процесів; візуалізація даних; підготовка графіків і рисунків; розроблення зразків безпечної імплементації з використанням сучасних бібліотек.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Brown, T., & White, K. (2022). The human factor in web application security: Training and auditing requirements. *International Journal of Cyber Education,* 8(3), 101–118.

Chan, E., & Gupta, M. (2021). Compatibility of combined security solutions: WAF, CDN, and static analysis. In *Proceedings of the IEEE Symposium on Security and Privacy* (pp. 210–224). IEEE.

Edgescan. (2024). *2023 vulnerability statistics report*. https://www.edgescan.com/wp-content/uploads/2024/03/2023-Vulnerability-Statistics-Report.pdf

García, F., & Müller, J. (2022). Empirical analysis of false-positive rates in web application firewalls. *ACM Transactions on Privacy and Security,* 25(3), Article 11, 1–25.

Halfond, W. G., Viegas, J., & Orso, A. (2006). A classification of SQL-injection attacks and countermeasures. In *Proceedings of the International Symposium on Secure Software Engineering* (pp. 1045–1051). IEEE.

IBM. (n.d.). *X-Force threat intelligence index*. https://www.ibm.com/reports/threat-intelligence

Intigriti. (2024). *The cyber threat landscape part 4: Emerging technologies and their security implications*. https://www.intigriti.com/blog/business-insights/the-cyber-threat-landscape-part-4-emerging-technologies/-and-their-security-implic

Ishaq, K., & Fareed, S. (2023). *Mitigation techniques for cyber attacks: A systematic mapping study*. arXiv. https://www.researchgate.net/publication/373450471

Jabeen, F., Li, X., & Kim, S. (2025). *Cybersecurity threats in FinTech: A systematic review*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0957417423031998

Martinez, P. J. (2022). Code complexity and integration effort for security middleware in Node.js. *International Journal of Software Engineering and Security,* 10(1), 45–62.

OWASP. (n.d.). *Testing for SQL injection*. https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application/_Security/_Testing/07-Input_Validation_Testing/02-Testing_for_SQL_Injection

OWASP Cheat Sheet Series. (n.d.). *SQL injection prevention cheat sheet*. https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_/Prevention_Cheat_Sheet.html

OWASP Foundation. (2025). *OWASP Top 10*. OWASP. https://owasp.org/www-project-top-ten/

OWASP Foundation. (2023). *OWASP Top Ten Project: Coverage analysis report*. https://owasp.org

Patel, R., Kumar, S., & Gupta, A. (2023). Maintenance overhead of web security frameworks: An empirical study. *Software Quality Journal,* 31(4), 1203–1220.

Security vulnerabilities and protective strategies for graphical passwords. (2023). *Electronics,* 13(15), 3042. https://www.mdpi.com/2079-9292/13/15/3042

Snyk. (2024). *2024 open source security report: Slowing progress and new challenges*. https://snyk.io/blog/2024-open-source-security-report-/slowing-progress-and-new-challenges-for/

Snyk. (2025). *Preventing SQL injection attacks in Node.js*. https://snyk.io/blog/preventing-sql-injection-attacks-node-js/

Wang, X., & Lee, Y. (2022). Performance overhead of parameterized queries in high-load web applications. *Journal of Web Security Studies,* 14(2), 75–89.

Zhang, L., Chen, Y., Wang, R., & Liu, X. (2023). Security vulnerabilities and protective strategies for graphical passwords. *Electronics,* 13(15), 3042. https://www.mdpi.com/2079-9292/13/15/3042

Zhang, Y., & Lee, M. (2025). *Modern hardware security: A review of attacks and countermeasures*. arXiv. https://arxiv.org/abs/2501.04394

References

Brown, T., & White, K. (2022). The human factor in web application security: Training and auditing requirements. *International Journal of Cyber Education,* 8(3), 101–118.

Chan, E., & Gupta, M. (2021). Compatibility of combined security solutions: WAF, CDN, and static analysis. In *Proceedings of the IEEE Symposium on Security and Privacy* (pp. 210–224). IEEE.

Edgescan. (2024). *2023 vulnerability statistics report*. https://www.edgescan.com/wp-content/uploads/2024/03/2023-Vulnerability-Statistics-Report.pdf

García, F., & Müller, J. (2022). Empirical analysis of false-positive rates in web application firewalls. *ACM Transactions on Privacy and Security,* 25(3), Article 11, 1–25.

Halfond, W. G., Viegas, J., & Orso, A. (2006). A classification of SQL-injection attacks and countermeasures. In *Proceedings of the International Symposium on Secure Software Engineering* (pp. 1045–1051). IEEE.

IBM. (n.d.). *X-Force threat intelligence index*. https://www.ibm.com/reports/threat-intelligence

Intigriti. (2024). *The cyber threat landscape part 4: Emerging technologies and their security implications*. https://www.intigriti.com/blog/business-insights/the-cyber-threat-landscape-part-4-emerging-technologies/-and-their-security-implic

Ishaq, K., & Fareed, S. (2023). *Mitigation techniques for cyber attacks: A systematic mapping study*. arXiv. https://www.researchgate.net/publication/373450471

Jabeen, F., Li, X., & Kim, S. (2025). *Cybersecurity threats in FinTech: A systematic review*. ScienceDirect. https://www.sciencedirect.com/science/article/abs/pii/S0957417423031998

Martinez, P. J. (2022). Code complexity and integration effort for security middleware in Node.js. *International Journal of Software Engineering and Security,* 10(1), 45–62.

OWASP. (n.d.). *Testing for SQL injection*. https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security/_Testing/07-Input_Validation_Testing/02-Testing_for_SQL_Injection

OWASP Cheat Sheet Series. (n.d.). *SQL injection prevention cheat sheet*. https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_/Prevention_Cheat_Sheet.html

OWASP Foundation. (2025). *OWASP Top 10*. OWASP. https://owasp.org/www-project-top-ten/

OWASP Foundation. (2023). *OWASP Top Ten Project: Coverage analysis report*. https://owasp.org

Patel, R., Kumar, S., & Gupta, A. (2023). Maintenance overhead of web security frameworks: An empirical study. *Software Quality Journal,* 31(4), 1203–1220.

Security vulnerabilities and protective strategies for graphical passwords. (2023). *Electronics,* 13(15), 3042. https://www.mdpi.com/2079-9292/13/15/3042

Snyk. (2024). *2024 open source security report: Slowing progress and new challenges*. https://snyk.io/blog/2024-open-source-security-report-/slowing-progress-and-new-challenges-for/

Snyk. (2025). *Preventing SQL injection attacks in Node.js*. https://snyk.io/blog/preventing-sql-injection-attacks-node-js/

Wang, X., & Lee, Y. (2022). Performance overhead of parameterized queries in high-load web applications. *Journal of Web Security Studies,* 14(2), 75–89.

Zhang, L., Chen, Y., Wang, R., & Liu, X. (2023). Security vulnerabilities and protective strategies for graphical passwords. *Electronics,* 13(15), 3042. https://www.mdpi.com/2079-9292/13/15/3042

Zhang, Y., & Lee, M. (2025). *Modern hardware security: A review of attacks and countermeasures*. arXiv. https://arxiv.org/abs/2501.04394

Отримано редакцією журналу / Received: 16.06.25
Прорецензовано / Revised: 19.06.25
Схвалено до друку / Accepted: 30.06.25

ISSN 2707-1758

Ivan OPIRSKYY, DSc (Engin.), Prof.
ORCID ID: 0000-0002-8461-8996
e-mail: ivan.r.opirskyi@lpnu.ua
Lviv Polytechnic National University, Lviv, Ukraine

Serhii KOREN, Student
ORCID ID: 0009-0003-9239-136X
e-mail: serhii.koren.kb.2023@lpnu.ua
Lviv Polytechnic National University, Lviv, Ukraine

Anton HUNDERYCH, Student
ORCID ID: 0009-0005-0207-581X
e-mail: anton.hunderych.kb.2023@lpnu.ua
Lviv Polytechnic National University, Lviv, Ukraine

IMPLEMENTATION OF WEB APPLICATION SECURITY ON NODE.JS: MAIN THREATS AND SECURITY METHODS

Background. *The intensive development of web technologies and the growing popularity of web applications developed on the Node.js platform open new opportunities for digital business while simultaneously increasing cyber threat risks. One of the key problems of modern cybersecurity is protecting such systems from unauthorized access, data integrity violations, and ensuring their stable operation. In the context of increasing attacks on web resources, security measures that can be integrated without significant performance degradation become particularly relevant. The application of multi-level mechanisms, including access control, input validation, and database query parameterization, forms the foundation of the modern approach to secure development.*

Methods. *This work conducted a systematic analysis of modern methods for ensuring web application security, particularly in the Node.js context. Methods of theoretical modeling, comparative analysis, practical testing of security systems, and effectiveness analysis of various strategies were used. Special attention was paid to the integration of protection mechanisms such as rate limiting, parameterized SQL query processing, user input filtering, and application of the principle of least privilege. For each method, the level of performance load, attack detection accuracy, and compatibility with Node.js architecture were evaluated.*

Results. *The analysis showed that the most effective approach to web application protection is combining several complementary strategies. For example, using parameterized queries significantly reduces the risk of SQL injections, while access control to critical resources prevents unauthorized data modification. It was proven that combining rate limiting and input filtering significantly increases application resistance to brute force and script injection attacks. At the same time, such measures do not create substantial system load, allowing their implementation in real-world conditions. Optimal configurations of protective mechanisms were determined depending on the threat level and functional requirements of the web application.*

Conclusions. *Protecting web applications built on Node.js requires a systematic and comprehensive approach. Combining several protection methods allows achieving a high level of security without reducing application efficiency. The research results can be used to build integrated systems for detecting and preventing cyber threats, taking into account the architectural features of Node.js. Beyond technical aspects, the importance of implementing security policies that encompass both technological and organizational protection components is emphasized. Systematic implementation of such approaches will ensure application resilience even under conditions of increasing cyber threat complexity.*

Keywords: *Node.js, web application security, SQL injections, access control, rate limiting, input filtering, multi-level protection, data confidentiality, software vulnerability, cybersecurity.*

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

КОМП'ЮТЕРНІ НАУКИ

УДК 004.8:004.65:004.728.8:004.056
DOI: https://doi.org/10.17721/ISTS.2025.9.74-80

Віктор МОРОЗОВ, канд. техн. наук, проф.
ORCID ID: 0000-0001-7946-0832
e-mail: viktor.morozov@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Владислав ДЕЙНЕГА, асп.
ORCID ID: 0009-0008-3123-2302
e-mail: thedynkan@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

ВИЯВЛЕННЯ СПАМУ В ТЕКСТОВИХ ПОВІДОМЛЕННЯХ ІЗ ВИКОРИСТАННЯМ ЛОГІСТИЧНОЇ РЕГРЕСІЇ НА БАЗІ ГРАДІЄНТНОГО СПУСКУ

Вступ. *Зі зростанням обсягів електронного листування проблема фільтрації спаму набуває все більшої актуальності. За даними статистичних досліджень, спам становить значну частку глобального поштового трафіка, що створює ризики як для безпеки, так і для ефективності електронної комунікації. У цьому контексті особливого значення набувають методи оброблення природної мови (NLP) та машинного навчання. Метою цієї роботи є побудова моделі для класифікації електронних повідомлень на спам і не спам, із використанням логістичної регресії, реалізованої через градієнтний спуск, у поєднанні з методами оброблення текстових даних.*

Методи. *Для навчання моделі використано датасет, що містить понад 5000 електронних повідомлень, мічених як спам або не спам. Дані було попередньо очищено з видаленням шумових компонентів: пунктуації, цифр, стоп-слів, коротких слів, а також застосовано лематизацію. Тексти перетворено на числову форму за допомогою TF-IDF векторизації із L2-нормалізацією. Для боротьби з дисбалансом між класами застосовано метод SMOTE. Навчання моделі здійснювалось за класичною схемою градієнтного спуску з використанням сигмоїдної функції активації та логарифмічної функції втрат.*

Результати. *Побудована модель досягла високих результатів на тестовій вибірці: загальна точність становила 98 %, f1-score для класу спам – 0.92, а для не спаму – 0.99. Значення recall для спаму дорівнювало 0.90, що свідчить про здатність моделі виявляти більшість небажаних повідомлень без надмірних помилкових спрацьовувань. Баланс precision і recall також підтверджується макросереднім і зваженим середнім f1-показником понад 0.96.*

Висновки. *Результати дослідження засвідчили ефективність поєднання логістичної регресії, градієнтного спуску та текстового препроцесингу для задачі класифікації спаму навіть за умов дисбалансованих даних. Запропонований підхід є ефективним й інтерпретованим, що робить його придатним для практичного застосування в системах фільтрації електронної пошти.*

Ключові слова: *машинне навчання, оброблення природної мови, градієнтна оптимізація, фільтрація електронної пошти, текстовий препроцесинг, класифікація.*

Вступ

В сучасному цифровому просторі електронна пошта залишається одним із ключових засобів комунікації, що використовується як приватними користувачами, так і комерційними структурами. Водночас цей канал щодня генерує колосальні обсяги повідомлень – за оцінками, понад 330 млрд електронних листів щодня надсилається по всьому світу. Серед них, за статистикою, спам-повідомлення становлять приблизно 46 % загального трафіка, що свідчить про стабільно високу частку небажаних листів у комунікаційній інфраструктурі. Отже, щодня у світі поширюється понад 150 млрд спам-листів. Такі повідомлення можуть містити рекламний або шахрайський контент, шкідливі вкладення або фішингові посилання, що створює не лише інформаційне перевантаження для користувача, а й потенційні ризики для безпеки даних. Ці показники підтверджують і масштабність, і сталість проблеми, що актуалізує потребу в розробленні ефективних та адаптивних систем автоматичної фільтрації небажаної електронної кореспонденції.

Проблема автоматичного виявлення спаму полягає в необхідності точно розпізнавати небажані повідомлення на основі їхнього текстового вмісту. Прості методи фільтрації вже недостатньо ефективні, тому зростає потреба в алгоритмах, здатних аналізувати зміст повідомлень і виявляти приховані шаблони. У зв'язку із цим зростає актуальність застосування методів оброблення природної мови (NLP) та машинного навчання, які дозволяють автоматично виявляти закономірності у текстах на основі попередньо навчених моделей.

Серед найбільш інтерпретованих і водночас ефективних моделей для задач бінарної класифікації – логістична регресія. Її особливість полягає в тому, що вона дає змогу отримувати ймовірнісні передбачення та не потребує надмірної кількості ресурсів. У цьому дослідженні логістична регресія реалізується через градієнтний спуск – класичний метод оптимізації, що дозволяє поступово оновлювати ваги моделі з урахуванням похибки.

Важливою передумовою якісного навчання є глибокий текстовий препроцесинг, який охоплює очищення повідомлень від пунктуації, чисел, надлишкових пробілів, стоп-слів і коротких слів, а також лематизацію. Саме цей етап дозволяє зменшити шум у даних і виділити лише ті слова, що несуть змістове навантаження. Далі тексти перетворено на числову форму за допомогою TF-IDF векторизації, що дозволяє враховувати не тільки частоту терміна, а і його інформативність у межах усього корпусу. У векторизації застосовують L2-нормалізацію, яка робить ознаки порівнюваними за масштабом і покращує роботу оптимізаційного алгоритму.

Окремим викликом у роботі з реальними даними є дисбаланс між класами, коли повідомлення певного типу (зазвичай – не спам) значно переважають. У таких

ISSN 2707-1758

умовах важливо застосовувати методи балансування, які дають змогу зробити навчання моделі чутливішим до менш представленого класу. У нашому дослідженні для цього використано підхід синтетичного збільшення вибірки (SMOTE – Synthetic Minority Over-sampling Technique), який дозволяє зберегти обсяг даних класу більшості, водночас підвищуючи представленість меншості.

Метою дослідження є розроблення моделі для класифікації електронних повідомлень на спам і не спам на основі логістичної регресії, оптимізованої методом градієнтного спуску, із залученням методів оброблення текстових даних.

Огляд літератури. Дослідження "Spam Statistics 2025: New Data on Junk Email, AI Scams & Phishing" (Spam Statistics, 2025) надає актуальні статистичні дані щодо спаму, фішингу та пов'язаних із ними загроз у сфері електронної пошти. Згідно з отриманими результатами, у 2023 р. щодня надсилалося близько 160 млрд спам-листів, що становило приблизно 46 % загального обсягу електронної пошти. Це свідчить про значну частку небажаних повідомлень у глобальному електронному трафіку. Опитування також показало, що 96,8 % користувачів отримували спам-повідомлення в тій чи іншій формі. Найпоширенішими темами спаму були: призи та розіграші (36,7 %), пропозиції роботи (36,3 %) та банківські повідомлення (34,6 %). Ці дані підкреслюють, що спамери часто використовують теми, які можуть привернути увагу користувачів і спонукати їх до взаємодії з повідомленням. Крім того, дослідження виявило, що 68,8 % респондентів повідомили про негативний вплив спаму та фішингових повідомлень на їхній психічний стан, що свідчить про психологічні наслідки таких загроз. Також зазначено, що фінансові установи є найчастішими цілями фішингових атак, отримуючи 27,7 % таких повідомлень. Це дослідження надає цінну інформацію для розуміння масштабів та еволюції спаму, а також підкреслює необхідність розроблення ефективних методів виявлення та фільтрації небажаних повідомлень.

Публікація автора Jyothiikaa Moorthy (2025) підтверджує ключові висновки попереднього дослідження щодо масштабності проблеми спаму, зафіксувавши його частку на рівні 45,6 % у 2023 р. з подальшим зростанням до 46,8 % у грудні 2024 р. Також проінформовано, що майже 96 % фішингових атак здійснюють за допомогою електронної пошти. Окрім кількісних даних, звіт також доповнює попереднє джерело деталізацією тематики спаму: найпоширенішими є маркетингові повідомлення (36 %), контент для дорослих (31,7 %) і повідомлення, пов'язані з фінансовими питаннями (26,5 %). Така типізація розширює уявлення про природу спаму й підкреслює потребу в адаптивних моделях виявлення залежно від його змісту.

У статті (Khanday et al., 2021) розглянуто використання логістичної регресії для класифікації електронних листів на спам і не спам. Автори підкреслюють універсальність логістичної регресії як одного з найкращих методів для класифікації електронних листів. Вказана стаття може бути корисною для дослідження, оскільки вона демонструє практичне застосування логістичної регресії у задачах класифікації спаму.

У роботі (Zhang et al., 2019) логістичну регресію застосовують до задачі класифікації у фінансовому секторі за умов значного дисбалансу класів. Автори демонструють, що навіть у таких складних умовах, із правильною адаптацією, цей метод залишається ефективним. Цей підхід є релевантним до нашого дослідження, де модель логістичної регресії також використовується в задачі з дисбалансованими даними, і доводить, що у разі належної структури оброблення даних модель здатна зберігати високу чутливість до меншості класу.

У статті (Khlevna, & Deineha, 2023) також розглянуто використання логістичної регресії для класифікації шахрайства, де так само можна бачити важливість попереднього оброблення даних, яке є одним із важливих елементів роботи.

У дослідженні (Rakhmanov, 2020) на підставі аналізу тональності в освітньому середовищі проведено порівняння методів векторизації та класифікації на основі понад 52 000 текстових коментарів. Автори показали, що Tf-IDF – Term Frequency–Inverse Document Frequency – перевершує CountVectorizer за точністю класифікації, що підтверджує доцільність використання саме цього підходу у задачах текстової класифікації. Отримані результати узгоджуються з підходом, застосованим у нашій роботі, де TF-IDF також використовується як основна техніка векторного подання тексту.

Зауважимо, що в дослідженні (Papageorgiou, Economou, & Bersimis, 2024) з класифікації бізнес-електронної пошти розглянуто ті самі ключові етапи, що і в нашій роботі: препроцесинг, векторизація тексту та класифікація email-повідомлень, що підтверджує актуальність обраного підходу.

Mohammed, N., Mouhajir, M., & Yassine S. (2023) описали логістичну регресію з градієнтним спуском, що прямо відповідає підходу, застосованому і в нашому дослідженні. Обраний метод навчання є релевантним і демонструє практичну цінність градієнтного спуску для задач класифікації.

Методи

Попереднє оброблення текстових даних є критично важливою складовою будь-якого завдання машинного навчання, пов'язаного з обробленням природної мови, оскільки саме на цьому етапі формується чисте, лаконічне й інформативне представлення вхідного тексту, яке згодом буде перетворене у вектори для навчання моделі. У межах цього дослідження реалізовано поетапний підхід до оброблення текстів, який охоплює лінгвістичне очищення та структуроване скорочення шуму в даних.

На першому етапі всі електронні повідомлення було перетворено у нижній регістр для уникнення зайвої варіативності слів, що мають однакову семантику, але різне написання. Це дозволяє уникнути дублювання слів, які відрізняються лише регістром, наприклад "Free" та "free". Далі застосовано регулярні вирази для видалення чисел, знаків пунктуації, символів нового рядка та надлишкових пробілів, що часто не несуть смислового навантаження у задачах класифікації спаму.

Після цього текст розбивався на токени – окремі слова – за допомогою стандартного токенізатора бібліотеки nltk. Кожен токен перевірявся на наявність у стандартному списку англійських стоп-слів, і непотрібні функціональні слова (як-от "the", "is", "and") були виключені. Це дозволяє моделі зосередитись на більш значущих елементах вхідного тексту.

Наступним етапом була лематизація, тобто приведення кожного токена до його початкової граматичної форми. Для цього використовували лематизацію з урахуванням частини мови (POS-теги),

ISSN 2707-1758

що дозволяє досягти кращої точності порівняно з простим зведенням до базової форми (рис. 1).

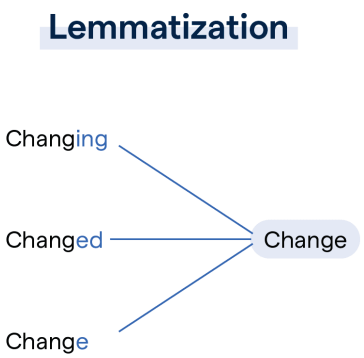


Рис. 1. Ілюстрація процесу лематизації

Додатково запроваджено фільтрацію коротких слів, зокрема тих, що мають довжину менше трьох символів. Це дозволяє зменшити кількість шумових токенів, які часто мають низьку інформативність.

