Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна

ВІСНИК

Харківського національного університету імені В.Н. Каразіна

Серія
«Математичне моделювання.
Інформаційні технології.
Автоматизовані системи управління»

Випуск 67

Серія заснована 2003 р.

BULLETIN

of V.N. Karazin Kharkiv National University

Series
«Mathematical Modeling.
Information Technology.
Automated Control Systems»

Issue 67

First published in 2003

Харків
2025

Засновник журналу Харківський національний університет імені В. Н. Каразіна, Харків, Україна. Рік заснування 2003. Періодичність: 4 випуски на рік. https://periodicals.karazin.ua/mia

Статті містять дослідження у галузі математичного моделювання та обчислювальних методів, інформаційних технологій, захисту інформації. Висвітлюються нові математичні методи дослідження та керування фізичними, технічними та інформаційними процесами, дослідження з програмування та комп'ютерного моделювання в наукоємних технологіях.

Для викладачів, наукових працівників, аспірантів, працюючих у відповідних або суміжних напрямках.

Наказом Міністерства освіти і науки України від 17.03.2020 № 409 наукове фахове періодичне видання Вісник Харківського національного університету імені В.Н. Каразіна серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління» включено до Категорії «Б» Переліку наукових фахових видань України за наступними спеціальностями: 113 – Прикладна математика;122 – Комп'ютерні науки та інформаційні технології; 123 – Комп'ютерна інженерія; 125 – Кібербезпека.

Затверджено до друку рішенням Вченої ради Харківського національного університету імені В. Н. Каразіна (протокол № 28 від 27.10.2025 р.)

Адреса редакційної колегії: 61022, м. Харків, майдан Свободи, 6, Харківський національний університет імені В. Н. Каразіна, к. 534.
Тел. +380 (57) 705-42-81, Email: journal-mia@karazin.ua.

Мова публікації: українська, англійська.

Статті пройшли внутрішнє та зовнішнє рецензування.

Ідентифікатор медіа у Реєстрі суб'єктів у сфері медіа: R30-04456
(Рішення № 1538 від 09.05.2024 р Національної ради України з питань телебачення і радіомовлення. Протокол № 15)

The founder of the Journal is V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
Year of foundation 2003. The journal is published four times a year.
https://periodicals.karazin.ua/mia

The articles are present research in the field of mathematical modeling and computing methods, information technologies, information security. New mathematical methods of research and management of physical, technical and information processes, research on programming and computer modeling in science-intensive technologies are covered.

For teachers, researchers, graduate students working in relevant or related fields.

By the order of the Ministry of Education and Science of Ukraine from 17.03.2020 № 409 scientific professional periodical Bulletin of V.N. Karazin Kharkiv National University series "Mathematical modeling. Information Technologies. Automated control systems" is included in Category "B" of the List of scientific professional publications of Ukraine in the following specialties: 113 – Applied Mathematics, 122 – Computer Science and Information Technology; 123 – Computer engineering; 125 – Cybersecurity.

Approved for publication by the decision of the Academic Council of V.N. Karazin Kharkiv National University (Minutes № 28 of 27.10.2025).

Editorial Address: 61022, Kharkiv, Svobodi sq., 6, V.N. Karazin Kharkiv National University, r. 534.
Phone. +380 (57) 705-42-81, Email: journal-mia@karazin.ua.

Language of publication: Ukrainian, English.

The articles pass internal and external review.

Media identifier in the Register of the field of Media Entities: R30-04456
(Decision № 1538 dated May 9, 2024 of the National Council of Television and Radio Broadcasting of Ukraine, Protocol № 15)

ЗМІСТ

CONTENTS

DOI: https://doi.org/10.26565/2304-6201-2025-67-01

УДК (UDC) 004.056 : 004.89

Blinov Maksym *Student V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine e-mail: blinov2020kb12@student.karazin.ua; https://orcid.org/ 0009-0006-2164-3779*

Svatovskiy Igor *Ph.D., Associate Professor V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine e-mail: i.svatowsky@karazin.ua; https://orcid.org/0000-0002-1836-5599*

Analysis of the implementation of the combined Suricata intrusion detection system with a machine learning model