В результаті оброблення кожне повідомлення перетворено в очищений і лематизований текст, з якого видалено шумові та малозначущі елементи. Такий підхід дозволяє зберегти змістовну суть повідомлення та забезпечує узгоджене представлення всіх текстів у корпусі.

Цей етап препроцесингу є критично важливим для коректної роботи моделі логістичної регресії, яка не має внутрішнього механізму розуміння тексту, і працює винятково із числовим представленням вхідних даних. У нашій роботі завдяки структурованому препроцесингу вдалося підготувати тексти до векторизації через TF-IDF, забезпечивши як однорідність корпусу, так і збереження релевантної лексичної інформації, необхідної для успішного навчання моделі.

Після завершення етапу текстового препроцесингу всі вхідні повідомлення мають вигляд очищених рядків, що складаються з лематизованих слів без пунктуації, чисел, стоп-слів і зайвих символів. У такому вигляді текст все ще залишається неструктурованим і непридатним до оброблення алгоритмом машинного навчання, оскільки модель працює виключно із числовими значеннями. Тому наступним важливим етапом є перетворення текстових документів у векторну форму – процес, відомий як векторизація.

Після завершення препроцесингу всі очищені повідомлення було розділено на навчальну та тестову вибірки у пропорції 70:30. Такий розподіл дозволяє забезпечити об'єктивне оцінювання моделі на нових, раніше не бачених даних, і запобігає переоцінюванню її точності.

У цьому дослідженні для представлення текстів у вигляді числових векторів використано метод TF-IDF, який враховує не лише наявність слова у повідомленні, а і його важливість у межах усього корпусу. Цей підхід дозволяє приділяти більше уваги тим словам, які є характерними для окремих повідомлень, і приглушувати вагу слів, що часто трапляються повсюдно. Наприклад, якщо слово часто зустрічається в усіх повідомленнях, то його інформативність вважається нижчою.

Метод TF-IDF поєднує два складники. Перший – TF (term frequency) – відображає частоту появи терміна у конкретному документі. Другий – IDF (inverse document frequency) – зменшує вагу слів, що зустрічаються у багатьох документах, і підвищує значущість тих, які є рідкісними, але характерними. Добуток цих двох значень формує остаточну вагу кожного слова в повідомленні. У результаті векторизації кожен документ отримує числове представлення, яке відображає важливість використаних у ньому термінів. Формула для обчислення TF-IDF:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right), \quad (1)$$

де $w_{i,j}$ – вага (важливість) терміна i в документі j, $tf_{i,j}$ – частота терміна i в документі j (Term Frequency), N – загальна кількість документів, df_i – кількість документів, у яких зустрічається термін i (Document Frequency).

Для реалізації TF-IDF векторизації використано TfidfVectorizer із бібліотеки scikit-learn. Попередньо очищені повідомлення спочатку були подані до методу fit_transform() для навчального набору, що дозволило побудувати словник усіх термінів та обчислити відповідні ваги. Тестова вибірка векторизується за допомогою методу transform() – виключно на основі словника, отриманого з навчальних даних. Такий підхід запобігає витіканню інформації з тестової вибірки до моделі та забезпечує коректність оцінювання якості класифікації.

У процесі векторизації була застосована L2-нормалізація, яка перетворює кожен текстовий вектор так, щоб його довжина у просторі ознак дорівнювала одиниці. Це сприяє уніфікації масштабів ознак і забезпечує коректну роботу оптимізатора. Нормалізація є важливою складовою підготовки даних для ефективного застосування градієнтного спуску.

У результаті етапу векторизації весь текстовий корпус було перетворено у числову форму, придатну для навчання моделі логістичної регресії. Отримані вектори містять зважену інформацію про частотність і значущість слів у контексті задачі, що забезпечує якісну основу для подальшої класифікації повідомлень як спам або не спам. Цей підхід поєднує простоту реалізації та високу інформативність, що робить TF-IDF ефективним інструментом у задачах оброблення природної мови.

Однією з характерних особливостей задач класифікації у сфері електронної пошти є суттєвий дисбаланс класів. У типових наборах даних спостерігається значна перевага одного класу – наприклад, легітимних повідомлень – над іншим, яким часто є спам. У нашому дослідженні такий дисбаланс початково становив близько 13 % і до 87 %, а це створює загрозу, що модель навчиться надавати перевагу домінуючому класу, ігноруючи менш представлений.

Щоб компенсувати цей ефект, застосовано метод штучного збільшення вибірки меншості – SMOTE. Вказаний метод SMOTE є одним із найпопулярніших підходів до oversampling і полягає у синтетичному створенні нових прикладів меншості на основі наявних. На відміну від простого дублювання, SMOTE генерує нові точки інтерполяцією між наявними прикладами того самого класу. Це дозволяє моделі бачити більшу варіативність прикладів меншості і краще навчатися на її представленнях (рис. 2).

SMOTE застосовано лише до навчальної вибірки (X_train і y_train). Тестова вибірка залишалася незмінною для забезпечення об'єктивності та репрезентативності оцінки продуктивності моделі. Такий підхід узгоджується із загальноприйнятими практиками машинного навчання і запобігає "витіканню інформації" з тестових даних до процесу навчання.

ISSN 2707-1758

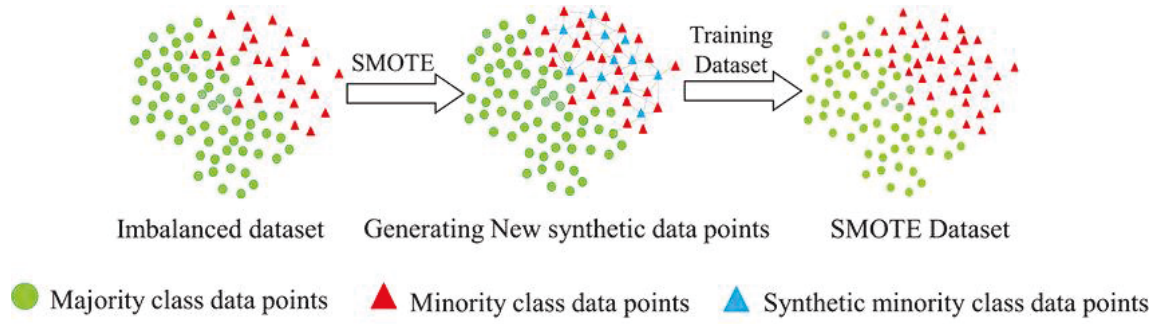


Рис. 2. Візуалізація процесу застосування методу SMOTE

Перед застосуванням SMOTE усі вхідні тексти вже були подані у векторизованому TF-IDF вигляді, а отже, представлені у вигляді числових ознак, що є необхідною умовою для його коректної роботи. SMOTE не працює безпосередньо з текстами, а лише із числовим простором ознак – тому його місце у конвеєрі оброблення даних розташовано після векторизації, але до навчання моделі.

Для реалізації методу SMOTE використовувався модуль SMOTE з бібліотеки imbalanced-learn. Алгоритм базується на підході, який використовує *k*-ближчих сусідів для генерації нових синтетичних зразків класу меншості, зберігаючи його основну структуру у векторному просторі.

Рішення обмежитися лише SMOTE без використання undersampling пояснюється прагненням зберегти всі наявні приклади класу більшості, оскільки видалення частини даних могло б призвести до втрати важливої інформації та зниження загальної стійкості моделі.

Після балансування класів за допомогою SMOTE навчальна вибірка стала симетричною, що дозволяє алгоритму логістичної регресії навчатись без упередженості до одного з класів. Отже, застосування SMOTE стало ключовим етапом у підготовці навчальних даних до моделювання і дозволило уникнути типових помилок, пов'язаних з ігноруванням меншості класу, а також забезпечити збалансовану основу для побудови ефективної класифікаційної моделі.

До векторного представлення повідомлень було додано допоміжну колонку зі значенням 1 для кожного прикладу. Це стандартний технічний прийом, який дозволяє моделі враховувати зміщення під час навчання.

Для розв'язання задачі бінарної класифікації в цьому дослідженні застосовується логістична регресія, яка дозволяє прогнозувати ймовірність належності об'єкта до одного з двох класів. На відміну від лінійної регресії, логістична модель обмежує вихід значенням у межах інтервалу [0;1], що дозволяє трактувати результат як імовірність позитивного класу – у нашому випадку, повідомлення, класифікованого як "спам".

Ключовим елементом логістичної регресії є сигмоїдна функція активації, яка перетворює лінійну комбінацію вхідних ознак у значення від 0 до 1:

$$\sigma(x) = \frac{1}{1+e^{-x}}, \quad (2)$$

де *x* – вхідне значення (лінійна комбінація вхідних ознак та їхніх ваг), *e* – число Ейлера.

Ваги визначають у процесі навчання моделі на навчальному наборі даних. Таким способом, кожному прикладу присвоюється ймовірність належності до позитивного класу. Остаточне рішення приймається шляхом порівняння з порогом (зазвичай, 0,5): якщо значення сигмоїди більше порога – прогнозується клас 1, в іншому разі – клас 0.

Навчання моделі здійснюється за допомогою градієнтного спуску – одного з найпоширеніших чисельних методів оптимізації. Ідея цього методу полягає у поступовому оновленні параметрів моделі в напрямку, який найшвидше зменшує значення функції втрат. У логістичній регресії такою функцією виступає логарифмічна функція втрат (Log*Loss*):

$$\mathrm{Log}Loss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log(p_i) + (1-y_i)\cdot \log(1-p_i)), \quad (3)$$

де *N* – кількість спостережень, y_i – фактичний бінарний результат (0 або 1) для *i*-го спостереження, p_i – прогнозована ймовірність того, що *i*-те спостереження належить до класу 1.

Оновлення ваг відбувається за формулою

$$\theta_{new} = \theta_{old} - \alpha \cdot \nabla J(\theta), \quad (4)$$

де θ – параметри моделі, α – швидкість навчання, а $\nabla J(\theta)$ – градієнт функції втрат $J(\theta)$.

Отже, під час кожної ітерації модель оновлює параметри у напрямку зменшення похибки. Процес продовжується доти, поки або зміни у функції втрат не стануть незначними, або не буде досягнуто заданої кількості ітерацій.

Важливим аспектом є початкова ініціалізація ваг, яка може здійснюватися нульовими або малими випадковими значеннями. Також критичним є вибір параметра α, оскільки надто велике значення може спричинити нестабільність, а занадто мале – уповільнити збіжність.

У підсумку градієнтний спуск дозволяє поступово налаштувати модель логістичної регресії для точного прогнозування класу спам / не спам на основі векторизованих текстових даних. Завдяки аналітичній простоті й математичній інтерпретованості цей підхід є придатним як для практичного використання, так і для глибокого контролю над навчанням моделі.

Оцінювання ефективності класифікаційної моделі є ключовим етапом, що дозволяє визначити її здатність узагальнювати закономірності та правильно розпізнавати об'єкти обох класів. У задачі виявлення спаму, де класи істотно нерівномірно представлені, особливо важливо враховувати не лише загальну точність, але й інші якісні характеристики моделі.

Основним інструментом оцінювання є матриця невідповідностей (confusion matrix), яка фіксує кількість правильно та неправильно класифікованих повідомлень кожного класу. У нашому випадку клас "спам" позначено як позитивний (positive class).

ISSN 2707-1758

Отже маємо:

- True Positives (TP) – спам-повідомлення, правильно класифіковані як спам;
- False Negatives (FN) – спам, помилково класифікований як не спам;
- False Positives (FP) – не спам, класифікований як спам;
- True Negatives (TN) – не спам, правильно класифікований як не спам.

На основі цієї матриці обчислюють основні метрики:

Accuracy – загальна частка правильних передбачень:

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP}. \quad (5)$$

Precision (точність) для класу "спам" показує, яку частку з усіх передбачених як спам повідомлень справді становить спам:

$$Precision = \frac{TP}{TP+FP}. \quad (6)$$

Recall (повнота) для класу "спам" визначає, яку частку зі справжніх спамів модель змогла правильно розпізнати:

$$Recall = \frac{TP}{TP+FN}. \quad (7)$$

F1-score є гармонічним середнім між точністю та повнотою:

$$F1 = 2\frac{Precision*Recall}{Precision+Recall}. \quad (8)$$

У підсумку багатовимірна система оцінювання дозволяє всебічно проаналізувати роботу класифікатора. Метрики precision, recall та f1-score дають можливість окремо оцінити здатність моделі виявляти спам та уникати хибних спрацьовувань для не спаму. Це має вирішальне значення для реальних сценаріїв автоматичного фільтрування електронної пошти, де помилки можуть призводити як до втрати важливих повідомлень, так і до пропуску небажаного контенту.

Результати

Для проведення експериментального дослідження використано датасет The Enron Email Dataset, доступний на платформі Kaggle (2025). Цей набір містить 5 157 електронних повідомлень, кожне з яких має текстову частину та мітку, що вказує на його належність до одного з двох класів: "ham" (звичайне повідомлення) або "spam" (небажане повідомлення).

Дані представлені у вигляді двох колонок:

- Message – текст електронного листа (англійською мовою).
- Category – мітка класу (ham або spam), що використовується як цільова змінна для класифікації.

Розподіл класів є нерівномірним:

- 87 % повідомлень належать до класу ham (не спам),
- 13 % – до класу spam.

Такий дисбаланс типово спостерігається в реальних електронних скриньках і створює додаткові складнощі для навчання моделей, оскільки алгоритми машинного навчання мають схильність до домінуючого класу. Саме тому під час оброблення даних було використано методи, спрямовані на компенсацію цього дисбалансу, зокрема і SMOTE.

Самі повідомлення у датасеті мають довжину від одного до кількох речень і містять як звичайну, так і рекламну лексику. Спам-повідомлення, зазвичай, включають фрази на кшталт "win", "free entry", "urgent", які характерні для фішингових або рекламних листів. Натомість повідомлення класу "ham" охоплюють щоденне листування, жарти, короткі особисті повідомлення тощо.

Повідомлення не мають структурованих полів, таких як тема чи адресат, і розглядаються виключно як неструктурований текст, що робить задачу лінгвістичною. Це дозволяє сфокусуватись на аналізі змісту повідомлень і якісному препроцесингу, без залежності від метаданих.

Отже, вибраний набір даних є репрезентативним прикладом задачі фільтрації електронної пошти на рівні змісту повідомлень і дозволяє моделювати класифікаційні алгоритми з орієнтацією на реальні сценарії застосування.

Розроблена модель логістичної регресії, навчена за допомогою градієнтного спуску, продемонструвала високі показники якості при класифікації електронних повідомлень на дві категорії: "спам" і "не спам" (рис. 3). Для оцінювання її ефективності використовували тестову вибірку, яка була попередньо відокремлена від навчальної на етапі препроцесингу. Таким способом, було забезпечено об'єктивність вимірювання результатів і виключено можливість витікання інформації з навчального етапу.

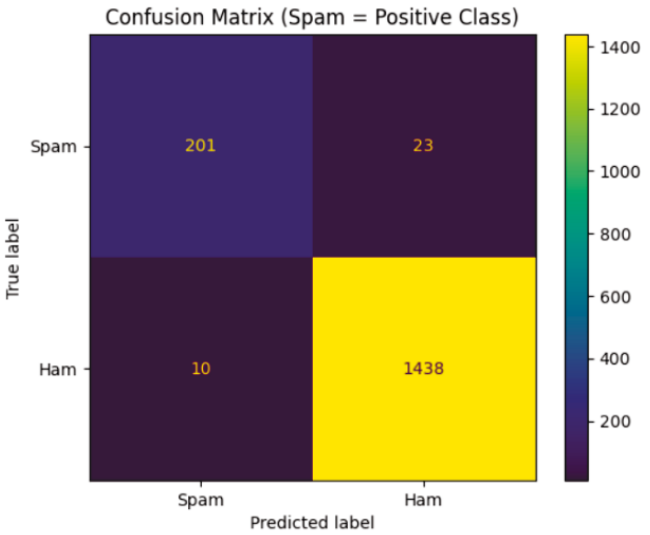


Рис. 3. Матриця невідповідностей для тестової вибірки (Spam – позитивний клас)

ISSN 2707-1758

Результати класифікації представлено у вигляді матриці невідповідностей (confusion matrix), яка відображає кількість правильно та неправильно класифікованих повідомлень обох класів (рис. 4):

- 201 спам-повідомлення були правильно класифіковані як спам (True Positives),
- 23 спам-листи були помилково розпізнані як не спам (False Negatives),
- 10 звичайних повідомлень були помилково віднесені до спаму (False Positives),
- 1438 не спамів були коректно класифіковані як не спам (True Negatives).

Загальна точність класифікації (accuracy) становила 98 %, що свідчить про те, що абсолютна більшість передбачень моделі були правильними. Проте під час оцінювання якості класифікації в умовах початкового дисбалансу класів важливо не обмежуватися лише загальною точністю. Саме тому проаналізовано глибші метрики: precision, recall і f1-score для кожного з класів окремо.

	precision	recall	f1-score	support
Ham	0.98	0.99	0.99	1448
Spam	0.95	0.90	0.92	224
accuracy			0.98	1672
macro avg	0.97	0.95	0.96	1672
weighted avg	0.98	0.98	0.98	1672

Рис. 4. Метрики оцінювання моделі логістичної регресії для тестової вибірки

Для класу "Ham", який є чисельнішим у датасеті, модель досягла таких результатів:

- Precision – 0.98: лише 2 % повідомлень, класифікованих як не спам, насправді були спамом.
- Recall – 0.99: з усіх реальних повідомлень, які не є спамом, 99 % були правильно виявлені.
- F1-score – 0.99: гармонійний баланс між точністю і повнотою.

Для класу "Spam", який є менш представленим:

- Precision – 0.95: з усіх передбачень класу спам, 95 % виявилися правильними.
- Recall – 0.90: модель виявила 90 % усіх реальних спамів у тестовій вибірці.
- F1-score – 0.92: значення, що підтверджує добре поєднання точності та виявлення спаму.

Macro average F1-score становить 0.96, що демонструє збалансовану продуктивність моделі для обох класів незалежно від їхньої частки в наборі даних. Weighted average F1-score дорівнює 0.98, що також підтверджує надійність моделі у практичному застосуванні.

Найбільший виклик традиційно полягає у виявленні меншості класу – спаму. Незважаючи на те, що Recall для нього дещо нижчий (0.90), модель продемонструвала здатність правильно класифікувати більшість спам-повідомлень і водночас не плутати їх із легітимними листами, що важливо для користувацького досвіду.

Отже, модель логістичної регресії на основі градієнтного спуску, у поєднанні з ретельним препроцесингом і методами балансування вибірки, продемонструвала високу ефективність у класифікації текстових електронних повідомлень, зокрема й у виявленні спаму в умовах обмежених і незбалансованих даних.

Дискусія і висновки

Результати дослідження підтверджують, що логістична регресія, реалізована з використанням градієнтного спуску, є ефективним і водночас інтерпретованим інструментом для класифікації текстових повідомлень на спам і не спам. Досягнута точність класифікації на рівні 98 % свідчить про високу здатність моделі до узагальнення навіть за умов обмеженого обсягу навчальних даних і наявного дисбалансу між класами.

Одним із ключових чинників, що вплинув на якість моделі, є етап підготовки текстових даних. Ретельне виконання препроцесингу – включаючи видалення пунктуації та чисел, фільтрацію стоп-слів, лематизацію та відсікання коротких токенів – дозволило перетворити необроблений текст на структуроване представлення з мінімумом шуму. Це створило необхідні передумови для ефективної векторизації та подальшого навчання.

Застосування TF-IDF векторизації забезпечило зважене врахування і частоти термінів, і їхньої релевантності в межах корпусу. Такий підхід дозволяє зосередитись на тих словах, що мають інформативну цінність, та зменшити вплив часто вживаних, але малозначущих слів. Важливою деталлю є використання L2-нормалізації, що вирівнює довжини векторів і сприяє стабільності та коректності сходження градієнтного алгоритму.

Оскільки початкові дані мали суттєвий дисбаланс між класами (87 % не спаму та 13 % спаму), особливу роль відіграло застосування методу SMOTE, що дозволив штучно збалансувати вибірку генерацією синтетичних прикладів меншості. Такий підхід дав змогу моделі побачити більше варіацій спам-повідомлень і підвищити recall без втрати точності.

Модель продемонструвала сильні показники як для переважаючого класу ("не спам"), так і для менш представленого класу ("спам"). Для класу спаму значення precision склало 0.95, recall – 0.90, а f1-score – 0.92, що свідчить про ефективність виявлення небажаних повідомлень при мінімальній кількості помилок. Для класу не спаму ці показники ще вищі: precision – 0.98, recall – 0.99, f1-score – 0.99. Макросереднє значення F1-метрики (0.96) і зважене середнє (0.98) підтверджують збалансованість моделі за всіма метриками, що важливо в умовах початкового перекосу у вибірці. Це означає, що модель не лише правильно розпізнає більшість повідомлень, але й зберігає стабільну якість на обох класах.

Загалом, результати підтверджують практичну придатність обраного підходу до автоматичної класифікації електронних листів. Поєднання логістичної регресії, градієнтного спуску, грамотно

ISSN 2707-1758

організованого препроцесингу й балансування вибірки дало змогу досягти високої точності при збереженні прозорості та керованості всієї моделі.

Внесок авторів: Віктор Морозов – концептуалізація; методологія; аналіз джерел, підготування огляду літератури або теоретичних засад дослідження; Владислав Дейнега – збір емпіричних даних та їхня валідація; емпіричне дослідження.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Jyothiikaa Moorthy. (2025). *23 Email Spam Statistics to Know in 2025.* https://www.mailmodo.com/guides/email-spam-statistics/

Kaggle, (2025). *The Enron Email Dataset.* https://www.kaggle.com/datasets/mohinurabdurahimova/maildataset

Khanday A., Shahbaz P., & Suraiya P. (2021). *Logistic Regression Based Classification of Spam and Non-Spam Emails.* https://doi.org/10.4108/eai.27-2-2020.2303291.

Mohammed, N., Mouhajir, M., & Yassine S.. (2023). *High Performance Computing Applied to Logistic Regression: A CPU and GPU Implementation Comparison.* https://www.researchgate.net/publication/373263539_High_Performance_Computing_Applied_to_Logistic_Regression_A_CPU_and_GPU_Implementation_Comparison

Papageorgiou, G., Economou, P., & Bersimis, S. (2024). A method for optimizing text preprocessing and text classification using multiple cycles of learning with an application on shipbrokers emails. *Journal of Applied Statistics*, 51(13), 2592–2626. https://doi.org/.10.1080/02664763./2024.2307535. PMID: 39290353; PMCID: PMC11404377.

Rakhmanov, O. (2020). A Comparative Study on Vectorization and Classification Techniques in Sentiment Analysis to Classify Student-Lecturer Comments. *Procedia Computer Science,* 178, 194–204. https://doi.org/10.1016/j.procs.2020.11.021

Spam Statistics 2025 (2025). *New Data on Junk Email, AI Scams & Phishing.* https://www.emailtooltester.com/en/blog/spam-statistics/

Zhang, L., Ray, H., Priestley, J., & Tan, S. (2019). A descriptive study of variable discretization and cost-sensitive logistic regression on imbalanced credit data. *Journal of Applied Statistics*, 47(3), 568–581. https://doi.org/10.1080/02664763.2019.1643829

References

Jyothiikaa Moorthy. (2025). *23 Email Spam Statistics to Know in 2025.* https://www.mailmodo.com/guides/email-spam-statistics/

Kaggle, (2025). *The Enron Email Dataset.* https://www.kaggle.com/datasets/mohinurabdurahimova/maildataset

Khanday A., Shahbaz P., & Suraiya P. (2021). *Logistic Regression Based Classification of Spam and Non-Spam Emails.* https://doi.org/10.4108/eai.27-2-2020.2303291.

Mohammed, N., Mouhajir, M., & Yassine S.. (2023). *High Performance Computing Applied to Logistic Regression: A CPU and GPU Implementation Comparison.* https://www.researchgate.net/publication/373263539_High_Performance_Computing_Applied_to_Logistic_Regression_A_CPU_and_GPU_Implementation_Comparison

Papageorgiou, G., Economou, P., & Bersimis, S. (2024). A method for optimizing text preprocessing and text classification using multiple cycles of learning with an application on shipbrokers emails. *Journal of Applied Statistics*, 51(13), 2592–2626. https://doi.org/.10.1080/02664763.2024.2307535. PMID: 39290353; PMCID: PMC11404377.

Rakhmanov, O. (2020). A Comparative Study on Vectorization and Classification Techniques in Sentiment Analysis to Classify Student-Lecturer Comments. *Procedia Computer Science,* 178, 194–204. https://doi.org/10.1016/j.procs.2020.11.021

Spam Statistics 2025 (2025). *New Data on Junk Email, AI Scams & Phishing.* https://www.emailtooltester.com/en/blog/spam-statistics/

Zhang, L., Ray, H., Priestley, J., & Tan, S. (2019). A descriptive study of variable discretization and cost-sensitive logistic regression on imbalanced credit data. *Journal of Applied Statistics*, 47(3), 568–581. https://doi.org/10.1080/02664763.2019.1643829

Отримано редакцією журналу / Received: 07.05.25
Прорецензовано / Revised: 16.05.25
Схвалено до друку / Accepted: 30.05.25

Viktor MOROZOV, PhD (Engin.), Prof.
ORCID ID: 0000-0001-7946-0832
e-mail: viktor.morozov@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Vladyslav DEINEHA, PhD Student
ORCID ID: 0009-0008-3123-2302
e-mail: thedynkan@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

SPAM DETECTION IN TEXT MESSAGES USING LOGISTIC REGRESSION BASED ON GRADIENT DESCENT

Background. ***With the increasing volume of email communication, the problem of spam filtering is becoming more and more relevant. According to statistical research, spam constitutes a significant portion of global email traffic, creating risks both for security and for the efficiency of electronic communication. In this context, natural language processing (NLP) and machine learning methods are gaining particular importance. The aim of this study is to develop a model for classifying email messages into spam and non-spam using logistic regression implemented via gradient descent, in combination with text data processing methods.***

Methods. ***The model was trained on a dataset containing over 5,000 email messages labeled as spam or non-spam. The data were preprocessed by removing noise components such as punctuation, numbers, stop words, and short tokens, followed by lemmatization. The cleaned texts were converted into numerical format using TF-IDF vectorization with L2 normalization. To address the class imbalance, the SMOTE method was applied. The model was trained using a classical gradient descent scheme with a sigmoid activation function and a logarithmic loss function.***

Results. ***The resulting model achieved high performance on the test set: overall accuracy was 98%, with an F1-score of 0.92 for the spam class and 0.99 for the non-spam class. The recall for spam reached 0.90, indicating the model's ability to detect most unwanted messages without excessive false positives. The balance between precision and recall is also reflected in macro and weighted average F1-scores, both exceeding 0.96.***

Conclusions. ***The findings demonstrate the effectiveness of combining logistic regression, gradient descent, and text preprocessing for the spam classification task, even in the presence of imbalanced data. The proposed approach is both efficient and interpretable, making it suitable for practical implementation in email filtering systems.***

Keywords: ***machine learning, natural language processing, gradient optimization, email filtering, text preprocessing, classification.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

УДК 621.398:004.032.26 + 004.056
DOI: https://doi.org/10.17721/ISTS.2025.9.81-92

Юлій БОЙКО, д-р техн. наук, проф.
ORCID ID: 0000-0003-0603-7827
e-mail: boiko_julius@ukr.net
Хмельницький національний університет, Хмельницький, Україна

Ілля ПЯТІН, канд. техн. наук, доц.
ORCID ID: 0000-0003-1898-6755
e-mail: ilkhmel@ukr.net
Хмельницький політехнічний фаховий коледж
Національного університету "Львівська політехніка", Хмельницький, Україна

Володимир ДРУЖИНІН, д-р техн. наук, проф.
ORCID ID: 0009-0009-5049-0099
e-mail: v_druzhinin@ukr.net
Київський національний університет імені Тараса Шевченка, Київ, Україна

Олександр ЄРЬОМЕНКО, канд. техн. наук, доц.
ORCID ID: 0000-0001-5110-3761
e-mail: yeromenko_s@ukr.net
Хмельницький національний університет, Хмельницький, Україна

ШВИДКЕ ПЕРЕТВОРЕННЯ ФУР'Є В OFDM: АЛГОРИТМІЧНІ ПІДХОДИ ТА ЇХНЯ РОЛЬ В ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ

Вступ. *Технологія мультиплексування з ортогональним частотним поділом каналів – OFDM (Orthogonal frequency-division multiplexing) є ключовою технологією сучасних інформаційних систем і використовується в мобільних мережах 4G, 5G, стандарті IEEE 802.11 (Wi-Fi) і цифровому телебаченні (DVB-T). Збільшення пропускної здатності каналів зв'язку вимагає оптимального вибору параметрів перетворень сигналів для ефективного використання апаратних ресурсів програмованої вентильної матриці FPGA (Field-programmable gate array) для реалізації OFDM.*

Методи. *Використано такі методи: математичне моделювання системи зв'язку з OFDM у середовищі Simulink, що дозволило дослідити функціональні перетворення сигналу, а також аналіз коефіцієнта бітових помилок для різних параметрів модуляції. Реалізацію алгоритмів FFT виконано з використанням HDL-кодування для порівняння ефективності алгоритмів швидкого перетворення Фур'є (FFT) Streaming Radix-2² і Burst Radix-2.*

Результати. *Результати моделювання показали, що використання передискретизації сигналу на передавачі покращує енергетику каналу, знижуючи необхідний рівень потужності на 12 дБ. Залежність коефіцієнта бітових помилок від відношення сигнал-шум демонструє, що збільшення довжини FFT з 512 до 2048 точок потребує підвищення відношення сигнал / шум SNR на 6 дБ. Аналіз впливу циклічного префікса (CP) показав, що оптимальна довжина CP становить 1/16 символу OFDM, що зменшує втрати швидкості передачі. Вплив модуляції на коефіцієнт бітових помилок (BER) свідчить про необхідність збільшення потужності під час переходу до вищих порядків квадратурної амплітудної модуляції (QAM).*

Висновки. *Зроблено висновок, що параметри FFT і передискретизації є критичними для ефективності системи OFDM. Отримані результати можуть бути використані для оптимізації реалізації OFDM на FPGA.*

Ключові слова: *ортогональне частотне мультиплексування (OFDM), швидке перетворення Фур'є (FFT), інформаційні системи, коефіцієнт бітових помилок, програмована вентильна матриця (FPGA).*

Вступ

Ортогональне частотне мультиплексування (OFDM) є ключовою технологією в сучасних системах зв'язку, зокрема у стандарті 5G NR (3GPP Release 15+), Wi-Fi 6 (IEEE 802.11ax) та LTE-A (3GPP Release 10+) (Kakkad et al., 2023). Завдяки ефективному використанню спектра та стійкості до багатопроменевого поширення сигналу, OFDM дозволяє суттєво підвищити швидкість передачі даних. Пропускна здатність системи збільшується за рахунок використання множини ортогональних піднесучих, які розташовані з мінімальними проміжками (на відміну від традиційних FDM-систем). Кожна піднесуча модулюється високоефективними схемами, такими як 16-QAM, 64-QAM або 256-QAM, залежно від умов каналу. Для мінімізації міжсимвольної інтерференції (ISI) та компенсації завмирань сигналу додається CP, який покращує стійкість до міжсимвольних спотворень, що особливо важливо в середовищах із відбиттям сигналу (напр., у міських умовах або промислових зонах). На передавальному боці здійснюється передискретизація сигналу та його передача через канал зв'язку, в якому можуть виникати затримки, частотні спотворення й адитивний білий гауссівський шум (AWGN). На приймальному боці реалізується оцінювання каналу (напр., за допомогою методів MMSE або LS estimation), вирівнювання частотної характеристики (equalization) і синхронізація. Демодуляція OFDM-сигналу включає зворотне швидке перетворення Фур'є (ЗШПФ=IFFT), усунення циклічного префікса (CP), зниження частоти дискретизації та демодуляцію QAM. На завершальному етапі проводиться аналіз BER, що дозволяє оцінити ефективність передачі даних (Бойко, & Новіков, 2021).

Завдяки можливості адаптації параметрів під конкретні умови каналу, OFDM залишається основою не лише для бездротових стандартів, а і для сучасних широкосмугових мереж зв'язку, включаючи OFDM-PON у пасивних оптичних мережах (PON) (Бойко, Єрьоменко, & Гур'єв, 2023) та когнітивних радіосистемах (CR). OFDM активно використовують в інформаційних технологіях для забезпечення високошвидкісного бездротового доступу до інтернету,

ISSN 2707-1758

зокрема в IoT, хмарних обчисленнях і системах відеостримінгу 4K/8K. Завдяки гнучкому розподілу спектра та підтримці MIMO-архітектури (Lalitha, & Kumar, 2024), ця технологія сприяє розвитку інтелектуальних мереж зв'язку, включаючи програмно-конфігуровані радіосистеми (SDR) і мережі 6G.

Метою статті є дослідження й оптимізація впливу параметрів перетворень сигналів на ефективність системи OFDM у контексті сучасних інформаційних технологій. Зокрема, розглядається реалізація алгоритмів FFT Streaming Radix-2² і Burst Radix-2 на FPGA, що є ключовим для підвищення продуктивності й оптимізації апаратних ресурсів. Проведений аналіз враховує використання OFDM у мобільних мережах 4G, 5G, стандарті IEEE 802.11 (Wi-Fi) і цифровому телебаченні (DVB-T), де вибір параметрів FFT, довжини CP та типу модуляції безпосередньо впливає на пропускну здатність і енергетичну ефективність системи.

***Огляд літератури*.** Основна концепція використання технології OFDM спрямована на підвищення швидкості передавання інформації (Alqahtani et al., 2024). Пропускна здатність каналу зв'язку безпосередньо залежить від кількості ініціалізованих піднесучих, а також від відстані між ними (Бойко та ін., 2022). При цьому кожна піднесуча звичайно модулюється методом квадратурної амплітудної модуляції (QAM). Зупинимось детальніше на результатах попередніх досліджень. Було досліджено, що одним із ключових способів мінімізації міжсимвольних завад та затухань у каналі є додавання CP до структури OFDM-сигналу (Zegrar, & Arslan, 2022). Ми акцентуємо, що для ефективного оброблення сигналу також застосовується процедура передискретизації. Згенерований та мультиплексований сигнал зазнає впливу завад, що зумовлює частотну залежність та виникнення затримки поширення. На боці приймача виконуються оцінка каналу, компенсація спотворень та синхронізація сигналу, як показано у (Kleider et al., 2005). Подальші операції включають демодуляцію OFDM, пониження частоти дискретизації та демодуляцію QAM-сигналів. Оцінка завадостійкості системи здійснюється шляхом аналізу коефіцієнта бітових помилок BER (Qiao et al., 2021). Усі зазначені аспекти детально висвітлено в низці наукових публікацій (Kishore et al., 2024; Shammaa, Mashaly, & El-mahdy, 2024).

У цій роботі ми представимо результати математичного моделювання OFDM-каналу, які доповнюють дослідження відомих авторів (Meenalakshmi, Chaturvedi, & Dwivedi, 2024) у контексті цифрової обробки сигналів. Особливу увагу приділено процедурі цифрового розбиття Digital Slice (DS) у перетвореннях Фур'є, що є важливим для мінімізації обчислювальної складності та підвищення пропускної здатності інформаційних систем. Робота також розглядає реалізацію алгоритмів FFT на рівні апаратного опису (Hardware Description Language, HDL) та архітектуру цифрових обчислювальних пристроїв для FPGA-застосувань (Shamani et al.). Представлено результати моделювання розробленої моделі в Simulink та досліджено показники завадостійкості.

Методи

Розглянемо систему зв'язку Simulink, модель якої зображено на рис. 1.

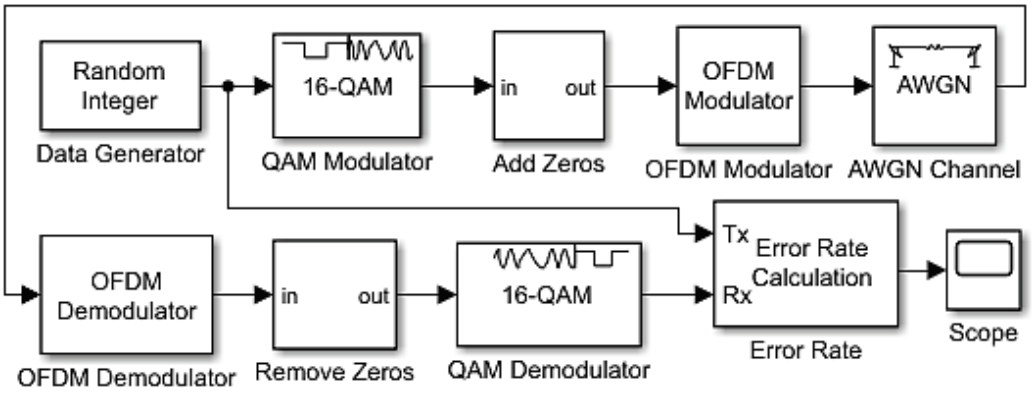


Рис. 1. Simulink-модель системи зв'язку з OFDM

Параметри моделі: модуляція 16-QAM; довжина ШПФ nFFT=128 точок; довжина циклічного префікса CPlen = 8; відстань між піднесучими scs = 20 кГц; частота дискретизації Fs = 1,28 МГц; тривалість відліку Ts = 7,8e-7 с; тривалість сигналу 128 відліків 1e-4 с.

Сигнал OFDM складається з великої кількості піднесучих, які разом утворюють спільну смугу частот. Швидкість передачі на одній піднесучій невелика, але об'єднання цих піднесучих в один потік даних суттєво підвищує швидкість передачі. Модуляція кожної піднесучої відбувається індивідуально. Сигнал на виході модулятора QAM може бути представлений виразом:

$$s(t)=\sum_{k} a(k)p(t-kT_s), \qquad (1)$$

де $a(k)$ – біти, що відповідають k-му символу, $p(t)$ – форма імпульсу, kT_s – тривалість імпульсу.

Передавач застосовує зворотне швидке перетворення Фур'є (IFFT) до символів. Вихід IFFT являє собою суму *N* ортогональних синусоїд:

$$x(t)=\sum_{k=0}^{N-1} X_k e^{j2\pi k\Delta ft}, \qquad (2)$$

де X_k – символи даних, *T* – тривалість символу OFDM. Символи даних X_k зазвичай комплексні й можуть бути з будь-якого сузір'я цифрової модуляції (напр., QPSK, 16-QAM, 64-QAM, ...).

Залежність коефіцієнта бітових помилок для різних видів модуляції зображено на рис. 2.

З отриманих залежностей можна зробити висновок, що перехід від модуляції QPSK до 256-QAM потребує збільшення енергії сигналу на 20 дБ.

ISSN 2707-1758

Пряме перетворення Фур'є та зворотне перетворення Фур'є для векторів *X* і *Y* визначають за виразами:

$$Y(k)=\sum_{j=1}^{n}X(j)W_n^{(j-1)(k-1)}, \quad (3)$$

$$X(j)=\frac{1}{n}\sum_{k=1}^{n}Y(k)W_n^{-(j-1)(k-1)}, \quad (4)$$

де $W_n=e^{-\frac{2\pi j}{n}}$ – корені одиничного числа.

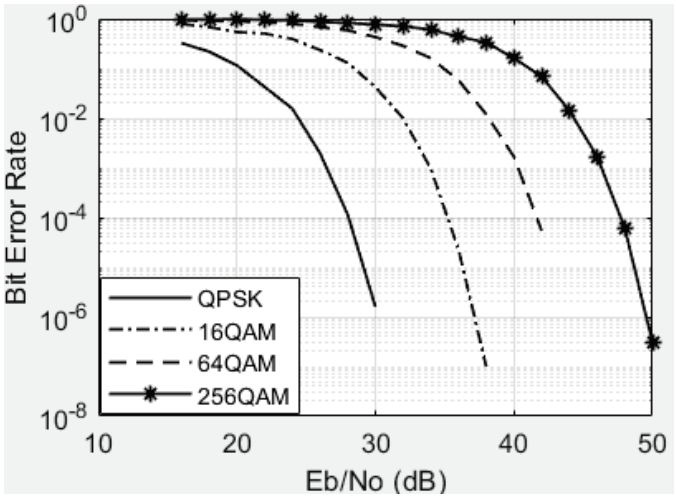


Рис. 2. Залежність коефіцієнта бітових помилок для різних видів модуляції

Використання OFDM із CP забезпечує вирівнювання та синхронізацію на основі FFT, що спрощує приймання порівняно з QAM для однієї несучої. Залежність коефіцієнта бітових помилок від коефіцієнта передискретизації сигналу наведено на рис. 3.

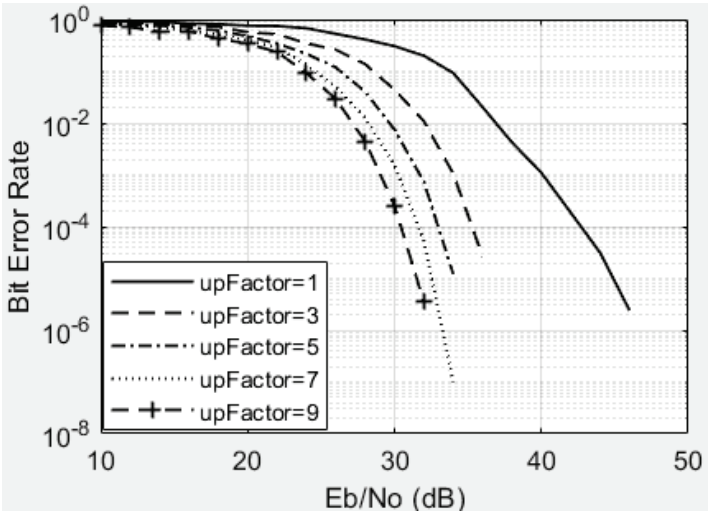


Рис. 3. Залежність коефіцієнта бітових помилок від коефіцієнта передискретизації сигналу

З отриманих залежностей можна зробити висновок, що підвищення частоти дискретизації на боці передавача покращує енергетику каналу зв'язку. Збільшення коефіцієнта передискретизації з 1 до 9 дозволяє понизити енергію сигналу на 12 дБ.

Передискретизація сигналу OFDM на боці передавача дозволяє краще боротися з пік-фактором на боці приймача. Крім того, передискретизація сприяє кращій апроксимації дискретного сигналу, зменшує обчислювальну складність зворотного перетворення Фур'є (IFFT). Залежність BER від SNR для різної кількості точок ШПФ наведено на рис. 4.

З отриманих залежностей можна зробити висновок, що збільшення довжини ШПФ із 512 до 2048 точок потребує підвищення відношення сигнал-шум на 6 дБ. Збільшення довжини ШПФ розширює смугу пропускання системи зв'язку, і для покращення енергетичних характеристик потребує використання завадостійких кодів, які дозволяють коригувати помилки, що виникають під час передачі. Найпопулярнішими кодами в сучасних інформаційних мережах є коди LDPC (Low-Density Parity-Check) (Бойко, Єрьоменко, & Гур'єв, 2023), турбокоди (Zhurakovskiy et al., 2023) та polar-коди (Meenalakshmi, Chaturvedi, & Dwivedi, 2024). Вони забезпечують високу ефективність у боротьбі з помилками при малих значеннях сигнал / шум (SNR). Оскільки OFDM є дуже чутливою до багатопроменевих ефектів і інтерференцій,

ISSN 2707-1758

використання таких кодів дозволяє зберігати стабільність передачі навіть у складних умовах каналу. Специфіка OFDM із кодами полягає в тому, що розділення сигналів по частотних каналах дає можливість застосовувати адаптивне кодування та модуляцію для кожного підканалу, що забезпечує максимальну ефективність використання спектра і високу завадостійкість.