Relevance. The study presents a comparative analysis of intrusion detection and prevention systems (IDS/IPS) functioning with and without artificial intelligence (AI) integration. Conventional signature-based systems such as Suricata effectively detect known threats but often fail to recognize new or modified attack patterns. Therefore, integrating AI technologies offers a promising way to enhance adaptability and minimize false positives.
Objective. The study aimed to evaluate the efficiency of the open-source Suricata system in two configurations: a standard mode using signature-based detection and a modified version enhanced with a machine learning module. The goal was to determine how AI affects detection accuracy, response time, and alert reliability under various cyberattack scenarios, including DoS and brute-force attempts. The experiment was performed in a virtualized environment consisting of three nodes: Kali Linux as the attacker, Windows 10 as the target, and Suricata as the monitoring system.
Research Methods. Methods of statistical modeling and comparative analysis were applied. In its base form, Suricata relied solely on predefined rules, while in the AI-extended version, an analytical module employing the Random Forest algorithm processed log data to classify network events. The model was trained on labeled datasets containing normal and malicious traffic, using extracted statistical and protocol-level features.
Results. Analysis showed that the baseline Suricata achieved a detection rate of 87–92% and precision of 80–85%, generating excessive alerts during DoS simulations. After AI integration, the number of alerts decreased more than threefold, the detection rate increased to 93–96%, and precision rose to 90–94%. Additionally, the average response time was reduced to 1–1.5 seconds.
Conclusions. Integrating machine learning algorithms into the capabilities of Suricata IDS significantly increased its efficiency, reduced the number of false positives, and improved the system's ability to adapt to new cyber threats. The results confirm that combining a signature approach with AI-based analytics provides a more reliable and intelligent approach to modern network security.

***Keywords**: cybersecurity, Intrusion Detection System, Artificial Intelligence, Machine Learning, Suricata, statistical analysis, comparative analysis.*

How to quote: M. Blinov., I. Svatovskiy, “Analysis of the implementation of the combined Suricata intrusion detection system with a machine learning model”, *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 67, pp. 6-17, 2025. https://doi.org/10.26565/2304-6201-2025-67-01

Як цитувати: Blinov M., Svatovskiy I. Analysis of the implementation of the combined Suricata intrusion detection system with a machine learning model. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.6-17. https://doi.org/10.26565/2304-6201-2025-67-01

Introduction

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are now an integral component of modern communication infrastructures operating in the complex and dynamic environment of cyberspace. They provide a significantly higher level of protection than traditional security measures such as antivirus software, spam filters, and standard firewalls. With cyber threats constantly evolving and new forms of attacks emerging, the role of IDS and IPS has grown significantly: these systems have gone from being auxiliary mechanisms to becoming key components of a comprehensive information security system [1, 2, 8].

IDS systems monitor network traffic to detect potentially dangerous actions, security policy violations, or unauthorized access attempts. Their main purpose is to identify suspicious activity and promptly notify the administrator of possible threats. IDS analyzes traffic using different approaches: signature-based, where data is checked against known attack patterns, and behavior-based, where the system detects deviations from normal user or service behavior [6]. Traditional signature-based intrusion detection methods, which rely on established attack models and their signatures, have proven to be insufficiently effective in the face of constantly changing and complex cyber threats [3, 12]. In the second case, artificial intelligence technologies are increasingly being used, which allow the system to learn independently based on previous data and recognize new, previously unknown types of attacks. This ensures high efficiency in countering new methods of cyberattacks that do not have known signatures in databases. However, with the growth in the amount of data, detecting anomalies and malicious behavior on the network is becoming an increasingly difficult task when training machine learning models [3, 11]. Unlike IDS, IPS not only detects threats but also actively responds to them. While IDS is a monitoring system that only warns of suspicious activity, IPS is an active defense system capable of automatically blocking dangerous traffic, interrupting connections, or changing data routing to prevent damage. Thus, IPS can be considered the next stage in the development of IDS, as it combines analysis and response capabilities in a single solution.

Both systems are often integrated into the overall cyber security architecture of an organization [1]. IDS is usually located "out of band" — that is, it processes a copy of the traffic without affecting its transmission speed. This avoids delays and ensures network continuity even under heavy load. IPS, on the other hand, is installed directly in the traffic flow — in the "gap" between network segments, allowing it to actively intervene in the data transfer process, filter malicious packets, and prevent attacks from penetrating. However, this location has its drawbacks — in case of overload or malfunction, IPS can become a bottleneck in the system, affecting the overall network throughput [4].

One of the main advantages of IDS/IPS is their ability to work in conjunction with other security measures, such as firewalls, access control systems, or antivirus solutions. In modern network architectures, IPS is often integrated into a next-generation firewall (NGFW), creating a single platform that simultaneously performs filtering, monitoring, and attack prevention functions [5, 6]. This allows you to increase the level of protection for your organization and minimize response time to security incidents.

Modern IDS/IPS systems also support real-time threat detection. They collect traffic data, analyze user behavior, check access to external resources, and track abnormal activity. For example, the system can detect attempts to connect to botnet command centers, unauthorized requests to external IP addresses, or suspicious activity within the corporate network [5]. If a security policy violation is detected, the system generates a message for the administrator or automatically applies appropriate measures—blocks the source of the threat, isolates the network segment, or activates other security mechanisms.

Thus, IDS and IPS perform complementary functions in ensuring information security. IDS focuses on detailed traffic analysis and identifying potential risks, while IPS not only detects but also actively counteracts attacks. Together, they create a multi-layered protection system that allows an organization to respond to cyber threats in a timely manner, reduce the risk of information leakage, and maintain the stability of the network infrastructure [6]. In a world where the number and complexity of attacks are growing daily, the use of IDS/IPS is a prerequisite for building a reliable cyber defense system for any modern organization.

In machine learning models, the goal is to create an implicit or explicit model. Although they are resource-intensive by nature, such schemes can change their execution strategy as new details are obtained. A hybrid methodology works with a combination of two or more methodologies, allowing the strengths of each individual methodology to be leveraged. For example, when an anomaly-based mechanism for data filtering is combined with a signature-based mechanism that detects intrusions, the result is a hybrid detection system [2].

A distinctive feature of IDS/IPS in modern wireless networks is the need to use hybrid threat detection methodologies. This is due, among other things, to the impact on network traffic of natural and artificial

interference factors that are inevitably present in their radio channels and can significantly degrade the signal-to-noise ratio. However, anomaly detection systems produce a high percentage of false positives, since even statistically normal events can be mistakenly identified as anomalies [2]. For example, even strong natural fluctuations in the level of radio signals in the network can be perceived by the intrusion detection system as a denial-of-service attack.

Such features make it relevant to develop combined intrusion detection systems based on signature and behavioral approaches that use effective artificial intelligence methods for use in modern wireless networks.

1. Building a machine learning model

Despite the fact that traditional signature-based systems have difficulty detecting unknown types of attacks, their advantage is their potentially high performance. This advantage determines the need to use signature-based IDS/IPS in high-speed wireless networks. Adding the ability to assess abnormal system behavior for event classification can significantly improve the overall effectiveness of the system [4, 10, 11]. The open source Suricata system from the Open Information Security Foundation was used as the signature system. It was combined with a machine learning model to provide additional event classification. These systems are very good at detecting both common and anomalous threats because they use complex algorithms for self-learning and adaptation based on the evaluation of network performance parameters.

To build an artificial intelligence model that filters IDS alerts, an experimental network was used in which Suricata generated a stream of events in eve.json format [5]. At the same time, the attacks_schedule.log file was kept, where the time limits for performing test attacks were recorded. This data was combined into a single source of truth, where each record from eve.json was labeled "1" if its time fell within the attack window, or "0" if it belonged to normal traffic. This approach ensured automatic and reproducible data labeling without the need for manual classification.

After that, feature extraction was performed—a set of parameters characterizing traffic behavior was formed from each alert and related flows. These features included the signature ID, threat level, source and destination ports, protocol, frequency of alerts from a single source over a given period of time, number of unique ports, session duration, time of day, and private IP address usage. A simple mechanism for counting previous alerts in a sliding time window was implemented, which made it possible to obtain basic but informative features for training.

The model was trained offline. Based on the analysis of the capabilities and practicality of implementing machine learning algorithms [2, 3, 9-12] to solve the task at hand, the Random Forest algorithm was chosen, which combines high prediction accuracy, the ability to process different types of data, and clear interpretation of results [9]. The input dataset was divided into training and test samples while maintaining the ratio between positive and negative examples. After training, the metrics of accuracy, completeness, F1-measure, and confusion matrix were evaluated. The main focus was on achieving a high level of recall (so as not to miss real attacks) while reducing the number of false positives (precision).