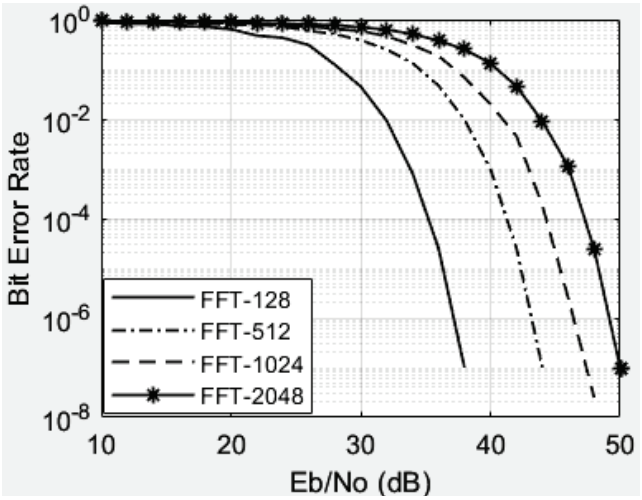


Рис. 4. Залежність BER від SNR для різної кількості точок ШПФ

Для розглянутої моделі частота дискретизації пов'язана з інтервалом між піднесучими та довжиною ШПФ. Довжина циклічного префікса має бути більшою за тривалість затримки у багатопроменевому каналі (рис. 5).

Для уникнення ISI, тривалість циклічного префікса має перевищувати тривалість багатопроменевої затримки. В межах представленої роботи нами проведене дослідження CP різної довжини: 1/2, 1/4, 1/8 і 1/16 тривалості символу OFDM. Установлено, що у випадку, якщо довжина циклічного префікса дорівнює 1/4 тривалості символу, швидкість передачі падає на 25 %, але це збільшує тривалість багатопроменевої затримки сигналу. Найчастіше використовують тривалість циклічного префікса, яка рівна 1/16 тривалості символу, що зменшує швидкість передачі на 6 %.

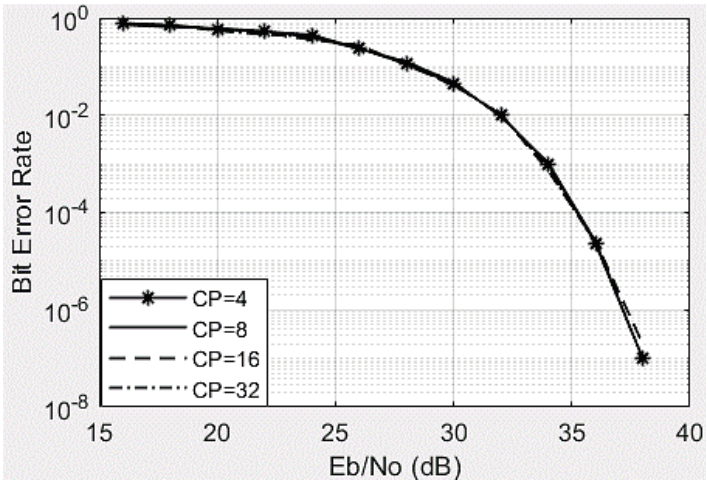


Рис. 5. Залежність BER від довжини циклічного префікса

Імпульсна характеристика каналу зв'язку з багатопроменевим завмиранням визначається виразом

$$h(t,\tau)=\sum_{q=0}^{Q-1}h_q(t)\delta\left(\tau-\tau_q\right), \qquad (5)$$

де Q – кількість затриманих складових сигналу; h_q – коефіцієнт ослаблення та фазового зсуву для q-ї складової, τ – затримка. Такий канал характеризується складним випадковим процесом, що включає компоненти з різними допплерівськими зсувами. Допплерівські ефекти порушують ортогональність між піднесучими частотами в OFDM, що може спричиняти міжканальні завади (ICI) і деградацію прийому.

Прийнятий сигнал $Y(f)$ представимо згорткою переданого сигналу $U(f)$ з імпульсною характеристикою каналу $H(f)$:

$$Y(f)=H(f)\cdot U(f). \qquad (6)$$

Приймачі OFDM використовують вирівнювання прийнятого сигналу. Циклічний префікс дозволяє ефективно застосовувати OFDM у неідеальному каналі (рис. 6) із невідомою затримкою розповсюдження.

ISSN 2707-1758

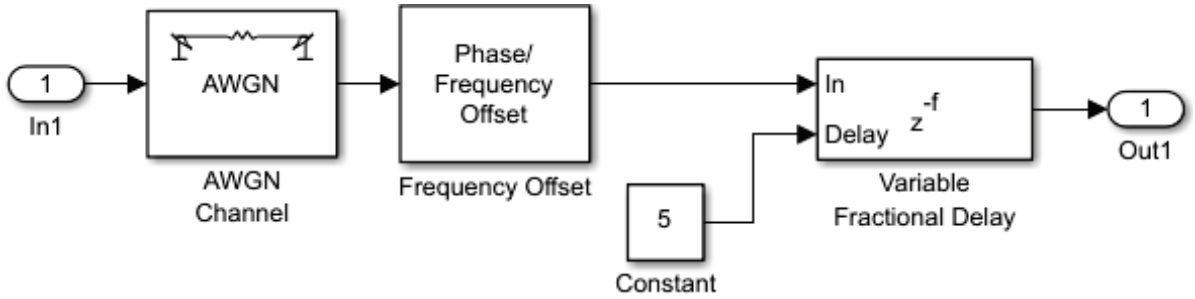
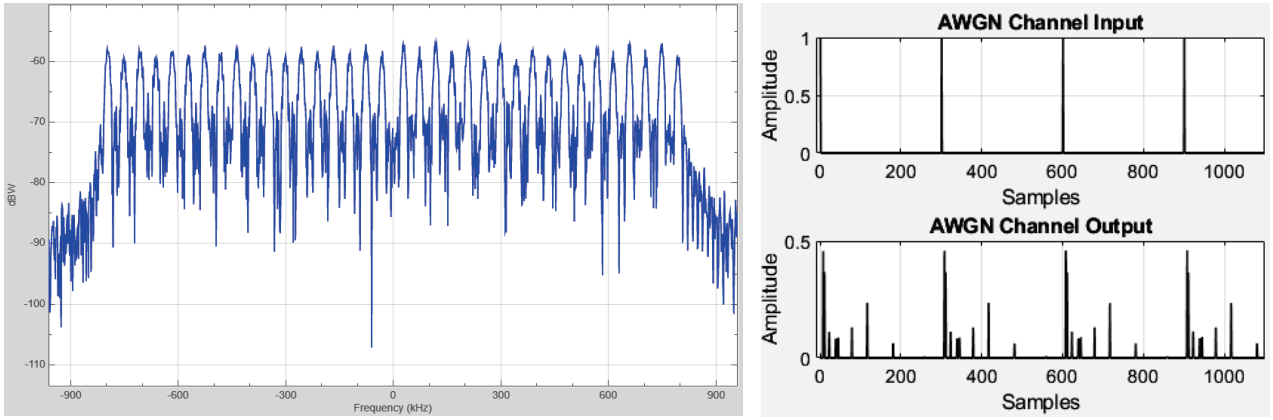


Рис. 6. Simulink-модель каналу зв'язку

Модель каналу зв'язку включає: вплив адитивного білого гауссівського шуму (AWGN Channel); частотного і фазового зсуву сигналу внаслідок ефекту Допплера; дробової затримки в каналі зв'язку; завмирань сигналу. Відстань між імпульсами має бути більшою, ніж максимально можлива затримка каналу. Результати дослідження імпульсної характеристики каналу наведено на рис. 7а. На рис. 7б подано спектр сигналу на вході приймача.



а б

Рис. 7. Характеристики каналу: а – імпульсні; б – спектральна

Спектр на рис. 7б побудовано для сигналу, що містить 128 піднесучих, частота дискретизації становить 1,92 МГц, довжина циклічного префікса – 9 відліків. Метод оцінювання спектра – метод періодограми Уелша (Welch's periodogram method).

Побудова систем зв'язку на базі FPGA ефективна через можливість паралельного оброблення даних (Algnabi et al., 2018). Операція множення призводить до великої затримки у часі, тому використаємо методику реалізації процесора ШПФ без помножувача за технікою цифрового зрізу.

Ідея цифрових зрізів полягає в тому, що будь-яке комплексне число *F* можна розбити на дрібніші блоки, кожен з яких має коротшу довжину слова *p*, як показано в таких рівняннях (для додаткового коду – two's complement):

$$F = \sum_{k=0}^{b-1} \left(2^{p-1}\right)^k FR_k + j \sum_{k=0}^{b-1} \left(2^{p-1}\right)^k FI_k , \qquad (7)$$

$$FR_k = -\left(2^{p-1}\right) FR_{(p-1),i} + \sum_{i=0}^{p-2} \left(2^i\right) FR_{k,i} , \qquad (8)$$

$$FI_k = -\left(2^{p-1}\right) FI_{(p-1),i} + \sum_{i=0}^{p-2} \left(2^i\right) FI_{k,i} , \qquad (9)$$

де $FI_{k,i}$ та $FR_{k,i}$ набувають значення нуль або одиниця.

Алгоритм Radix 2^2 SDF DIF FFT будують на основі виразу

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} , \qquad (10)$$

де $x[n]$, $X[k]$ – комплексні числа; $W_N^{nk} = e^{-j2\pi/N}$ – фазовий множник; інші параметри мають відповідати умовам:

$$n \le \frac{N}{2} n_1 + \frac{N}{4} n_2 + n_3 > N ;$$

$$k \le k_1 + 2k_2 + 4k_3 > N .$$

і не повинні перевищувати довжину ШПФ.

На кожному кроці алгоритму довжина ШПФ ділиться навпіл. На рис. 8 показано графік потоку сигналів у вигляді метелика для алгоритмів ШПФ Radix 2^2.

На рис. 9 показано алгоритм Radix 2^2 для 8-точкового ШПФ.

Кожний етап містить два метелика з однонаправленим зворотним зв'язком із затримкою (SDF) та контролером пам'яті. Перша стадія SDF – звичайний метелик. Другий етап множить вихідні дані першого етапу на $-j$. Щоб уникнути апаратного помножувача, блок змінює місцями дійсну й уявну частини входів і виходів. На кожному етапі відбувається множення результату на фазовий множник.

ISSN 2707-1758

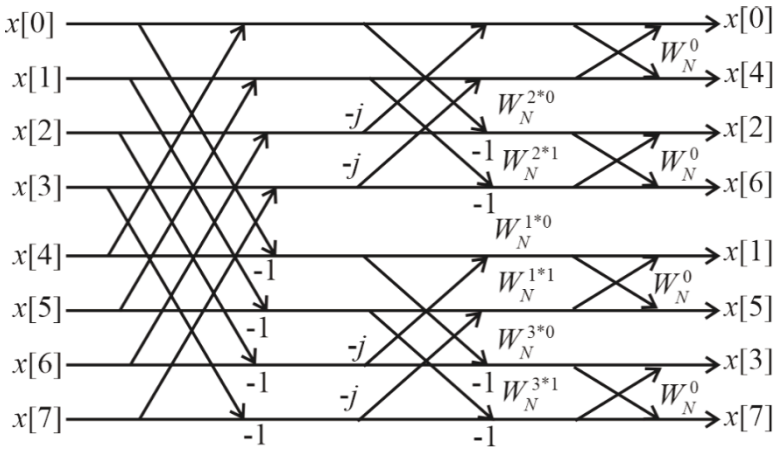


Рис. 8. Граф потоку сигналів для 8-точкового Radix 2^2 DIF FFT

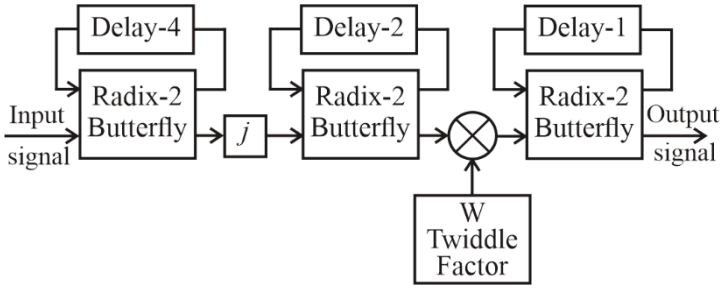


Рис. 9. Структура 8-точкового Radix 2^2 SDF DIF FFT

У найпростішому випадку комплексний помножувач може бути реалізований за допомогою чотирьох помножувачів для дійсних сигналів, одного підсумовувача й одного віднімача, як показано на рис. 10, згідно з таким виразом:

$$(a_r + ja_i)(b_r + jb_i) = (a_r b_r - a_i b_i) + j(a_i b_r + a_r b_i)\,. \tag{11}$$

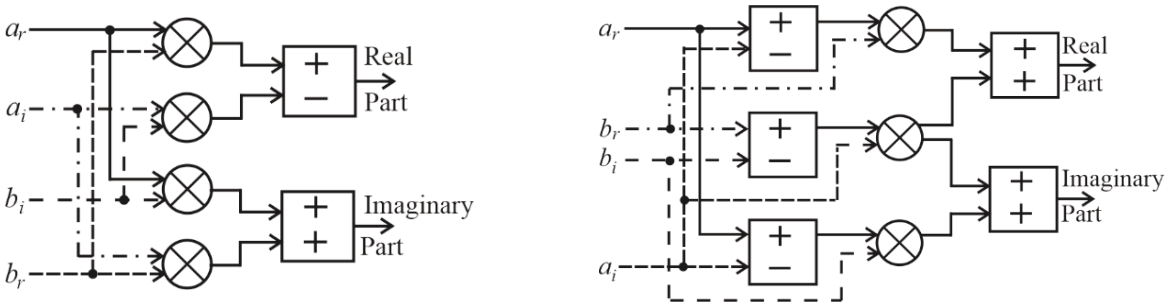


Рис. 10. Приклади реалізації комплексного помножувача:
а – із чотирма реальними помножувачами і двома підсумовувачами;
б – із трьома реальними помножувачами і 5 підсумовувачами

Зазначимо, що схема комплексного помножувача, представлена на рис. 10а, займає велику площу кристала у реалізації FPGA. Комплексний помножувач також може бути реалізований за допомогою трьох помножувачів для дійсного сигналу та п'яти суматорів або віднімачів, як показано на рис. 11, на основі такого рівняння:

$$(a_r + ja_j)(a_r + ja_j) = \{b_r(a_r - a_i) + a_i(b_r - b_i)\} + \\ + j\{b_i(a_r + a_i) + a_i(b_r - b_i)\}. \tag{12}$$

Вказана структура комплексного помножувача займає менше місця на кристалі FPGA.

Розглянемо шляхи застосування методу цифрового зрізу для побудови компонента метелика R2^2 SDF, щоб зменшити складність обчислень і підвищити пропускну здатність.

Операція множення двох чисел додає затримку перетворення і займає багато ресурсів FPGA. В апаратному забезпеченні використано цифровий зріз без множника (digit slicing multiplier-less) для підвищення швидкодії операції множення. Операція множення замінюється операціями зсуву і додавання. Концепція формування архітектури цифрового зрізу ґрунтується на тому, що будь-яке двійкове число

ISSN 2707-1758

можна розбити на кілька блоків коротших двійкових чисел, причому кожен блок має різну вагу.

Для представлення зрізу даних у додатковому коді, використовуємо вираз (13):

$$x = \left[\sum_{k=0}^{b-1} 2^{pk} X_k\right] 2^{-(pb-1)}, \quad X_k = \sum_{j=0}^{p-1} 2^j X_{k,j}, \qquad (13)$$

де x розбито на b блоків, p – бітова ширина блоку, а $X_{k,j}$ набуває значення 0 або 1, за винятком $X_k = b-1$, $j = p-1$, який може дорівнювати 0 або –1. Архітектура цифрового зрізу (Digital Slice) була застосована до вхідних даних комплексного помножувача для розподілу їх на дві групи (для 8-точкового ШПФ), кожна з яких містить чотири біти, як показано на рис. 11.

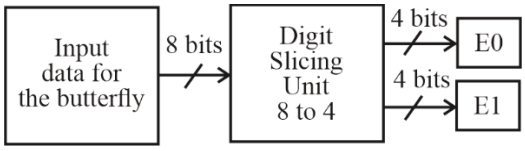


Рис. 11. Структура цифрового зрізу

Комплексний помножувач реалізовано за схемою рис. 10б, а для вхідних даних реального помножувача застосовано цифровий зріз, щоб зробити процес множення паралельним. Тобто час оброблення буде зменшено.

Оскільки фазові множники ШПФ заздалегідь відомі (див. рис. 8), можливі варіанти множення для 8-бітного коефіцієнта обертання та множення на 4-бітні вхідні дані можуть бути збережені в одному ОЗП для кожного коефіцієнта обертання. Це приведе до скорочення часу перетворення та зменшення площі кристала, яку займає цифровий зріз (digit slicing).

Конструкція цифрового зрізу помножувача складається з таблиці істинності (lookup table – ROM), зсувача та підсумовувача.

Результати

Для HDL (Hardware Description Language) реалізації FFT (Fast Fourier Transform) алгоритмів (Ayhan, Dehaene, & Verhelst, 2014) необхідно організувати інтерфейс потокових даних, апаратну затримку та сигнали управління функціональними блоками. Проведено дослідження двох архітектур ШПФ (рис. 12а, 12б):

1) Streaming Radix 2^2 – потокова архітектура, призначена для підтримки схем із високою пропускною здатністю, вона використовується для програм з обмеженими ресурсами FPGA, особливо за великої довжини ШПФ.

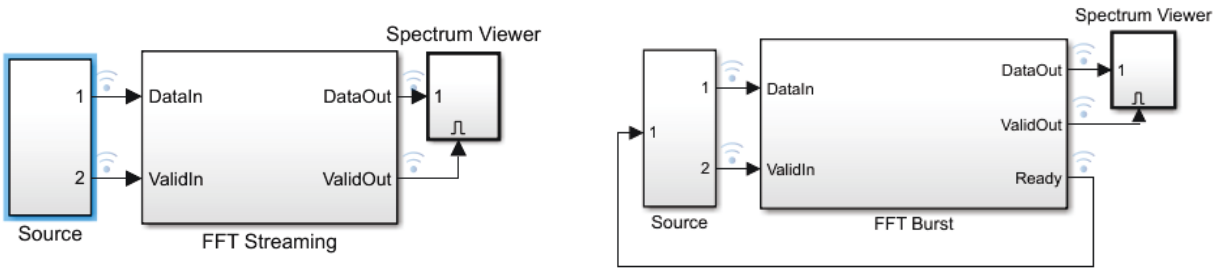
Burst Radix-2 – пакетна архітектура, призначена для схем із малою площею.

Одним із способів підвищення швидкості оброблення інформації на FPGA є паралельне оброблення кількох вибірок у схемах з нижчою тактовою частотою. Багато сучасних FPGA підтримують стандартний інтерфейс JESD204B, який приймає вхідні скалярні дані з високою тактовою частотою та генерує вектор вибірок із нижчою тактовою частотою.

2) Для архітектури Streaming Radix 2^2 вхідними даними є три синусоїдальні хвилі із частотами 200 кГц, 230 кГц і 260 кГц за частоти дискретизації 2 МГц. Розмір вхідного вектора становить 8 відліків. Застосовано контроль правильності вхідного сигналу на кожному другому циклі оброблення. Спектр вихідного сигналу показано на рис. 13.

Контроль рівнів сигналів на вході і виході схеми здійснюють за допомогою додатка Logic Analyzer у середовищі Simulink, як зображено на рис. 14.

Logic Analyzer дозволяє переглядати вхідні та вихідні сигнали підсистеми FFT Streaming. Форма хвилі показує, що допустимий вхідний сигнал має високий рівень на кожному другому циклі і що існує деяка затримка, перш ніж блок поверне першу допустиму вихідну вибірку.



а б

Рис. 12. Simulink-модель дослідження: а – Streaming Radix 2^2; б – Burst Radix 2

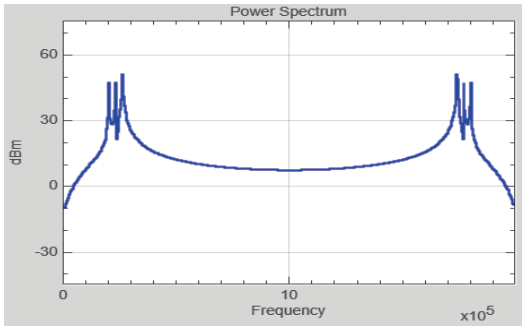


Рис. 13. Спектр вихідного сигналу для архітектури Streaming Radix 2^2

ISSN 2707-1758

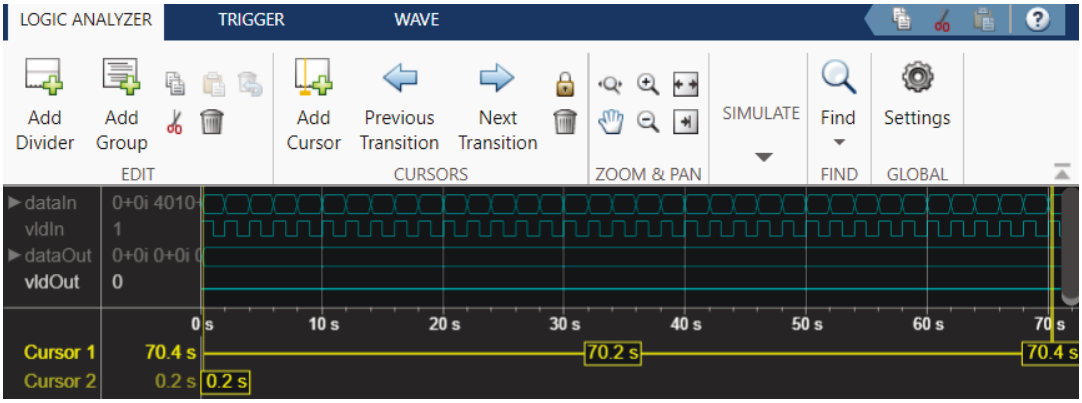


Рис. 14. Дослідження сигналів у додатку Logic Analyzer для архітектури Streaming Radix 2^2

Архітектуру Burst Radix 2 (рис. 15) використовують для додатків з обмеженими ресурсами FPGA, особливо для великої довжини ШПФ (Kumar, Selvakumar, & Sobha, 2015). Сигнал готовності встановлюється в 1 (істина), коли блок може приймати дані, і починає оброблення після того, як весь фрейм ШПФ буде збережено у пам'яті. Блок ігнорує сигнал на вході, поки прапорець готовності не дорівнює логічному нулю.