After validation, the model was saved as a file (rf_model.pkl) and integrated into Suricata as an additional post-processing layer. In the simplest scenario, the model worked offline: after the traffic collection session was completed, the alerts were exported to a CSV file, passed through a feature generation pipeline, and the classification results were stored as a filtered set (filtered_alerts.csv) containing only the most relevant events. In another application, a real-time script was used to monitor the eve.json stream, build features for each new record, and predict its value using the model. If an anomaly was detected, the alert was forwarded to the monitoring system or SIEM, while normal events were marked as insignificant or filtered out [10].

In practice, this model significantly reduced the number of noise alerts generated by Suricata, allowing analysts to focus on truly important incidents. Thanks to the use of real-world log training, the model learned to recognize characteristic attack patterns and respond to atypical activity even when standard IDS signatures failed. Thus, the system demonstrated its ability to adapt to new types of threats and increased the overall accuracy of their detection. Our study used the logical diagram of the artificial intelligence model training algorithm shown in Fig. 1. First, several types of attacks were carried out, the data for which are presented in Table 1. The traffic generated by these attacks was subsequently processed

by the Suricata IDS system and saved in the eve.json file. The model was trained based on the resulting file and on typical sets of Internet traffic from GitHub.

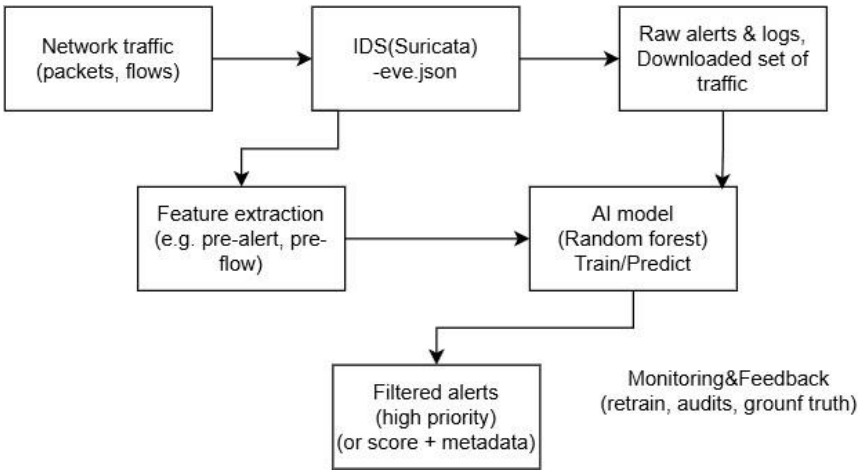


Fig. 1 Diagram of the IDS model training process based on network traffic
Рис. 1. Схема процесу навчання моделі IDS на основі мережевого трафіку

At the same time, the limitations of the approach were taken into account. The model remained sensitive to the specifics of the environment in which the training took place and required periodic retraining on updated data. Therefore, in production scenarios, it was considered an auxiliary intelligent filter rather than the sole source of truth. All model decisions were logged for further audit, which allowed tracking its performance, identifying classification errors, and improving the algorithm. This architecture provided a balance between detection accuracy, resistance to new attacks, and practical usability in cyber defense systems.

2. Comparative analysis of systems

The experiment was performed in a virtualized environment consisting of three nodes: Kali Linux as the attacker with IP address 172.22.254.171, Windows 10 as the target with IP address 172.22.242.178, and Suricata as the monitoring system with IP address 172.22.251.181.

As part of this experiment, the effectiveness of the intrusion detection system (IDS Suricata) was evaluated during the detection of attacks in a controlled environment. The aim of the study was to determine the system's ability to identify network threats, evaluate the accuracy of responses, response speed, and the level of false alarms.

Three main data sources were used for this purpose:

1. A PCAP file recording all network traffic during the experiment;
2. The eve.json file containing the IDS Suricata alert log;
3. A file with the time intervals of the attacks, which serves as a "ground truth" for comparison.

2.1. Methodology for evaluating intrusion detection systems

The effectiveness of the intrusion detection system was evaluated by comparing the performance of IDS Suricata in its basic configuration and after integrating the artificial intelligence module. The methodology involved a comprehensive test of the system in a controlled network environment [4]. At the same time, a qualitative analysis of the results was carried out, which involved building a timeline of IDS triggers, analyzing the dynamics of the system's response, and identifying the types of attacks that were successfully detected or missed. For objectivity of comparison, data from IDS without AI and with an integrated machine learning model were analyzed under the same conditions and using the same attack scenarios.

The methodology also included evaluating the effectiveness of the artificial intelligence model used to filter alerts. To do this, we used classification results obtained using the Random Forest algorithm, which was trained on previous Suricata logs with events labeled as "attack" or "normal activity." Comparing the

results of the two systems made it possible to determine the impact of AI on reducing the number of false positives, improving attack detection accuracy, and reducing response time.

The following key metrics were used to evaluate the effectiveness of the IDS [4, 8]:

- False Positives (FP) — the number of alerts that did not correspond to any real attack;
- False Negatives (FN) — the number of attacks that the IDS did not detect;
- Detection latency — the average time between the start of an attack and the moment the first alert about it was generated.

2.2. Modeling analysis results

For the analysis, several test attacks were carried out from the Kali Linux attack machine on a Windows system protected by IDS.

Figure 2 shows an example of IDS Suricata logs from the eve.json file. As can be seen from the figure, there are several warnings about LOCAL Possible volumetric spike (many flows) and LOCAL UDP volumetric spike to DNS port, indicating a possible DOS attack.


```
{"timestamp":"2025-10-12T20:47:39.615661+0300","src_ip":"172.22.251.181","dest_ip":"172.22.240
.1","event_type":"alert","alert":"LOCAL UDP volumetric spike to DNS port"}
{"timestamp":"2025-10-12T20:47:39.615661+0300","src_ip":"172.22.251.181","dest_ip":"172.22.240
.1","event_type":"dns","alert":null}
{"timestamp":"2025-10-12T20:47:39.632377+0300","src_ip":"172.22.240.1","dest_ip":"172.22.251.1
81","event_type":"dns","alert":null}
{"timestamp":"2025-10-12T20:47:39.633261+0300","src_ip":"172.22.251.181","dest_ip":"91.189.91.
49","event_type":"alert","alert":"LOCAL Possible volumetric spike (many flows)"}
{"timestamp":"2025-10-12T20:47:39.856682+0300","src_ip":"172.22.251.181","dest_ip":"91.189.91.
49","event_type":"http","alert":null}
{"timestamp":"2025-10-12T20:47:40.876217+0300","src_ip":"172.22.251.181","dest_ip":"172.22.240
.1","event_type":"flow","alert":null}
{"timestamp":"2025-10-12T20:47:41.824497+0300","src_ip":"172.22.251.181","dest_ip":"172.22.240
.1","event_type":"alert","alert":"LOCAL UDP volumetric spike to DNS port"}
```


Fig. 2 Example of IDS Suricata traffic filtering
Рис. 2. Приклад фільтрації трафіку системою IDS Suricata

Figure 3 shows the exact start and end times of the attacks. This data can be used to analyze the IDS in detail.

```
maksym@maksym-Virtual-Machine:~/stats2$ cat attacks_schedule.log
2025-10-14T16:30:54+0000 START fast_nmap_portscan
2025-10-14T16:31:12+0000 END fast_nmap_portscan
2025-10-14T16:32:00+0000 START ssh_bruteforce
2025-10-14T16:32:14+0000 END ssh_bruteforce
2025-10-14T16:32:41+0000 START rdp_bruteforce
2025-10-14T16:32:51+0000 END rdp_bruteforce
2025-10-14T16:33:19+0000 START SQL_like
2025-10-14T16:33:33+0000 END SQL_like
2025-10-14T16:34:30+0000 START DOS
2025-10-14T16:34:45+0000 END DOS
2025-10-14T16:35:19+0000 START slow_nmap_portscan
2025-10-14T16:43:46+0000 END slow_nmap_portscan
```

Fig. 3 Timelines of attack start and end
Рис. 3. Хронологія початку та завершення атаки

Tables 1 and 2 present the statistics of the analysis of the pcap file, which stores traffic during testing, the eve.json file with IDS Suricata filtering, and the attacks_schedule.csv file, which specifies the start and end times of the attacks.