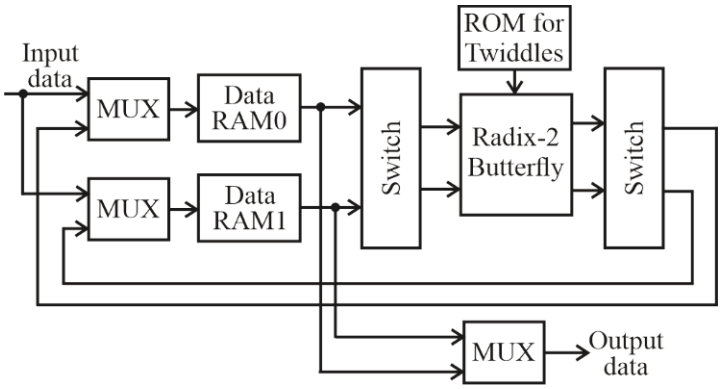


Рис. 15. Архітектура Burst Radix 2

Вхідні дані зберігаються послідовно в RAM0 і RAM1 (рис. 15). Перші чотири точки даних зберігаються в RAM0, а останні чотири – у RAM1. Схеми адресації можна виразити як

$$RA_0 = R_4\left[0\,1\,2\,3\right]^T,$$

$$RA_{i+1} = L_2^4 RA_i,\ 0 \le i \le 2,$$

$$RB_0 = R_4\left[0\,1\,2\,3\right]^T,$$

$$RB_{i+1} = \left(J_2 \otimes I_2\right) L_2^4 RB_i,\ 0 \le i \le 2,$$

де RA_i – схема адресації для RAM0, RB_i – схема адресації для RAM1, i – ідентифікатор етапу, J_2 – матриця обміну даними.

Операцію Radix-2 Butterfly представимо таким виразом:

$$BF_{out} = \left[\left(\left(\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & \omega_8\end{bmatrix}^p \odot \begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\right) BF_{in}\right], \quad (14)$$

де BF_{out} і BF_{in} – вихід і вхід метелика за основою 2 на рис. 15 відповідно; $\omega_8 = e^{-j\frac{2\pi}{8}}$:

$$p = 8 \cdot 2^{-(i+1)} \left\lfloor \frac{2^{(i+1)} b}{8} \right\rfloor. \quad (15)$$

На рис. 16 наведено спектр сигналу на виході архітектури Burst Radix 2.

Порівняння рис. 13 і рис. 16 в межах нашого дослідження дозволяє зробити важливий висновок, що архітектура Streaming Radix 2^2 забезпечує чистіший спектр сигналу без виражених бічних складових, що є суттєвим фактором для багатьох застосувань, де важлива висока якість сигналу, наприклад, у комунікаційних системах і обробленні радіосигналів. Це дозволяє досягти кращих результатів у таких задачах, як розпізнавання сигналів або відновлення сигналу у зашумленому середовищі. Відсутність бічних складових також покращує ефективність використання спектра, що є важливим у випадках з обмеженими спектральними ресурсами. Вікно Logic Analyzer для архітектури Burst Radix 2, що показано на рис. 17, дає змогу отримати візуальне уявлення про часову поведінку сигналів, а це дозволяє точно оцінити характер їхнього оброблення.

Logic Analyzer є важливим інструментом для перегляду вхідних і вихідних сигналів підсистеми FFT Burst. Осцилограма, яку можна побачити у вікні Logic Analyzer, показує, що вхідні дані надходять у вигляді пачок допустимих вибірок, а також свідчить про наявність затримки перед тим, як блок поверне допустимі вихідні вибірки. Це може бути важливим аспектом для проєктувальників, які намагаються знизити затримки в обробленні даних для забезпечення ефективнішого реального часу в системах, таких як системи оброблення сигналів і телекомунікацій.

ISSN 2707-1758

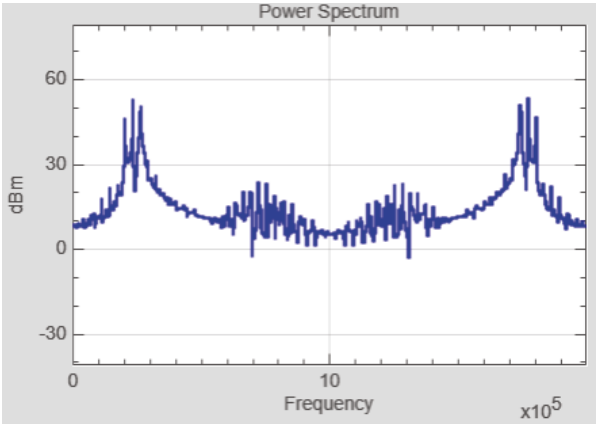


Рис. 16. Спектр сигналу на виході архітектури Burst Radix 2

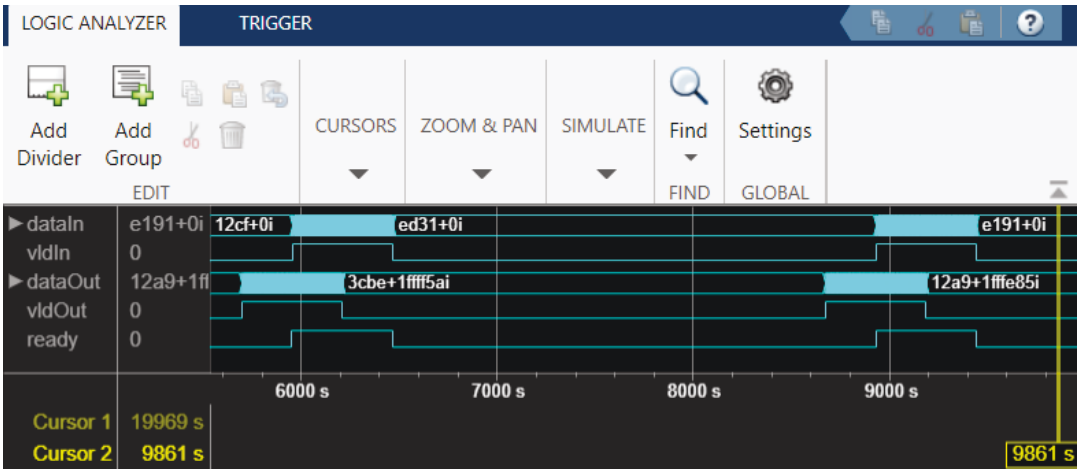


Рис. 17. Дослідження сигналів у додатку Logic Analyzer для архітектури Burst Radix 2

На рис. 18 представлено оцінку затримки вихідного сигналу для різних архітектур ШПФ.

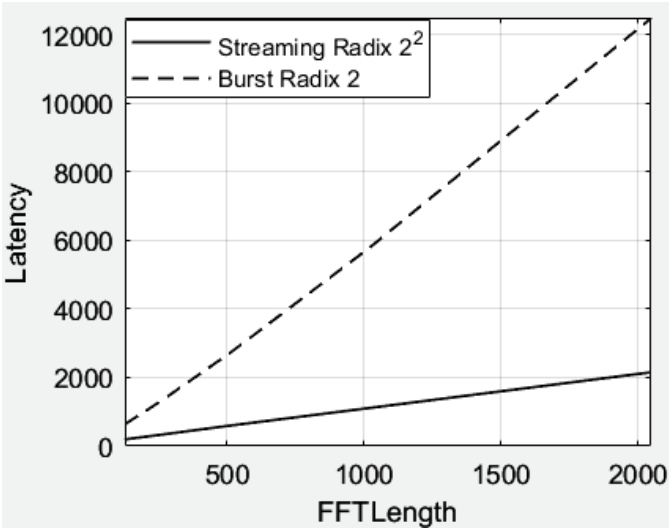


Рис. 18. Затримка вихідного сигналу для різних архітектур ШПФ

Результати дослідження затримки для різних архітектур показано на рис. 18, що дає змогу чітко порівняти ефективність кожної з архітектур. Згідно з рис. 18 можна зробити висновок, що архітектура Burst Radix 2 дає більшу затримку порівняно з архітектурою Streaming Radix 2² (Srinivasa, Madhumati, & Sailaja, 2024). Це свідчить про те, що для застосувань, де критично важлива мінімальна затримка, архітектура

ISSN 2707-1758

Streaming Radix 2² є оптимальнішим варіантом. Причому різниця у затримці збільшується із зростанням довжини ШПФ, що підкреслює важливість вибору правильної архітектури залежно від конкретних вимог до системи. Такий аналіз може бути корисним під час вибору архітектури для оброблення сигналів у реальному часі, особливо в таких сферах, як радіозв'язок і цифрове оброблення сигналів у складних умовах, де затримка є критичним параметром.

Дискусія і висновки

У межах проведеного дослідження системи зв'язку з OFDM у середовищі Simulink проаналізовано кілька ключових аспектів, що впливають на енергетичні характеристики й ефективність зв'язку. Результати показали, що перехід від модуляції QPSK до 256-QAM потребує значного збільшення енергії сигналу, що складає близько 20 дБ. Це підтверджує важливість вибору оптимальної модуляції залежно від умов каналу і вимог до якості зв'язку. З іншого боку, підвищення частоти дискретизації на боці передавача значно покращує енергетику каналу зв'язку, що критично важливо для досягнення високої пропускної здатності. Дослідження також показало, що збільшення коефіцієнта передискретизації з 1 до 9 дозволяє знизити енергію сигналу на 12 дБ. Це свідчить про важливість правильного налаштування параметрів передискретизації для досягнення оптимального балансу між енергетичними характеристиками і складністю системи. Збільшення довжини ШПФ з 512 до 2048 точок потребує підвищення відношення сигнал-шум (SNR) на 6 дБ. Цей результат підтверджує, що більша довжина ШПФ дозволяє покращити спектральні характеристики, однак вимагає більшої потужності та ресурсів для оброблення сигналів. Установлено, що збільшення довжини ШПФ розширює смугу пропускання системи зв'язку, що також вимагає використання завадостійких кодів для забезпечення високої якості зв'язку в умовах шуму. Це підкреслює важливість інтеграції ефективних кодів корекції помилок у сучасні телекомунікаційні системи, щоб забезпечити стабільність зв'язку на великих відстанях.

Що стосується CP, то проведене дослідження різної довжини CP (1/2, 1/4, 1/8 та 1/16 від довжини символу OFDM) показало, що для оптимального уникнення інтерференції між символами, тривалість циклічного префікса має бути більшою за тривалість багатопроменевої затримки. Зокрема, використання CP завдовжки 1/4 символу призводить до зменшення швидкості передачі на 25 %, хоча цей підхід ефективно компенсує багатопроменеві затримки. Найпоширенішим варіантом є CP завдовжки 1/16 символу, що дає зниження швидкості передачі лише на 6 %, і це є оптимальним для багатьох сучасних застосувань (Liu, & He, 2024).

Важливим етапом дослідження було визначення імпульсної характеристики каналу, що має критичне значення на етапі вирівнювання каналу на приймачі. Це дозволяє зменшити ефекти, пов'язані з багатопроменевістю та затримками, що важливо для досягнення стабільного з'єднання, особливо у складних умовах міського середовища чи за наявності завад.

Щодо HDL реалізації 8-точкових FFT алгоритмів Streaming Radix 2² та Burst Radix-2, то проведене дослідження показало, що архітектура Streaming Radix 2² забезпечує чистіший спектр сигналу, позбавлений бічних складових, що є суттєвим для застосувань, де необхідна висока якість сигналу, таких як радіосигнал або високочастотна комунікація. Водночас архітектура Burst Radix 2, хоч і має вищу затримку порівняно із Streaming Radix 2², може бути корисною для застосувань, де важлива висока швидкість оброблення даних і де менші вимоги до затримки в каналі. Зазначимо, що різниця у затримці між цими архітектурами зростає зі збільшенням довжини ШПФ, що робить вибір архітектури залежним від конкретних вимог до системи. Загалом, результати дослідження підтверджують важливість правильного вибору параметрів модуляції, довжини ШПФ, циклічного префікса і архітектури оброблення сигналу для досягнення оптимальних характеристик системи зв'язку. Врахування цих факторів дозволяє досягти ефективнішого використання енергетичних ресурсів, зменшити затримки і підвищити загальну продуктивність системи в умовах реального середовища.

Отримані результати мають велике значення для розвитку сучасних інформаційних систем, особливо в контексті телекомунікацій і бездротових мереж. Оптимізація параметрів модуляції, довжини ШПФ і циклічного префікса є важливою для підвищення енергетичної ефективності та швидкості передачі даних, що має безпосередній вплив на стабільність і якість зв'язку в реальних умовах (Бойко, Єрьоменко, & Пятін, 2024). Використання завадостійких кодів і вибір оптимальних архітектур оброблення сигналів дозволяє забезпечити високу пропускну здатність навіть в умовах шуму та багатопроменевості каналу. Ці результати важливі для розроблення більш ефективних і надійних інформаційних систем, зокрема і для 5G та майбутніх поколінь мобільних мереж, що вимагають високої продуктивності та стійкості до завадових впливів.

Внесок авторів: Юлій Бойко, Ілля Пятін – концептуалізація; методологія; аналіз джерел, підготування огляду літератури або теоретичних засад дослідження; Володимир Дружинін, Олександр Єрьоменко – збір емпіричних даних та їх валідація; емпіричне дослідження.

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Бойко, Ю., Єрьоменко, О., & Гур'єв, О. (2023). Можливості LDPC-кодів у підвищенні продуктивності оптичних телекомунікацій з OFDM. *Measuring and computing devices in technological processes*, 4, 13–26. https://doi.org/10.31891/2219-9365-2023-76-2

Бойко, Ю., Єрьоменко, О., & Пятін, І. (2024). Оцінка частотно-фазових спотворень в оптичних телекомунікаціях з OFDM. *Інфокомунікаційні технології та електронна інженерія*, 4(2), 130–140. https://doi.org/10.23939/ictee2024.02.130

Бойко, Ю., Пятін, І., Єрьоменко, О., & Шаюк, Д. (2022). Оцінювання впливу зміщення несучих частот на завадостійкість телекомунікацій з OFDM. *Measuring and computing devices in technological processes*, 3, 19–26. https://doi.org/10.31891/2219-9365-2022-71-3-3

Бойко Ю. М., & Новіков Д. В. (2021). Оцінка ефективності канального кодування у телекомунікаціях з OFDM. *Herald of Khmelnytskyi National University. Technical sciences*, 5(301), 150–159. https://www.doi.org/10.31891/2307-5732-2021-301-5-150-159

Algnabi, Y. S., Teymourzadeh, R., Othman, M., & Islam, M. S. (2018). FPGA Implementation of pipeline Digit-Slicing Multiplier-Less Radix 2 power of 2 DIF SDF Butterfly for Fourier Transform Structure. *Signal Processing*. https://doi.org/10.48550/arXiv.1806.04570

Alqahtani, A. S., Pandiaraj, S., Alshmrany, S. Ali Jaber A., Sandeep P. & U. Arun K. **(**2024). Enhancing MIMO-OFDM channel estimation in 5G and beyond with conditional self-attention generative adversarial networks. *Wireless Networks*, 30, 1719–1736. https://doi.org/10.1007/s11276-023-03615-y

Ayhan, T., Dehaene, W., & Verhelst, M. (2014). A 128:2048/1536 Point FFT Hardware Implementation with Output Pruning. In *European Signal Processing Conference (EUSIPCO)* (pp. 1–5). Isola delle Femmine – Palermo. https://www.eurasip.org/Proceedings/Eusipco/Eusipco2014/HTML/papers/1569925329.pdf

Kakkad, Y., Patel, D. K., Kavaiya, S., Sun, S. & López-Benítez, M. (2023). Optimal 3GPP Fairness Parameters in 5G NR Unlicensed (NR-U) and WiFi Coexistence. *IEEE Transactions on Vehicular Technology*, 72(4), 5373–5377. https://ieeexplore.ieee.org/document/9954188

ISSN 2707-1758

Kleider, J. E., Maalouli, G., Gifford, S., & Chuprun, S. (2005). Preamble and embedded synchronization for RF carrier frequency-hopped OFDM. *IEEE Journal on Selected Areas in Communications*, 23(5), 920–931. https://ieeexplore.ieee.org/document/1425638

Kishore, K. K., Suman, J. V., Mallam, M., Hema, M., & Guntreddi, V. (2024). Scrambled UFMC and OFDM Techniques With APSK Modulation in 5G Networks Using Particle Swarm Optimization. *IEEE Access*, 12, 104091–104101. https://doi.org/10.1109/ACCESS.2024.3421311

Kumar, V. M., Selvakumar, D. A., & Sobha, P. M. (2015). Area and frequency optimized 1024 point Radix-2 FFT processor on FPGA. In *IEEE International Conference on VLSI-SATA* (pp. 1–6). Bengaluru, India. https://ieeexplore.ieee.org/document/7050487

Lalitha, H. & Kumar, N. (2024). Carrier frequency synchronization for WLAN systems based on MIMO-OFDM-IM. *EURASIP Journal on Wireless Communications and Networking*, 77, (2024). https://doi.org/10.1186/s13638-024-02406-z

Liu, J. & He, J. (2024). Design and analysis of CP-free OFDM PDMA transmission system. *EURASIP Journal on Advances in Signal Processing*, 68, (2024). https://doi.org/10.1186/s 13634-024-01154-y

Meenalakshmi, M., Chaturvedi, S. & Dwivedi, V. K. (2024). Enhancing channel estimation accuracy in polar-coded MIMO–OFDM systems via CNN with 5G channel models. *AEU - International Journal of Electronics and Communications*, 173, 155016. https://doi.org/10.1016/j.aeue.2023.155016

Qiao, S., Wang, H., Zhao, C., & Zhang, T. (2021). BER enhanced receiver for PHO-OFDM-based optical wireless communications. *Physical Communication*, 47, 101350. https://doi.org/10.1016/j.phycom.2021.101350

Shamani, F., Shamani F., Airoldi R., Fakour V,Tapani A., & Nurmi J. (2016). FPGA Implementation Issues of a Flexible Synchronizer Suitable for NC-OFDM-Based Cognitive Radios. *Journal of Systems Architecture*, 76, 102–116. https://doi.org/10.1016/j.sysarc.2016.11.006

Shammaa, M., Mashaly, M., & El-mahdy, A. (2024). The Use of Deep Learning Techniques in OFDM Receivers for 5G NR: A Survey. *Procedia Computer Science*, 231, 32–39. https://doi.org/10.1016/j.procs.2023.12.154

Srinivasa, R. M., Madhumati, G. L., & Sailaja, M. (2024). An area efficient 64 point Radix-42 MDC FFT architecture for OFDM applications. *Integration*, 99, 102244. https://doi.org/10.1016/j.vlsi.2024.102244

Zegrar, S. E., & Arslan, H. (2022). Common CP-OFDM Transceiver Design for Low-Complexity Frequency Domain Equalization. *IEEE Wireless Communications Letters*, 11(7), 1349–1353. https://ieeexplore.ieee.org/document/9756521

Zhurakovskiy, B., Boiko, J., Druzhynin, V., & Pyatin, I. (2023). Performance Analysis of Concatenated Coding for OFDM Under Selective Fading Conditions. *CEUR Workshop Proceedings*, 3624, 403–413. https://ceur-ws.org/Vol-3624/Paper_33.pdf.