Table 1. Statistics on attacks detected by the Suricata IDS system
Таблиця 1. Статистика атак, виявлених системою IDS Suricata

Types of attacks	**Number of responses, first wave of attacks**	**Number of responses, second wave of attacks**	**Number of responses, third wave of attacks**
Fast port scan	2	6	3
Slow port scan	84	57	43
SSH brute force	1	1	1
RDP Brute-force	1	1	1
HTTP attacks SQLi-like (curl)	21	30	22
DoS / volumetric spike	287119	177623	93437

Table 2. Most common IDS Suricata security alerts
Таблиця 2. Найпоширеніші сповіщення безпеки системи IDS Suricata

Message	**Number of alerts during the first wave of attacks**	**Number of alerts during the second wave of attacks**	**Number of alerts during the third wave of attacks**
SURICATA STREAM 3way handshake excessive different SYNs	274862	169774	8998
SURICATA STREAM Packet with invalid ack	607	4559	1727
SURICATA STREAM SHUTDOWN RST invalid ack	6074	4559	1727
LOCAL Possible volumetric spike (many flows)	75	72	67
LOCAL UDP volumetric spike to DNS port	63	51	39
ET INFO Python BaseHTTP ServerBanner	23	21	22
LOCAL Fast Portscan (many SYNs)	3	3	2
LOCAL RDP Brute Force - multiple attempts	1	1	1
LOCAL HTTP SQLi-like pattern	1	1	1

Analysis of the results showed that IDS Suricata demonstrated a high level of sensitivity, detecting most of the attacks carried out during the experiment. The detection rate (recall) was approximately 87–92%, indicating effective recognition of most threats. Precision was at 80–85%, which means that a significant portion of the generated alerts actually corresponded to real attacks, although there was a moderate level of false positives. A small number of False Negatives were detected, i.e., attacks that were not recorded by the IDS — mostly slow or inconspicuous port scans, as well as some types of ICMP activity. The average response time (Detection latency) was about 2–3 seconds after the start of the attack activity, which is a good indicator for real- time systems. Analysis of the event timeline showed a clear

correlation between the start of attacks (according to ground truth) and the appearance of notifications in the eve.json log, confirming the correctness of synchronization and the effectiveness of detection rules.

Next, the effectiveness of the AI-based IDS model was tested. The same set of attacks was used for testing as for IDS Suricata. Tables 3 and 4 present statistics from the analysis of the pcap file that stores traffic during testing, filtered_alerts.csv, which contains security alerts, and the attacks_schedule.csv file, which specifies the start and end times of the attacks.

Figure 4 shows the contents of the filtered_alerts.csv file. This file contains security alerts about attacks detected by artificial intelligence.


```
{"timestamp":"2025-10-12T17:50:11.123456+00:00","src_ip":"172.22.251.181","dest_ip":"172.22.24
0.1","event_type":"alert","alert":"SURICATA STREAM 3way handshake excessive different SYNs"}
{"timestamp":"2025-10-12T17:50:11.223789+00:00","src_ip":"172.22.251.181","dest_ip":"172.22.24
0.1","event_type":"alert","alert":"SURICATA STREAM Packet with invalid ack"}
{"timestamp":"2025-10-12T17:50:12.001234+00:00","src_ip":"172.22.251.182","dest_ip":"172.22.24
0.2","event_type":"alert","alert":"SURICATA STREAM SHUTDOWN RST invalid ack"}
{"timestamp":"2025-10-12T17:50:13.450000+00:00","src_ip":"172.22.251.183","dest_ip":"172.22.24
0.3","event_type":"alert","alert":"LOCAL Possible volumetric spike (many flows)"}
{"timestamp":"2025-10-12T17:50:13.460000+00:00","src_ip":"172.22.251.184","dest_ip":"8.8.8.8",
"event_type":"alert","alert":"LOCAL UDP volumetric spike to DNS port"}
{"timestamp":"2025-10-12T17:50:14.005678+00:00","src_ip":"172.22.251.185","dest_ip":"172.22.24
0.4","event_type":"alert","alert":"ET INFO Python BaseHTTP ServerBanner"}
{"timestamp":"2025-10-12T17:50:15.999999+00:00","src_ip":"172.22.251.181","dest_ip":"172.22.24
0.1","event_type":"alert","alert":"LOCAL Fast Portscan (many SYNs)"}
{"timestamp":"2025-10-12T17:50:20.111111+00:00","src_ip":"172.22.251.190","dest_ip":"172.22.24
0.5","event_type":"alert","alert":"LOCAL RDP Brute Force - multiple attempts"}
{"timestamp":"2025-10-12T17:50:22.250000+00:00","src_ip":"172.22.251.192","dest_ip":"172.22.24
0.6","event_type":"alert","alert":"ET WEB_SERVER Possible SQL Injection Attempt"}
```


Fig. 4 Filtered set of security alerts
Рис. 4. Відфільтрований набір сповіщень безпеки

Table 3. Statistics on attack detection by the artificial intelligence model
Таблиця 3. Статистика виявлення атак моделлю штучного інтелекту

Types of attacks	Number of responses, first wave of attacks	Number of responses, second wave of attacks	Number of responses, third wave of attacks
Fast port scan	2	5	3
Slow port scan	75	52	41
SSH Brute-force	2	2	1
RDP Brute-force	1	1	1
HTTP attacks SQLi-like (curl)	20	27	21
DoS / volumetric spike	65	41,678	22,953

As part of the study, artificial intelligence was integrated into IDS Suricata as an additional event post-processing module. In standard mode, Suricata generated logs in eve.json format, which contained information about all recorded events on the network. This data was transferred to a machine learning module that had been pre-trained on real traffic sets containing both normal connections and various types of attacks. The model analyzed each record, evaluating it based on a set of behavioral characteristics—protocol type, request frequency, number of unique ports, IP activity, connection intensity, etc.—and classified the event as potentially dangerous or safe.

In this way, artificial intelligence acted as a filter that complemented Suricata's signature logic, weeding out repetitive or insignificant alerts and retaining only those that had a high probability of being a real threat. The processed results were stored in a separate file with "cleaned" alerts, which greatly

simplified further analysis. This integration made it possible to reduce the volume of uninformative messages, increase detection accuracy, and ensure more stable operation of the security system in conditions of high event volume.

Table 4. The most common security alerts
Таблиця 4. Найпоширеніші сповіщення безпеки

Message	Number of alerts during the first wave of attacks	Number of alerts during the second wave of attacks	Number of alerts during the third wave of attacks
SURICATA STREAM 3way handshake excessive different SYNs	63,812	39,247	21,684
SURICATA STREAM Packet with invalid ack	2,541	1,923	796
SURICATA STREAM SHUTDOWN RST invalid ack	2,472	1,885	755
LOCAL Possible volumetric spike (many flows)	4	46	43
LOCAL UDP volumetric spike to DNS port	37	33	28
ET INFO Python BaseHTTP ServerBanner	22	20	19
LOCAL Fast Portscan (many SYNs)	6	5	4
LOCAL RDP Brute Force – multiple attempts	5	3	4

After integrating the artificial intelligence module (a machine learning model trained on previous eve.json logs), automatic filtering of alerts coming from Suricata was implemented. The AI classified each alert as likely true or likely false positive. As a result, the total number of responses decreased, especially for attack types that often generate a large number of secondary or redundant alerts (e.g., DoS attacks).

The implementation of this intelligent filtering mechanism significantly improved the clarity and manageability of the monitoring process. Instead of overwhelming the analyst with thousands of repetitive or low-priority notifications, the system prioritized alerts with a high probability of being genuine threats. This optimization not only reduced the operator's workload but also allowed for faster incident response and more efficient allocation of computational resources. The AI continuously refined its classification accuracy by learning from feedback on previous decisions, ensuring that its filtering process adapted to new patterns of network behavior and evolving attack techniques. Over time, this dynamic improvement contributed to a noticeable increase in both detection accuracy and operational stability of the IDS.

Below are the results of the impact of the machine learning module on the Suricata IDS effectiveness by simulated attacks types:

1. DoS / volumetric spike.

The largest reduction in false positives (3–4 times). This is because during a massive attack, Suricata registers thousands of similar packets that do not carry additional information. The AI model has learned to recognize repetitive signatures and filter them as "noise," leaving only representative alerts. The result is a reduction in the number of false positives, while real incidents remain recorded;

2. Slow port scan.

The number of alerts has decreased slightly, but not significantly — AI has retained most of the records, as this type of attack is low-intensity and requires careful analysis. The model has learned to

distinguish real scans from safe background traffic, which has improved accuracy but has not radically reduced the number of records;

3. Fast port scan.

The results remained almost unchanged. Suricata signatures work quite accurately for fast scanning, and the AI did not filter most of them. Only a few alerts were marked as duplicates or uninformative;

4. SSH/RDP brute force.

The number of responses remained stable or increased slightly. This is because the AI was able to identify events that Suricata could perceive as "normal" activity, but which resembled password guessing attempts based on behavioral patterns. Thus, intelligent processing increased sensitivity to subtle attacks;

5. HTTP SQLi-like (curl).

A slight decrease in the number of alerts. The model learned to distinguish between "test" queries and real SQL injections, avoiding duplication of events caused by repeated queries in different sessions. The number of notifications has been significantly reduced, but without losing key attack indicators.