References

Boiko, Y., Eriomenko, O., & Huriyev, O. (2023). Capabilities of LDPC codes in improving the performance of optical telecommunications with OFDM. Measuring and Computing Devices in Technological Processes, (4), 13–26 [in Ukrainian]. https://doi.org/10.31891/2219-9365-2023-76-2

Boiko, Y., Eriomenko, O., & Pyatin, I. (2024). Evaluation of frequency-phase distortions in optical telecommunications with OFDM. Infocommunication Technologies and Electronic Engineering, 4(2), 130–140 [in Ukrainian]. https://doi.org/10.23939/ictee2024.02.130

Boiko, Y., Pyatin, I., Eriomenko, O., & Shaiuk, D. (2022). Evaluation of the impact of carrier frequency offset on the noise immunity of OFDM telecommunications. Measuring and Computing Devices in Technological Processes, (3), 19–26 [in Ukrainian]. https://doi.org/10.31891/2219-9365-2022-71-3-3

Boiko, Y. M., & Novikov, D. V. (2021). Evaluation of channel coding efficiency in OFDM telecommunications. Herald of Khmelnytskyi National University. Technical Sciences, 5(301), 150–159 [in Ukrainian]. https://doi.org/10.31891/2307-5732-2021-301-5-150-159

Algnabi, Y. S., Teymourzadeh, R., Othman, M., & Islam, M. S. (2018). FPGA Implementation of pipeline Digit-Slicing Multiplier-Less Radix 2 power of 2 DIF SDF Butterfly for Fourier Transform Structure. *Signal Processing*. https://doi.org/10.48550/arXiv.1806.04570

Alqahtani, A. S., Pandiaraj, S., Alshmrany, S. Ali Jaber A., Sandeep P. & U. Arun K. (2024). Enhancing MIMO-OFDM channel estimation in 5G and beyond with conditional self-attention generative adversarial networks. *Wireless Networks*, 30, 1719–1736. https://doi.org/10.1007/s11276-023-03615-y

Ayhan, T., Dehaene, W., & Verhelst, M. (2014). A 128:2048/1536 Point FFT Hardware Implementation with Output Pruning. In *European Signal Processing Conference (EUSIPCO)* (pp. 1–5). Isola delle Femmine – Palermo. https://www.eurasip.org/Proceedings/Eusipco/Eusipco2014/HTML/papers/1569925329.pdf

Kakkad, Y., Patel, D. K., Kavaiya, S., Sun, S. & López-Benítez, M. (2023). Optimal 3GPP Fairness Parameters in 5G NR Unlicensed (NR-U) and WiFi Coexistence. *IEEE Transactions on Vehicular Technology*, 72(4), 5373–5377. https://ieeexplore.ieee.org/document/9954188

Kleider, J. E., Maalouli, G., Gifford, S., & Chuprun, S. (2005). Preamble and embedded synchronization for RF carrier frequency-hopped OFDM. *IEEE Journal on Selected Areas in Communications*, 23(5), 920–931. https://ieeexplore.ieee.org/document/1425638

Kishore, K. K., Suman, J. V., Mallam, M., Hema, M., & Guntreddi, V. (2024). Scrambled UFMC and OFDM Techniques With APSK Modulation in 5G Networks Using Particle Swarm Optimization. *IEEE Access*, 12, 104091–104101. https://doi.org/10.1109/ACCESS.2024.3421311

Kumar, V. M., Selvakumar, D. A., & Sobha, P. M. (2015). Area and frequency optimized 1024 point Radix-2 FFT processor on FPGA. In *IEEE International Conference on VLSI-SATA* (pp. 1–6). Bengaluru, India. https://ieeexplore.ieee.org/document/7050487

Lalitha, H. & Kumar, N. (2024). Carrier frequency synchronization for WLAN systems based on MIMO-OFDM-IM. *EURASIP Journal on Wireless Communications and Networking*, 77, (2024). https://doi.org/10.1186/s13638-024-02406-z

Liu, J. & He, J. (2024). Design and analysis of CP-free OFDM PDMA transmission system. *EURASIP Journal on Advances in Signal Processing*, 68, (2024). https://doi.org/10.1186/s 13634-024-01154-y

Meenalakshmi, M., Chaturvedi, S. & Dwivedi, V. K. (2024). Enhancing channel estimation accuracy in polar-coded MIMO–OFDM systems via CNN with 5G channel models. *AEU – International Journal of Electronics and Communications*, 173, 155016. https://doi.org/10.1016/j.aeue.2023.155016

Qiao, S., Wang, H., Zhao, C., & Zhang, T. (2021). BER enhanced receiver for PHO-OFDM-based optical wireless communications. *Physical Communication*, 47, 101350. https://doi.org/10.1016/j.phycom.2021.101350

Shamani, F., Shamani F., Airoldi R., Fakour V,Tapani A., & Nurmi J. (2016). FPGA Implementation Issues of a Flexible Synchronizer Suitable for NC-OFDM-Based Cognitive Radios. *Journal of Systems Architecture*, 76, 102–116. https://doi.org/10.1016/j.sysarc.2016.11.006

Shammaa, M., Mashaly, M., & El-mahdy, A. (2024). The Use of Deep Learning Techniques in OFDM Receivers for 5G NR: A Survey. *Procedia Computer Science*, 231, 32–39. https://doi.org/10.1016/j.procs.2023.12.154

Srinivasa, R. M., Madhumati, G. L., & Sailaja, M. (2024). An area efficient 64 point Radix-42 MDC FFT architecture for OFDM applications. *Integration*, 99, 102244. https://doi.org/10.1016/j.vlsi.2024.102244

Zegrar, S. E., & Arslan, H. (2022). Common CP-OFDM Transceiver Design for Low-Complexity Frequency Domain Equalization. *IEEE Wireless Communications Letters*, 11(7), 1349–1353. https://ieeexplore.ieee.org/document/9756521

Zhurakovskiy, B., Boiko, J., Druzhynin, V., & Pyatin, I. (2023). Performance Analysis of Concatenated Coding for OFDM Under Selective Fading Conditions. *CEUR Workshop Proceedings*, 3624, 403–413. https://ceur-ws.org/Vol-3624/Paper_33.pdf.

Отримано редакцією журналу / Received: 26.03.25
Прорецензовано / Revised:28.04.25
Схвалено до друку / 22.05.25

ISSN 2707-1758

Juliy BOIKO, DSc (Engin.), Prof.
ORCID ID: 0000-0003-0603-7827
e-mail: boiko_julius@ukr.net
Khmelnytskyi National University, Khmelnytskyi, Ukraine

Ilya PYATIN, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0003-1898-6755
e-mail: ilkhmel@ukr.net
Khmelnytskyi Polytechnic Professional College by Lviv Polytechnic National University, Khmelnytskyi, Ukraine

Volodymir DRUZHYNIN, DSc (Engin.), Prof.
ORCID ID: 0009-0009-5049-0099
e-mail: v_druzhinin@ukr.net
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Oleksander EROMENKO, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0001-5110-3761
e-mail: yeromenko_s@ukr.net
Khmelnytskyi National University, Khmelnytskyi, Ukraine

FAST FOURIER TRANSFORM IN OFDM: ALGORITHMIC APPROACHES AND THEIR ROLE IN INFORMATION TECHNOLOGIES

Background. *Orthogonal Frequency Division Multiplexing (OFDM) is a key technology in modern information systems and is widely used in mobile networks such as 4G and 5G, the IEEE 802.11 standard (Wi-Fi), and digital television (DVB-T). The increase in the communication channel bandwidth requires an optimal selection of signal transformation parameters for efficient use of the hardware resources of Field-Programmable Gate Arrays (FPGA) in the implementation of OFDM.*

Methods. *The following methods were used: modeling of an OFDM-based communication system in the Simulink environment, which allowed for the study of signal processing transformations, as well as the analysis of the bit error rate (BER) for different modulation parameters. The implementation of FFT algorithms was carried out using HDL coding to compare the efficiency of the Fast Fourier Transform (FFT) algorithms Streaming Radix-2^2 and Burst Radix-2.*

Results. *Simulation results showed that using signal resampling at the transmitter improves the channel energy efficiency, reducing the required power level by 12 dB. The relationship between the bit error rate and the signal-to-noise ratio (SNR) demonstrates that increasing the FFT length from 512 to 2048 points requires a 6 dB increase in the SNR. The analysis of the cyclic prefix (CP) impact showed that the optimal CP length is 1/16 of the OFDM symbol, which reduces transmission speed losses. The effect of modulation on the bit error rate (BER) indicates the need for increased power when transitioning to higher-order Quadrature Amplitude Modulation (QAM).*

Conclusions. *It was concluded that the parameters of FFT and signal resampling are critical for the efficiency of the OFDM system. The results obtained can be used to optimize the implementation of OFDM on FPGA.*

Keywords: *Orthogonal Frequency Division Multiplexing (OFDM), Fast Fourier Transform (FFT), Information Systems, Bit Error Rate (BER), Field-Programmable Gate Array (FPGA).*

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

УДК 004.056:004.032.2:004.7
DOI: https://doi.org/10.17721/ISTS.2025.9.93-101

Руслан ПОЛІТАНСЬКИЙ, д-р техн. наук, проф.
ORCID ID: 0000-0003-0015-7123
e-mail: r.politansky@chnu.edu.ua
Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна

Валентин ЛЕСІНСЬКИЙ, канд. техн. наук, доц.
ORCID ID: 0000-0002-1259-1974
e-mail: v.lesinsky@chnu.edu.ua
Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна

Сергій ГАЛЮК, канд. техн. наук
ORCID ID: 0000-0003-3836-2675
e-mail: s.haliuk@chnu.edu.ua
Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна

Андрій КУЛЬКО, асп.
ORCID ID:0009-0006-1185-0774
e-mail: kulko452@gmail.com
Київський національний університет імені Тараса Шевченка, Київ, Україна

МЕТОДИ ВИЗНАЧЕННЯ РІВНОВАЖНОГО СТАНУ МЕРЕЖІ ІЗ ЗАДАНИМИ ПОТОКАМИ У ВУЗЛАХ ІНФОРМАЦІЙНИХ МЕРЕЖ КІБЕРФІЗИЧНИХ СИСТЕМ

Вступ. *Багатовимірність кіберпростору та високий рівень інформатизації об'єктів критичної інфраструктури як в Україні, так і у світі, становлять загрозу для людства на глобальному рівні на досяжну перспективу. Ситуація ускладнюється тим, що об'єкти критичної інфраструктури, які функціонують в єдиному інформаційному просторі й підтримують широкий спектр сучасних інформаційних технологій, всупереч колосальним зусиллям для протидії стороннім втручанням із кіберпростору та через кіберпростір, й надалі залишаються вразливими до загроз нового типу.*

Методи. *Виявлено, що дослідження складних систем використовують такі сучасні методи математичної науки, як топологія, теорія графів, лінійна алгебра.*

Результати. *Ентропійний метод аналізу складних систем широко застосовують у прикладних задачах дослідження природних, технічних і кіберфізичних систем. Ентропійний метод давно використовують для побудови ефективних систем кодування, шифрування та захисту інформації. У зв'язку із зростанням обчислювальних потужностей збільшилася практична цінність дослідження динамічних систем, зокрема й основаних на ентропії, оскільки ентропія є хорошим показником вибірки деяких станів системи із множини всіх її можливих станів.*

Висновки. *Оскільки дослідження структур і загальних закономірностей зустрічається в багатьох галузях науки і техніки, актуальними є задачі, що мають загальний характер, коли конкретна природа системи не береться до уваги. У цій роботі наведено алгоритм визначення стану багатовузлової мережі інформаційних потоків, у якому її ентропія набуває максимальне значення. Алгоритм основано на виведених авторами роботи аналітичних виразах ентропії, та її перших і других похідних, визначених на множині ядрових розв'язків системи лінійних рівнянь, до якої зводиться вся початкова задача. Наведено експериментальні результати визначення множини розв'язків і розв'язання оптимізаційної задачі пошуку стану з максимальним значенням ентропії для мереж із 3-, 4-, 5- та 6-ма вузлами. Проаналізовано можливості подальшого вдосконалення методу для розрахунку складніших мереж.*

Ключові слова: *програмний комплекс, мережний трафік, матриці, графи, ентропія, матриці Гессе, підматриця.*

Вступ

В умовах збройної агресії рф проти України безпека людини, суспільства і держави суттєво залежить від надійного функціонування об'єктів критичної інфраструктури. Окрім фізичних впливів на такі об'єкти летальною зброєю рф не полишає спроб разом зі своїми сателітами впливати нелетальними засобами – кіберзброєю – на системи управління об'єктів критичної інфраструктури через кіберпростір і з кіберпростору. З огляду на транскордонність кіберпростору та високий рівень інформатизації об'єктів критичної інфраструктури як в Україні, так і у світі, наприклад, об'єктів ядерної енергетики й об'єктів водопостачання, систем управління військами та зброєю Збройних сил України та її складових тощо, ризики масштабуються не тільки на національний безпековий вимір, а і становлять загрозу для людства на глобальному рівні на досяжну перспективу (Beshley et al., 2018).

Ситуація ускладнюється тим, що об'єкти критичної інфраструктури, які функціонують в єдиному інформаційному просторі й підтримують широкий спектр сучасних інформаційних технологій, всупереч колосальним зусиллям для протидії стороннім втручанням із кіберпростору та через кіберпростір, і надалі залишаються вразливими до загроз нового типу. Дослідження складних систем використовує такі сучасні галузі математичної науки, як топологія, теорія графів, лінійна алгебра. Оскільки дослідження структур і загальних закономірностей відбувається у багатьох галузях науки і техніки, то актуальними є задачі, що мають загальний характер, коли конкретну природу системи не беруть до уваги.

Огляд літератури. Обчислення глобальної ентропії, яка характеризує багатовузлову мережу, має практичне застосування у системах захисту інформації (Garofalakis, Gehrke, & Rastogi, 2016). Коли йдеться про раптову зміну значення ентропії, це може бути ознакою хакерської атаки типу DDoS (Distributed Denial of Service) на мережу (Li, & Sun, 2021). У роботі (Madarro-Capó et al., 2021) наведено алгоритми, що основані на ентропії системи, які дають можливість розв'язати "проблему зменшення комунікаційних витрат у розподілених системах" за рахунок зменшення надмірного використання деяких обчислювальних вузлів системи порівняно з іншими її вузлами. Ентропійний метод є також класичним методом аналізу ефективності систем шифрування інформації (Miletic et al., 2022).

ISSN 2707-1758

Методи

В роботі (Niven et al., 2019) представлено розроблення програмного комплексу аналізу мережного трафіка та виявлення вторгнень. Дослідженню систем з інтенсивними інформаційними потоками присвячено роботу (Pedrosa, & Costa, 2022), де виведено деякі важливі характеристики системи, що характеризують її загалом: основні методи вибірки та логічного висновку для потоків даних; часті схеми руху (найбільш вживані IP-адреси джерела й адресата); найбільш часті адресати для заданого джерела, і навпаки, тощо. Розглянуто також різні методи дискретизації потоків даних для їх подальшого аналізу (пуассонівські потоки, алгоритми резервуарних потоків Watterman і McLeod), кластеризація потоків даних.

Важливу роль дослідження ентропії набувають у зв'язку з виникненням і поширеним застосуванням динамічних мереж із змінною топологією з'єднання вузлів. Такі мережі виникають у задачах використання безпілотних літальних апаратів для забезпечення функціонування мережі, у технологіях ройового використання безпілотних літальних апаратів, технологіях бездротового зв'язку 5G у безлюдних місцевостях, де економічно необґрунтовано використовувати стаціонарні базові станції, а також у інших ситуаціях, де в інформаційному обміні беруть участь машини без участі людини і де це обумовлено правилами безпеки: місця природних і техногенних катастроф, космічні місії тощо. Дослідження динамічних станів мережі наведено в роботах (Politanskyi et al., 2022) на основі теорії мультиграфів, у (Shahar, Alfassi, & Keren, 2022) – на підставі прогнозування зміни станів мережі в бік збільшення її ентропії.

Розглянемо мережу, яку можна задати плоским повнозв'язним графом, із двонаправленими ребрами, без петльових дуг, що зображений на рис. 1 (для мережі з $n = 5$ вузлів). Для цієї мережі задано сумарні значення потоків, які надходять і виходять із кожного вузла (на рис. 1 вони позначені як f_{ij}):

$$\begin{aligned} \sum_{j=1}^{n-1} f_{ij} = s_i, i \in \{1,2,\dots,n\}, \\ \sum_{i=1}^{n-1} f_{ji} = d_j, j \in \{1,2,\dots,n\}, \end{aligned} \quad (1)$$

де n – порядок системи, що збігається з кількістю вузлів.

Вважатимемо, що петльових (самозамкнених) потоків у системі немає, тоді систему можна відобразити матрицею потоків:

$$\|F\|_{n\times n} = \|f_{ij}\|_n = \begin{bmatrix} 0 & \dots & f_{1n-1} \\ \dots & 0 & \dots \\ f_{n,1} & \dots & 0 \end{bmatrix}, \quad (2)$$

де $\|F\|_{n\times n}$ – матриця розмірності $n \times n$.

Отже, усього є $n \cdot (n-1)$ змінних системи. Вважаємо також, що суми всіх вхідних і всіх вихідних потоків збігаються:

$$\sum_{i=1}^{n} s_i = \sum_{i=1}^{n} d_i = S_0. \quad (3)$$

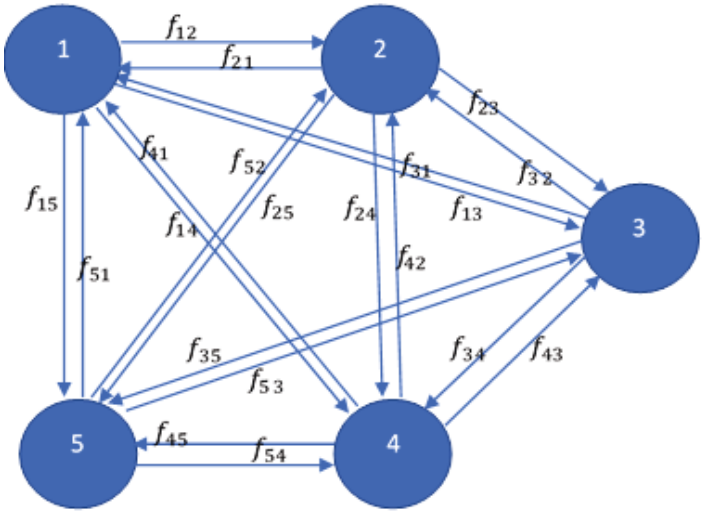


Рис.1. Графічне відображення мережі з п'ятьма вузлами

Тоді рівняння системи, яке враховує умови (1), можна записати у матричному вигляді

$$\|A\|_{(2\cdot n-1)\times(n^2-n)} \times \vec{f}_{n^2-n} = \vec{v}_{2\cdot n-1}, \quad (4)$$

де $\vec{v}$ – стовпець, складений із значень суми вхідних і вихідних потоків для кожного вузла системи, які впорядковані з метою реалізації алгоритму визначення множини розв'язків системи:

$$\vec{v} = (d_2, d_3, \dots, d_n, d_1, s_2, \dots, s_{n-1}, s_n). \quad (5)$$

Розмірність вектора $\vec{v}$ дорівнює $2 \cdot n - 1$ і збігається з кількістю рівнянь, які забезпечують виконання умов (1), одне рівняння є лінійно залежним від інших, оскільки виконується ще й умова (3). Стовпець змінних системи $\vec{f}$ складений з усіх значень потоків f_{ij} матриці, яка описує систему, які розміщені у порядку проходження стовпців матриці згори донизу, і зліва праворуч:

$$\vec{f} = \left(f_{21}, f_{31}, \dots, f_{n1}, f_{11}, f_{32}, \dots, f_{n2}, \dots, f_{n,n-1}, f_{1,n-1}, f_{2,n-1}, \dots, f_{n-1,n-1}\right). \quad (6)$$

Таке розміщення змінних і початкових умов задачі обумовлено подальшим спрощенням виконання алгоритму пошуку розв'язків задачі.

Кількість лінійно незалежних рівнянь у системі дорівнює $r = 2 \cdot n - 1$, і є меншою за загальну кількість змінних $n^2 - n$. Тому серед змінних можна виділити $\{\lambda_j\}$, $j \in \{1, \dots, s\}$, ядрових (незалежних змінних):

$$s = n^2 - 3 \cdot n + 1. \quad (7)$$

Усі інші $\{x_i\}$, $i \in \{1, \dots, r\}$, є лінійними функціями розв'язків $\{\lambda_j\}$.

ISSN 2707-1758

Визначення множини розв'язків системи для мережі з п'яти вузлів. Визначення множини розв'язків утвореної системи здійснюємо, виконуючи операції віднімання рядків матриці та перестановкою стовпців матриці, внаслідок якої у лівій частині матриці $\|A\|$ буде виділено одиничну діагональну матрицю, розмірність якої $(2 \cdot n - 1) \times (2 \cdot n - 1)$. Зауважимо, що варто дотримуватись такої логіки: після виконання операцій віднімання у деякому стовпці майбутньої одиничної матриці залишиться лише одна одиниця. Перестановку слід виконувати тоді, коли наступний стовпець, який має бути стовпцем одиничної матриці, не містить одиницю в потрібному рядку. Операція віднімання стовпців матриці не змінює розв'язки усієї системи, а операція перестановки стовпів матриці міняє місцями розв'язки, індекси яких у стовпці $\vec{f}$ збігаються з номерами стовпців матриці, що міняються місцями.

Отже, зручне розміщення змінних системи дає можливість одразу виділити у матриці $\|A\|$ одиничну діагональну підматрицю розмірності $n - 1$ (рис. 1а, де наведено матрицю $\|A\|$ для мережі з $n = 5$ вузлами).

Далі можемо збільшити ранг одиничної матриці без перестановок стовпців матриці. Для цього потрібно послідовно виконати дві операції: відняти від рядка 6 рядок 5 (рис. 1б); відняти від рядка 2 рядок 6 (рис. 1в).

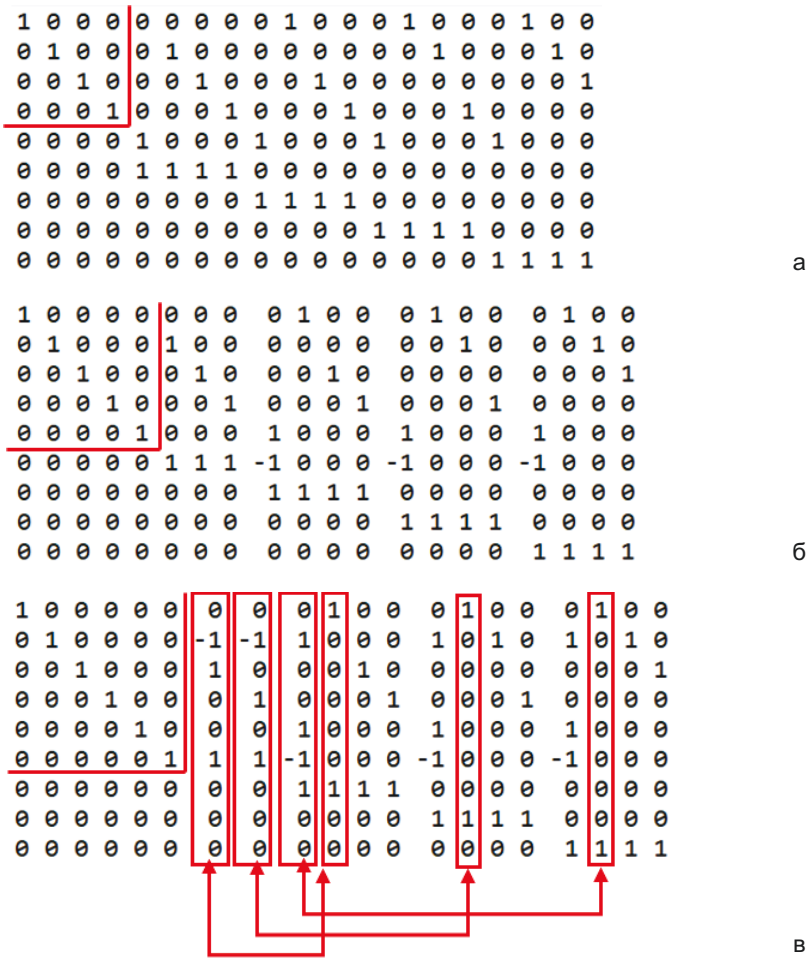


Рис. 1. Початковий етап виділення незалежних розв'язків системи (а та б), позначення перестановок стовпців (в)

Подальше виділення діагональної матриці можна виконувати тільки після перестановок стовпців. Для матриці, зображеної на рис. 1, потрібно виконати перестановку трьох наступних стовпців (7-й, 8-й і 9-й), для цього можна вибрати будь-які три інші стовпці, що містяться ліворуч від них і які мають одиницю в 7-му, 8-му і 9-му рядках відповідно. З погляду збільшення швидкодії алгоритму доцільним є вибір тих стовпців, які мають мінімальну кількість одиниць, щоб потім зменшити кількість операцій віднімання. Такими є стовпці 10-й, 14-й і 18-й (рис. 1в).

На рис. 2а показано вигляд матриці $\|A\|$ після перестановок стовпців. закінчивши перестановки потрібно відняти від першого рядка 7-й, 8-й і 9-й рядки. На рис. 2б бачимо кінцевий результат перетворення.

Отже, матриця, яку ми шукали має вигляд

$$\|R\|_5 = \begin{bmatrix} 0 & -1 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & 0 & 1 & -1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (8)$$

Матриця $\|R\|_5$ визначає лінійний зв'язок між 11 ядровими змінними та іншими 9 розв'язками системи.

Позначимо операцію віднімання i-го і j-го рядків як $Su(i,j)$, а їхню перестановку як $P(i,j)$. Тоді

ISSN 2707-1758

послідовність перетворень для матриці, яка описує мережу з 5 вузлами можна записати у вигляді

$$O_5 = Su(6,5) \to Su(2,6) \to P(7,10) \to P(8,14) \to P(9,18) \to Su(1,7) \to Su(1,8) \to Su(1,9),$$

або компактніше

$$O_5 = Su[(6,5),(2,6)] \to P[(7,10),(8,14),(9,18)] \to Su[(1,7),(1,8),(1,9)]. \quad (9)$$

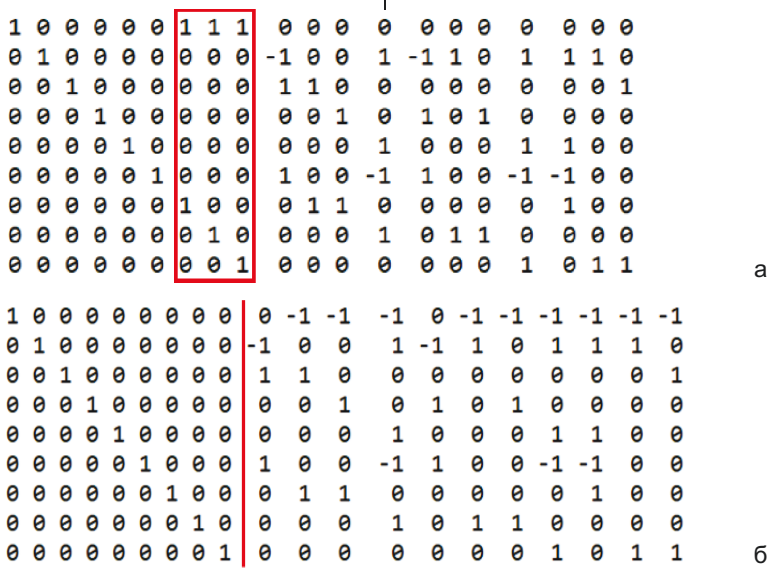


Рис. 2. Перестановка стовпців для виділення діагональної одиничної матриці: матриця після перестановок (а), кінцевий результат перестановок (б)

Разом із відніманням рядків потрібно так само віднімати елементи стовпця $\vec{v}$ у формулі (4), тоді як операції перестановок стовпців матиці $\|A\|$ не змінюють його. Відповідно до розташування елементів стовпця $\vec{v}_5$, операції віднімання змінять його, утворюючи у цьому разі стовпець

$$\vec{b}_5{:}\vec{v}_5 = \begin{bmatrix} d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix} \to \vec{b}_5 = \begin{bmatrix} d_2 - (s_3 + s_4 + s_5) \\ d_1 + d_3 - s_2 \\ d_4 \\ d_5 \\ d_1 \\ s_2 - d_1 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix} \quad (10)$$

Операція перестановок стовпців змінить вектор змінних системи: $\vec{f} \to \vec{f}$, але значення ентропії при цьому не змінюється.

Відтак розв'язки системи утворюють деякий лінійний підпростір 11-вимірного простору незалежних змінних системи $\vec{\lambda}_5^T = (\lambda_1, \lambda_2, \dots, \lambda_{11})$:

$$\vec{f}_5 = -\left\|\frac{R}{E_{11\times11}}\right\|_5 \times \vec{\lambda}_5 + \left\|\frac{\vec{b}_5}{\vec{O}_{11}}\right\|, \quad (11)$$

де $E_{11\times11}$ – одинична діагональна матриця розмірності 11, $\vec{O}_{11}^T = (0,0,0,0,0,0,0,0,0,0,0)$ – вектор, утворений із 11 елементів, що дорівнюють нулю.

Очевидним обмеженням на розв'язки задачі є те, що потоки повинні мати невід'ємні значення: $f_{ji} \geq 0$. Це накладає також обмеження на значення незалежних змінних системи: $\lambda_j \geq 0$. Виникають також інші, нетривіальні обмеження значень $\{\lambda_j\}$, які забезпечують невід'ємність інших розв'язків системи $\{x_i\}$:

$$\begin{cases} \sum_{j=1}^{11} \|r_{ij}\| \cdot \lambda_j \leq \vec{b}_5(i), i \in \{1, \dots, 9\} \\ \lambda_j \geq 0, j \in \{1, \dots, 11\} \end{cases}. \quad (12)$$

Алгоритм визначення розв'язків системи для довільної кількості вузлів. Узагальнимо розглянутий метод для системи довільного порядку.

В результаті перших двох операцій віднімання рядків виділяють одиничну діагональну матрицю розмірності $n+1$ (6 для матриці, що зображена на рис. 1). Вказані операції є однотипні для мереж із будь-якою кількістю вузлів. Причому номери рядків, що віднімаються на першому кроці дорівнюють $n+1$ та n, а на другому кроці – 2 та $n+1$. Отже, для мережі довільного порядку ці дві операції можна зобразити у символьному вигляді як $Su[(n+1,n),(2,n+1)]$. Очевидно, що кількість операцій перестановок стовпців становить $2 \cdot n - 1 - (n+1) = n-2$. Зауважимо, що номери стовпців, які переставляємо ліворуч, послідовно збільшуються, починаючи із стовпця з номером $n+2$. Стовпці, які переставляємо праворуч, можуть бути довільними, але для зручності реалізації алгоритму вибираємо ті, що містять тільки дві одиниці: в першому рядку і в тому рядку, який містить наступну одиницю діагональної матриці. Визначити номери таких стовпців можна, якщо проаналізувати структуру початкової матриці. Це стовпці, номери яких утворюють послідовність $\{2 \cdot n + i \cdot (n-1)\}$, де i –послідовний номер стовпця, який переставляємо, $i \in \{1,2,\dots,n-2\}$. Відтак для мережі довільного порядку символьне позначення операцій перестановок можна позначити, як $\prod_{i=1}^{n-2} P\big(n+i+1,2 \cdot n + i \cdot (n-1)\big)$, де символом $\prod_{i=1}^{n-2}$ позначено послідовне виконання операцій із змінним індексом $i \in \{1,2,..,n-2\}$. Після операцій перестановок потрібно виконувати віднімання рядка з номером стовпця, що переставляється від першого рядка табл. 1 (рис. 2б, відповідні стовпці виділено червоним кольором), у символьному позначенні операції віднімання можна позначити як

ISSN 2707-1758

$\prod_{i=1}^{n-2} S(1, n+i+1)$. Отже узагальнена формула виконання операцій алгоритму має вигляд

$$S[(n+1,n),(2,n+1)] \to \prod_{i=1}^{n-2} P[n+i+1, 2n+i(n-1)] \to \prod_{i=1}^{n-2} S(1, n+i+1). \quad (13)$$

Таким способом початкова матриця $\|A\|$ розділиться на дві частини: ліворуч утворюється одинична діагональна матриця, а праворуч – залишкова матриця $\|R\|$, яка є основою для подальшого визначення розв'язків системи:

$$\|A\|_{(2\cdot n-1)\times n\cdot(n-1)} \to \|E\|_{(2\cdot n-1)\times(2\cdot n-1)} \| \|R\|_{(2\cdot n-1)\times(n^2-3\cdot n+1)}. \quad (14)$$

Розмірність матриці $\|R\|$ визначається кількістю незалежних рівнянь системи і кількістю незалежних змінних:

$$dim(\|R\|) = (2\cdot n-1)\times s. \quad (15)$$

Елементи матриці $\|R\|$ можуть набувати лише три значення:

$$r_{ij} \in \{-1,0,1\},\ i \in \{1,2,\dots,2\cdot n-1\},\ j \in \{1,2,\dots,s\}.$$

Враховуючи тільки операції віднімання, із формули (13) визначимо коефіцієнт вільних доданків $\vec{b}_n$:

$$\begin{cases} b_i = v_i, i \neq \{1,2,n+1\} \\ b_1 = v_1 - \sum_{i=1}^{n-2} v_{i+2} \\ b_2 = v_2 - (v_{n+1} - v_n) \\ b_{n+1} = v_{n+1} - v_n \end{cases}. \quad (17)$$

Підставляючи значення $\{v_i\}$, знаходимо значення $\{b_i\}$, виражені через початкові умови задачі:

$$\begin{cases} b_1 = d_2 - \sum_{i=1}^{n-2} s_{i+2} \\ b_2 = d_1 + d_3 - s_2 \\ b_i = d_{i+1}, i \in \{3,4,\dots,n-1\} \\ b_n = d_1 \\ b_{n+1} = s_2 - d_1 \\ b_i = s_i, i \in \{3,4,\dots,n\} \end{cases}. \quad (18)$$

Множину розв'язків усієї системи можна записати у вигляді матричного рівняння:

$$\vec{\tilde{f}}_n = -\left\| \frac{\|R\|_{(2\cdot n-1)\times(n^2-3\cdot n+1)}}{E_{(n^2-3\cdot n+1)\times(n^2-3\cdot n+1)}} \right\|_n \times \vec{\lambda}_{n^2-3\cdot n+1} + \left\| \frac{\vec{b}_n}{\vec{O}_{n^2-3\cdot n+1}} \right\|. \quad (19)$$

Значення незалежних змінних $(\lambda_1, \lambda_2, \dots, \lambda_s)$ повинні забезпечувати виконання умови невід'ємності потоків усіх потоків $\vec{\tilde{f}}_n$ у мережі. Із цього слідують обмеження на розв'язки системи $\{\lambda_j\}$, викладені у вигляді нерівностей:

$$\begin{cases} \sum_{j=1}^{s} r_{ij}\cdot\lambda_j \leq b_i \text{ for } i \in \{1,2,\dots,2\cdot n-1\} \\ \lambda_j \geq 0 \text{ for } j \in \{1,2,\dots,n^2-3\cdot n+1\} \end{cases}. \quad (20)$$

Визначення оптимального стану системи ентропійним методом. Маючи аналітичний вигляд розв'язків системи, можемо поставити задачу визначення оптимального значення ентропії. Ентропія мережі з інформаційними потоками може бути визначена на основі нормованого значення потоків, сума яких для всієї матриці потоків рівна 1.

$$f_{ij}^{norm} = f_{ij}/S_0, \quad (21)$$

де f_{ij} – значення потоку між вузлами i та j, f_{ij}^{norm} – нормоване значення потоку між вузлами i та j, S_0 – сума всіх потоків у мережі: $S_0 = \sum_{i=1}^{n}\sum_{j=1}^{n-1} f_{ij}$, тут n – кількість вузлів у мережі, тоді виконується стандартна умова нормування:

$$\sum_{i=1}^{n}\sum_{j=1}^{n-1} f_{ij}^{norm} = 1, \quad (22)$$

яка виконується, коли ентропія визначається за ймовірностями станів системи. Тому можливе використання аналогії між інтенсивністю потоків між деякими вузлами та ймовірністю стану системи, в якому реалізується це значення.

Далі використовуємо логарифмічну міру для визначення ентропії (за аналогією з ентропією системи, яка визначається за ймовірністю її станів):

$$E = -\sum_{i=1}^{n}\sum_{j=1}^{n-1} \frac{f_{ij}}{S_0}\cdot\log_2\left(\frac{f_{ij}}{S_0}\right). \quad (23)$$

Для подальшого аналізу змінимо послідовну нумерацію множини розв'язків системи, виділивши окремо незалежні змінні $(\lambda_1, \lambda_2, \dots, \lambda_s)$:

$$\vec{\tilde{f}}_n = (x_1, x_2, \dots, x_{2\cdot n-1}, \lambda_1, \lambda_2, \dots, \lambda_s), \quad (24)$$

де розв'язки $\{x_i\}$ можуть бути визначені з матричного рівняння у вигляді лінійної комбінації незалежних змінних:

$$x_i = -\sum_{k=1}^{s} r_{ik}\cdot\lambda_k + b_i, i \in \{1,2,\dots,2\cdot n-1\}. \quad (25)$$

Тоді вираз для ентропії матиме вигляд

$$E = -\frac{1}{S_0}\cdot\sum_{i=1}^{2\cdot n-1} x_i\cdot\log_2\left(\frac{x_i}{S_0}\right) - \frac{1}{S_0}\cdot\sum_{i=1}^{s}\lambda_s\cdot\log_2\left(\frac{\lambda_s}{S_0}\right) = E_1 + E_2. \quad (26)$$

З огляду на вище наведені формули, можемо зробити висновок, що ентропія є функцією незалежних (ядрових) змінних системи:

$$E = E(\lambda_1, \lambda_2, \dots, \lambda_{s-1}, \lambda_s). \quad (27)$$

Далі шукатимемо стан мережі, в якому ентропія набуває максимальне значення з урахуванням обмежень (20). Оптимізаційна задача пошуку рівноважного стану всієї мережі може бути сформульована так:

$$\max_{\lambda_1,\lambda_2,\dots,\lambda_{s-1},\lambda_s} E = E(\lambda_1, \lambda_2, \dots, \lambda_{s-1}, \lambda_s)$$
$$\text{subject to } \sum_{i=1}^{s} r_{ji}\cdot\lambda_i \leq b_j \text{ for } j \in \{1,2,\dots,2\cdot n-1\}. \quad (28)$$
$$\lambda_i \geq 0 \text{ for } i \in \{1,2,\dots,s\}$$

Відтак для визначення оптимального стану мережі визначимо частинні похідні ентропії $E(\lambda_1, \lambda_2, \dots, \lambda_{s-1}, \lambda_s)$ за кожною незалежною змінною $\lambda_i \in \{\lambda_1, \lambda_2, \dots, \lambda_{s-1}, \lambda_s\}$. Для зручності визначатимемо похідні ентропії від $E_1(\lambda_1, \lambda_2, \dots, \lambda_{s-1}, \lambda_s)$ та $E_2(\lambda_1, \lambda_2, \dots, \lambda_{s-1}, \lambda_s)$ окремо (26):

$$\frac{\partial E_1}{\partial\lambda_j} = \frac{\partial}{\partial\lambda_j}\left[-\frac{1}{S_0}\cdot\sum_{i=1}^{2\cdot n-1} x_i\cdot\log_2\left(\frac{x_i}{S_0}\right)\right] = -\frac{1}{S_0}\cdot\sum_{i=1}^{2\cdot n-1}\frac{\partial}{\partial\lambda_j}\left[x_i\cdot\log_2\left(\frac{x_i}{S_0}\right)\right], \quad (29)$$

ISSN 2707-1758

$$\frac{\partial E_2}{\partial \lambda_j} = \frac{\partial}{\partial \lambda_j}\left[-\frac{1}{S_0}\cdot\sum_{i=1}^{s}\lambda_i\cdot\log_2\left(\frac{\lambda_i}{S_0}\right)\right] = -\frac{1}{S_0}\cdot\sum_{i=1}^{s}\frac{\partial}{\partial \lambda_j}\left[\lambda_i\cdot\log_2\left(\frac{\lambda_i}{S_0}\right)\right]. \quad (30)$$

Знайдемо похідні від кожного доданка (29) і (30):

$$\frac{\partial}{\partial \lambda_j}\left[x_i\cdot\log_2\left(\frac{x_i}{S_0}\right)\right] = -r_{ij}\cdot\left[\frac{1}{\ln 2}+\log_2\left(\frac{x_i}{S_0}\right)\right], \quad (31)$$

$$\frac{\partial}{\partial \lambda_j}\left[\lambda_i\cdot\log_2\left(\frac{\lambda_i}{S_0}\right)\right] = \delta_{ij}\cdot\left[\frac{1}{\ln 2}+\log_2\left(\frac{\lambda_i}{S_0}\right)\right], \quad (32)$$

де δ_{ij} символ Кронекера: $\delta_{ij} = 1$ for $i = j$, $\delta_{ij} = 0$ for $i \neq j$.

Далі підставляємо (31) і (32) у (29) і (30), і додаємо вирази для похідних $\frac{\partial E_1}{\partial \lambda_j}$ і $\frac{\partial E_2}{\partial \lambda_j}$:

$$\frac{\partial E}{\partial \lambda_j} = -\frac{\left[\frac{1}{\ln 2}+\log_2\left(\frac{\lambda_j}{S_0}\right)\right]}{S_0}+\frac{1}{S_0}\cdot\sum_{i=1}^{2\cdot n-1} r_{ij}\cdot\left[\frac{1}{\ln 2}+\log_2\left(\frac{-\sum_{k=1}^{s} r_{ik}\cdot\lambda_k+b_i}{S_0}\right)\right]. \quad (33)$$

Так само знаходимо похідні другого порядку $\frac{\partial^2 E}{\partial \lambda_l \partial \lambda_j}$:

$$\frac{\partial^2 E}{\partial \lambda_l \partial \lambda_j} = -\frac{1}{\ln 2\cdot S_0}\cdot\left\{\frac{1}{\lambda_j}\cdot\delta_{lj}+\sum_{i=1}^{2n-1}\frac{r_{ij}\cdot r_{il}}{b_i-\sum_{k=1}^{s} r_{ik}\cdot\lambda_k}\right\}, \quad (34)$$

або

$$\frac{\partial^2 E}{\partial \lambda_l \partial \lambda_j} = -\frac{1}{\ln 2\cdot S_0}\cdot\left\{\frac{1}{\lambda_j}\cdot\delta_{lj}+\sum_{i=1}^{2n-1}\frac{r_{ij}\cdot r_{il}}{x_i}\right\}. \quad (35)$$

Результати

Визначення множини цілочисельних розв'язків системи. Задача визначення всієї множини розв'язків системи є дуже складною з погляду швидкодії та потужності обчислювальних ресурсів, які необхідні для її розв'язання. Алгоритм цієї задачі ґрунтується на циклічній перевірці умов (1) на значення незалежних змінних, і в разі їх генерування цілочисельних розв'язків задачі. Але у зв'язку з обчислювальною складністю, крок визначення значень незалежних параметрів варіювався від 2 до 5 залежно від порядку системи. У табл. 1 наведено результати обчислень.

Таблиця 1

Результати визначення підмножин розв'язків системи із *n* = 3, 4, 5 і 6 вузлів

Порядок системи, початкові умови задачі	Крок вибірки незалежних параметрів	Кількість розв'язків	Середнє значення ентропії	Максимальне значення ентропії
$n = 3,\ S_0 = 1000$	1	1499	1,21	2,42
$n = 4,\ S_0 = 1000$	20	1125	1,7	3,40
	10	33469	1,75	3,41
$n = 5,\ S_0 = 1130$	50	3517	1,99	3,97
$n = 6,\ S_0 = 1580$	80	899	2,22	4,44

Очевидно, що пошук розв'язків методом циклічної перевірки умов (1) значно обмежує можливість його застосування для задач високої розмірності ($n > 7$). Тому залишається відкритим питання про можливість ефективного пошуку хоча б одного розв'язку задачі для застосування оптимізаційних алгоритмів для розв'язання задач більш розмірності ($n > 7$).

Визначення оптимального стану з максимальним значенням ентропії на основі аналітичних виразів. Оскільки метод визначення всієї множини розв'язків потребує значних обчислювальних ресурсів, або значних витрат машинного часу, то розв'язки визначали з деяким кроком дискретизації у просторі $\{\lambda_j\}$. Тому оптимальний розв'язок, для якого реалізується глобальний максимум ентропії, у більшості випадків пропущено, хоча стан системи є наближеним до рівноважного.

Для точного визначення глобального максимуму ентропії використовуємо властивість унімодальності цієї функції. Наведемо без доведення дві леми, які дозволяють зробити висновок про те, що ентропія є унімодальною (Shahar, Alfassi, & Keren, 2022).

Лема 1. Сума унімодальних функцій вигляду

$$f_j(\lambda_j) = -\sum_{i=1}^{r}\frac{(\tilde{b}_{ij}-r_{ij}\cdot\lambda_j)}{S_0}\cdot\log_2\left(\frac{\tilde{b}_{ij}-r_{ij}\cdot\lambda_j}{S_0}\right)-\frac{\lambda_j}{S_0}\cdot\log_2\left(\frac{\lambda_j}{S_0}\right),$$
$$\tilde{b}_{ij} = b_i-\sum_{k=1,k\neq j}^{s} r_{ik}\cdot\lambda_k,\ j\in\{1,\dots,s\}$$

є унімодальною функцією змінної λ_j.

Лема 2. Багатовимірна функція $f(\lambda_1,\dots,\lambda_s)$ є унімодальною функцією.

Для знаходження оптимального стану мережі можемо далі застосувати чисельні методи, а саме метод другого порядку – метод Ньютона:

$$\vec{\lambda}^{(k+1)} = \vec{\lambda}^{(k)} - \left(\|H\|^{(k)}\right)^{-1}\times\nabla E^{(k)}, \quad (36)$$

де $\|H\|$ – матриця Гессе (гесіан), що складена з похідних другого порядку (33):

$$\|H\| = \left\|\frac{\partial^2 E}{\partial \lambda_l \partial \lambda_j}\right\|. \quad (37)$$

Одним із критеріїв збіжності методу є визначення норми градієнта на кожному кроці ітерації. Для цього використано евклідову міру відстані і довжини вектора у багатовимірному просторі:

$$|\nabla E| = \sqrt{\sum_{j=1}^{s}\left(\frac{\partial E}{\partial \lambda_j}\right)^2}. \quad (38)$$

Для найменш складної системи $n = 3$ на рис. 3 зображено графіки ентропії, першої і другої похідних, обчислених на множині значень єдиного параметра λ для системи із $S_0 = 100$.

ISSN 2707-1758

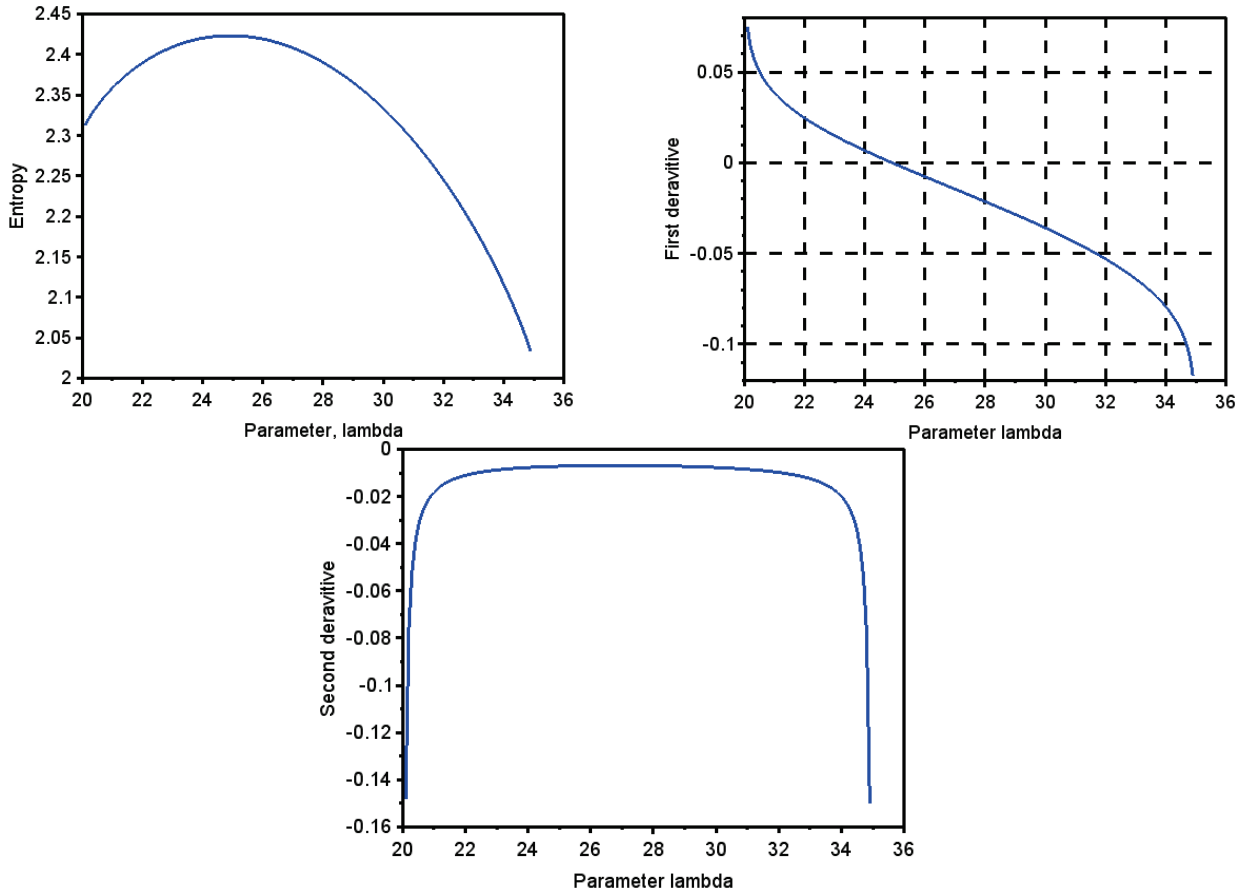


Рис. 3. Ентропія $E(\lambda)$та похідна $\frac{\partial E}{\partial \lambda}$ для системи n = 3

Чисельний метод показує дуже хорошу збіжність: для визначення оптимального значення для всіх досліджених порядків системи (n = 3, 4, 5, 6) достатньо всього 5 ітерацій методу. Результати ітераційного процесу зображено на рис. 4а, де показано обчислену ентропію, і на рис. 4б, де показано норму градієнта на кожному кроці ітерації.

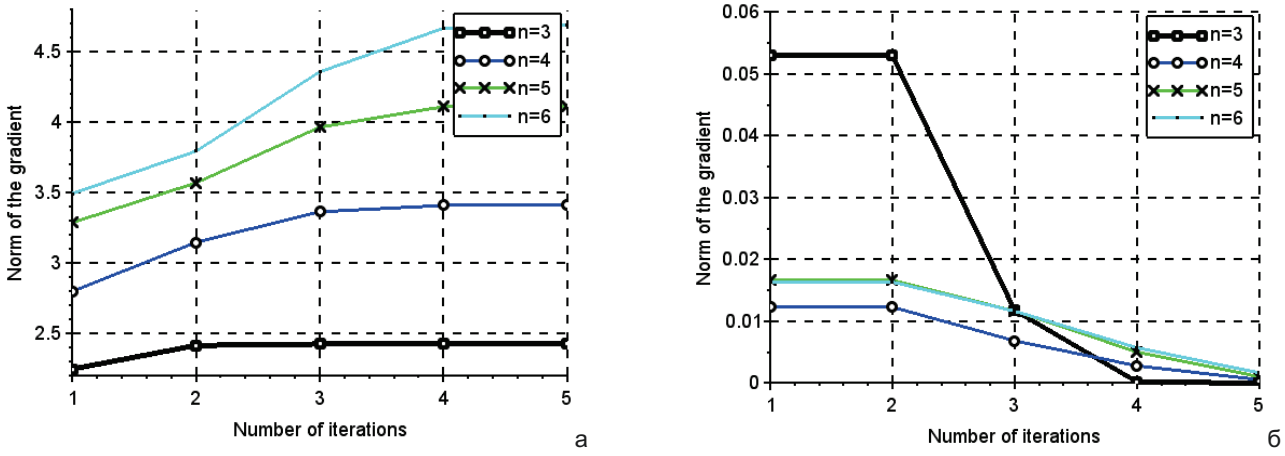


Рис. 4. Збіжність методу для системи дослідженої розмірності (n = 3, 4, 5, 6): а – ентропія; б – норма градієнта

Зауважимо, що залежно від вибору початкової точки алгоритм, який оснований на застосуванні методу Ньютона, може визначити кілька рівноправних розв'язків, для яких ентропія мережі має однакові значення. Це можна зрозуміти, якщо зважити на те, що абсолютне значення градієнта має досить малі значення. Правильність такого результату підтверджують статистичні розподіли ентропії, які одержані нами на основі множини розв'язків системи (табл. 1). Для мереж із 3, 4, 5 і 6 вузлами на рис. 5 показано гістограми значень ентропії, розділені на 10 інтервалів.

ISSN 2707-1758

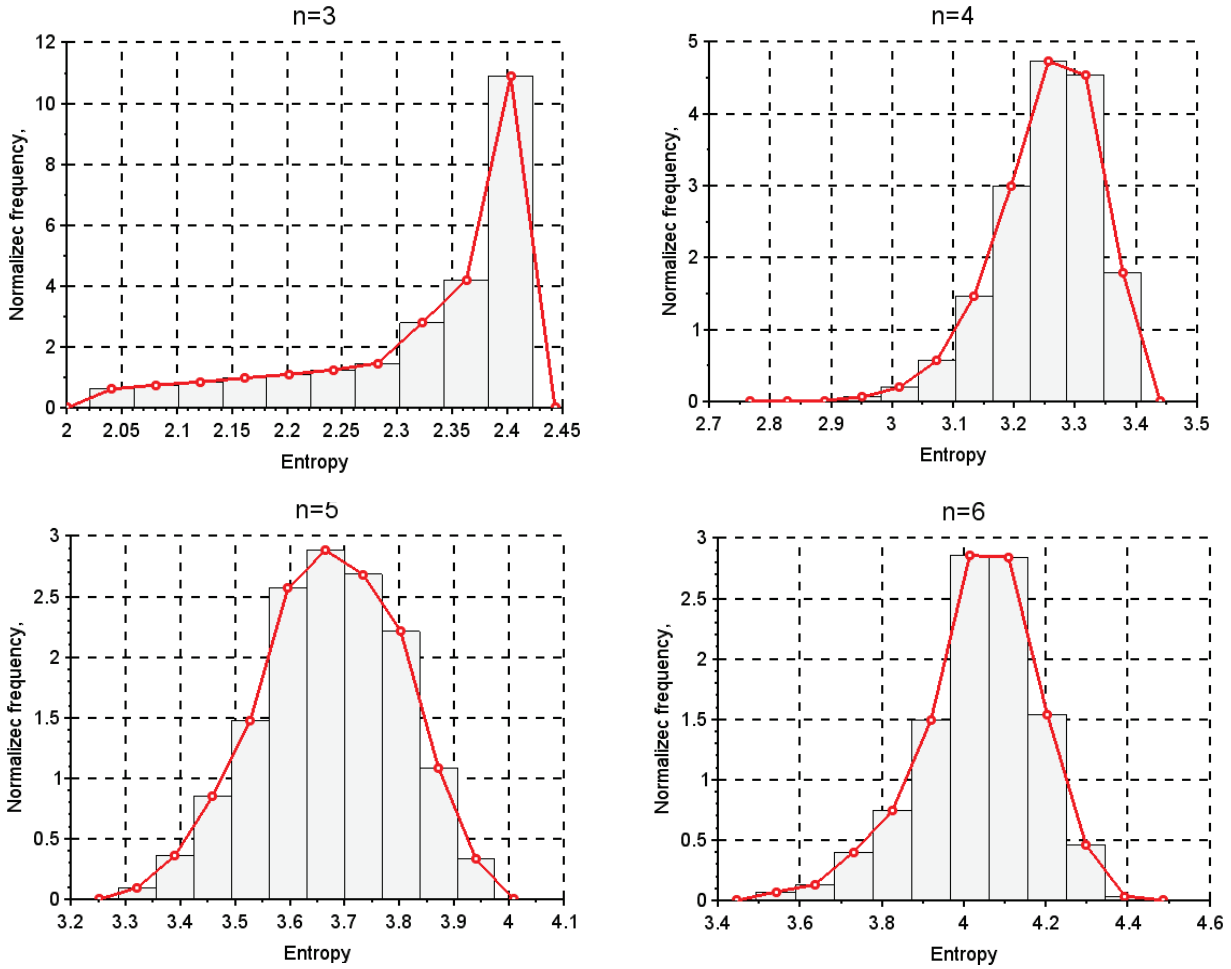


Рис. 5. Гістограми розподілу ентропії для мережі з 3, 4, 5 і 6 вузлами, побудовані на основі множини цілочисельних розв'язків із поділом на 10 інтервалів (табл.1)

Розв'язок оптимізаційної задачі, що оснований на аналітичних виразах (29) і (33) має очевидні обмеження: $\lambda_j \neq 0,\ j \in \{1,2,\dots,s\}$ і $x_i \neq 0,\ j \in \{1,2,\dots,2n-1\}$. Але в загальній постановці задачі (28) нульові розв'язки є допустимими. Тому для більш строгого пошуку розв'язку задачі варто окремо проаналізувати значення ентропії у точках, що є вершинами опуклої множини у *s*-вимірному просторі незалежних змінних $\{\lambda_j\}$, де $\lambda_{t1} = 0$ для деяких $t_1 \in \{1,2,\dots,s\}$ або $x_{t2} = 0$ для деяких $t_2 \in \{1,2,\dots,2n-1\}$.

Дискусія і висновки

Внаслідок проведених досліджень розв'язано такі задачі:

- розроблено алгоритм визначення множини цілочисельних розв'язків задачі (4), які відповідають умовам (1) і (3);
- одержано аналітичні вирази для ентропії, як функції змінних $\{\lambda_j\}$;
- одержано аналітичні вирази для градієнта ентропії, визначеного у просторі ядрових змінних $\{\lambda_j\}$;
- одержано аналітичні вирази для всіх других похідних від ентропії у просторі $\{\lambda_j\}$, які утворюють матрицю Гессе;
- на основі градієнта ентропії і її матриці Гессе розроблено алгоритм визначення стану мережі, в якому її ентропія набуває максимальне значення (Politanskyi et al., 2022).

Перерахуємо переваги розробленого методу. По-перше, це є швидкодія визначення залишкової матриці. Для матриці з n = 10 вузлів, матриця ‖R‖ розмірністю 19×71 обчислюється за допомогою звичайного персонального комп'ютера за час, що не перевищує 0,5 с. По-друге, виконується хороша збіжність ітераційного процесу, яка для досліджених розмірностей (n = 3, 4, 5, 6) практично не залежить від порядку системи, і тому цей процес не потребує значних витрат обчислювальних потужностей.

Очевидною проблемою застосування аналітичного методу є вибір початкового розв'язку задачі ($\vec{\lambda}^{(0)}$ у (36)). Для цього використовуємо алгоритм визначення цілочисельних розв'язків системи, описаний вище. Внаслідок значної кількості вкладених циклів перевірки умов задачі (28), це досить складна задача, яка потребує значних витрат машинного часу навіть для пошуку єдиного розв'язку задачі. Тому визначення початкового розв'язку для оптимізаційної задачі має вирішальне значення для застосування алгоритму у системах, розмірність яких перевищує 7 вузлів.

Внесок авторів: Руслан Політанський – концептуалізація; методологія; Валентин Лесінський – аналіз джерел, підготування огляду літератури; Сергій Галюк – аналіз теоретичних засад дослідження; Андрій Кулько – збір емпіричних даних та їх валідація; емпіричне дослідження.

ISSN 2707-1758

Джерела фінансування. Це дослідження не отримало жодного гранту від фінансової установи в державному, комерційному або некомерційному секторах.

Список використаних джерел

Beshley, M., Toliupa, S., Pashkevych, V., & Kolodiy, R. (2018). Development of software system for network traffic analysis and intrusion detection. *2018 International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2018 – Proceedings*. Igor Sikorsky Kyiv Polytechnic Institute. https://doi.org/10.1109/UkrMiCo43733.2018.9047546

Garofalakis, M. N., Gehrke, J., & Rastogi, R. (Eds.). (2016). *Data stream management – Processing high-speed data streams. Data-Centric Systems and Applications.* Springer. https://doi.org/10.1007/978-3-540-28608-0

Li, X., & Sun, Q. (2021). Identifying and ranking influential nodes in complex networks based on dynamic node strength. *Algorithms,* 14(3), 82. https://doi.org/10.3390/a14030082

Madarro-Capó, E. J., Legón-Pérez, C. M., Rojas, O., & Sosa-Gómez, G. (2021). Information theory based evaluation of the RC4 stream cipher outputs. *Entropy,* 23(7), 896. https://doi.org/10.3390/e23070896

Miletic, S., Pokrajac, I., Pena-Pena, K., Arce, G. R., & Mladenovic, V. A. (2022). Multigraph-defined distribution function in a simulation model of a communication network. *Entropy, 24*(9), 1294. https://doi.org/10.3390/e24091294

Niven, R. K., Abel, M., Schlegel, M., & Waldrip, S. H. (2019). Maximum entropy analysis of flow networks: Theoretical foundation and applications. *Entropy,* 21(8), 776. https://doi.org/10.3390/e21080776

Pedrosa, V. G., & Costa, M. H. M. (2022). Index coding with multiple interpretations. *Entropy,* 24(8), 1149. https://doi.org/10.3390/e24081149

Politanskyi, R. L., Bobalo, Y. Y., Zarytska, O. L., Kiselychyk, M. D., & Vistak, M. V. (2022). Entropy calculation for networks with determined values of flows in nodes. *Mathematical Modelling and Computing,* 9(4), 936–944. https://doi.org/10.23939/mmc2022.04.936

Shahar, A., Alfassi, Y., & Keren, D. (2022). Communication efficient algorithms for bounding and approximating the empirical entropy in distributed systems. *Entropy,* 24(11), 1611. https://doi.org/10.3390/e24111611

Ruslan POLITANSKYI, DSc (Engin.), Prof.
ORCID ID: 0000-0003-0015-7123
e-mail: r.politansky@chnu.edu.ua
Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine

Valentyn LESINSKYI, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0002-1259-1974
e-mail: v.lesinsky@chnu.edu.ua
Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine

Serhii HALIUK, PhD (Engin.)
ORCID ID: 0000-0003-3836-2675
e-mail: s.haliuk@chnu.edu.ua
Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine

Andrii KULKO, PhD Student
ORCID ID: 0009-0006-1185-0774
e-mail: kulko452@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

References

Beshley, M., Toliupa, S., Pashkevych, V., & Kolodiy, R. (2018). Development of software system for network traffic analysis and intrusion detection. *2018 International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2018 – Proceedings*. Igor Sikorsky Kyiv Polytechnic Institute. https://doi.org/10.1109/UkrMiCo43733.2018.9047546

Garofalakis, M. N., Gehrke, J., & Rastogi, R. (Eds.). (2016). *Data stream management – Processing high-speed data streams. Data-Centric Systems and Applications.* Springer. https://doi.org/10.1007/978-3-540-28608-0

Li, X., & Sun, Q. (2021). Identifying and ranking influential nodes in complex networks based on dynamic node strength. *Algorithms,* 14(3), 82. https://doi.org/10.3390/a14030082

Madarro-Capó, E. J., Legón-Pérez, C. M., Rojas, O., & Sosa-Gómez, G. (2021). Information theory based evaluation of the RC4 stream cipher outputs. *Entropy,* 23(7), 896. https://doi.org/10.3390/e23070896

Miletic, S., Pokrajac, I., Pena-Pena, K., Arce, G. R., & Mladenovic, V. A. (2022). Multigraph-defined distribution function in a simulation model of a communication network. *Entropy, 24*(9), 1294. https://doi.org/10.3390/e24091294

Niven, R. K., Abel, M., Schlegel, M., & Waldrip, S. H. (2019). Maximum entropy analysis of flow networks: Theoretical foundation and applications. *Entropy,* 21(8), 776. https://doi.org/10.3390/e21080776

Pedrosa, V. G., & Costa, M. H. M. (2022). Index coding with multiple interpretations. *Entropy,* 24(8), 1149. https://doi.org/10.3390/e24081149

Politanskyi, R. L., Bobalo, Y. Y., Zarytska, O. L., Kiselychyk, M. D., & Vistak, M. V. (2022). Entropy calculation for networks with determined values of flows in nodes. *Mathematical Modelling and Computing,* 9(4), 936–944. https://doi.org/10.23939/mmc2022.04.936

Shahar, A., Alfassi, Y., & Keren, D. (2022). Communication efficient algorithms for bounding and approximating the empirical entropy in distributed systems. *Entropy,* 24(11), 1611. https://doi.org/10.3390/e24111611

Отримано редакцією журналу / Received: 26.03.25
Прорецензовано / Revised: 28.04.25
Схвалено до друку / Accepted: 22.05.25

METHODS FOR DETERMINING THE EQUILIBRIUM STATE OF A NETWORK WITH GIVEN FLOWS IN NODES OF INFORMATION NETWORKS OF CYBER-PHYSICAL SYSTEMS

Background. ***The multidimensionality of cyberspace and the high level of informatization of critical infrastructure objects, both in Ukraine and globally, pose a threat to humanity at the global level in the foreseeable future. The situation is complicated by the fact that critical infrastructure objects, which operate within a unified information space and support a wide range of modern information technologies, remain vulnerable to new types of threats, despite tremendous efforts to counteract external cyber intrusions.***

Methods. ***It has been identified that the study of complex systems employs modern mathematical methods such as topology, graph theory, and linear algebra.***

Results. ***The entropic method for analyzing complex systems has found wide application in practical problems related to the study of natural, technical, and cyber-physical systems. This method has long been used to develop efficient systems for coding, encryption, and information protection. With the increase in computational power, the practical value of researching dynamic systems – including those based on entropy – has grown, as entropy serves as a good indicator for selecting certain states of a system from the set of all its possible states.***

Conclusions. ***Since the study of structures and general patterns is common across many branches of science and technology, tasks of a general nature – where the specific nature of the system is not taken into account – remain relevant. This study presents an algorithm for determining the state of a multi-node information flow network in which its entropy reaches a maximum value. The algorithm is based on analytical expressions for entropy and its first and second derivatives, derived by the authors and defined on the set of kernel solutions of a system of linear equations to which the original problem is reduced. Experimental results are presented for determining the solution set and solving the optimization problem of finding the state with the maximum entropy value for networks with 3, 4, 5, and 6 nodes. Possibilities for further improvement of the method for calculating more complex networks are analyzed.***

Keywords: ***software complex, network traffic, matrices, graphs, entropy, Hessian matrices, submatrix.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2707-1758

Наукове видання

БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ

№ 1(9)/2025

Редактор *Л. Магда*

Оригінал-макет виготовлено Видавничо-поліграфічним центром "Київський університет"
Виконавець *В. Гаркуша*

**Формат 60x84$^{1/16}$. Обл.-вид. арк. 12.9. Ум. друк. арк. 12,8. Наклад 100. Зам. № 225-11523.
Гарнітура Times New Roman. Папір офсетний. Друк офсетний. Вид. № Іт1.
Підписано до друку 29.08.25**

**Видавець і виготовлювач
ВПЦ "Київський університет",
б-р Тараса Шевченка, 14, м. Київ, 01601, Україна
☎ (38044) 239 32 22; (38044) 239 31 58; (38044) 239 31 28
e-mail: vpc@knu.ua; vpc_div.chief@univ.net.ua; redaktor@univ.net.ua
http: vpc.knu.ua
Свідоцтво суб'єкта видавничої справи ДК № 1103 від 31.10.02**