This was achieved because the AI analyzed patterns in traffic behavior and rejected duplicates or low-information events typical of overloaded streams. As a result, after the implementation of artificial intelligence, the number of general alerts decreased more than threefold, particularly for high-volume or noise events.

The results of experiments using artificial intelligence showed that IDS Suricata with a built-in machine learning model demonstrates a noticeable improvement in accuracy and stability. The detection rate (recall) increased to 93–96%, and precision to 90–94%, indicating a reduction in false positives and an improvement in the ability to detect even hidden or atypical attacks.

Thanks to behavioral analysis of network traffic, the system more effectively detected slow port scans, password guessing attempts, and HTTP requests with SQLi-like characteristics that might have gone unnoticed before. The average response time decreased to 1–1.5 seconds, allowing the system to respond in near real time.

The number of duplicate or redundant alerts was reduced by approximately 40%, and the structure of the eve.json log became more organized. Overall, the integration of AI improved Suricata's performance, providing more accurate threat detection, faster response, and reduced system load without compromising security control quality.

The main effect of combining signature-based IDS and attack classification using the Random Forest algorithm is that the system no longer gets "bogged down" in a large number of uninformative logs and leaves only those events that have real analytical value. This allows you to:

- Reduce the load on security analysts (SOC);
- Reduce the number of false positives;
- Maintain high accuracy in detecting real incidents;
- Track the actual dynamics of attacks while filtering out statistical noise.

3. Conclusions

As a result of experimental research, a comparative analysis was conducted of the operation of the Suricata combined intrusion detection system in two modes: without the use of artificial intelligence and with the connection of a trained machine learning model. The goal was to determine how the integration of intelligent data processing methods affects the accuracy, speed, and stability of the system's response to various types of attacks.

The results showed that in the basic configuration without AI, Suricata generates a large number of alerts, a significant portion of which are duplicates or insignificant. This leads to system overload with messages, especially during high-frequency DoS attacks, where more than 200,000 alerts are recorded. The integration of an artificial intelligence model has significantly reduced the number of such alerts through automatic event classification and filtering of duplicate or insignificant records.

In the course of the experiment, additional tests were conducted to evaluate the system’s adaptability to changing attack vectors. The AI-augmented Suricata successfully recognized modified payloads and traffic anomalies that differed from the training dataset, demonstrating its ability to generalize and detect zero-day threats. Moreover, the model’s continuous learning mechanism allowed it to refine its

classification accuracy over time, maintaining stability even under increased network load. This adaptability is especially valuable for dynamic environments where new threats emerge rapidly and traditional rule-based systems struggle to keep up.

The AI model analyzed behavioral characteristics of network traffic, such as packet frequency, connection sequence, number of session establishment attempts, and signs of typical attack patterns. As a result, the system learned to more accurately distinguish normal activity from suspicious activity. For slow port scan attacks, the number of detections decreased by almost 40%, indicating a reduction in false positives. At the same time, for SSH Brute-force and RDP Brute-force attacks, the model was able to detect attempts even when the traditional signature-based system did not record events due to the insignificant number of requests.

Overall, the results of the research showed that the additional use of AI improves the quality of network traffic analytics, makes the combined IDS system adaptive to new attack scenarios, and suitable for use in high-speed wireless networks. Reducing the number of duplicate messages and noise allows the operator to focus on truly critical threats. In addition, real-time data processing with machine learning provides dynamic improvement in attack detection through continuous model updates.

Thus, we can conclude that combining traditional IDS mechanisms with artificial intelligence technologies significantly increases of the protection system effectiveness, reduces the number of false positives, and allows for a more rapid response to modern cyber threats.

СПИСОК ЛІТЕРАТУРИ

1. J. Green. Security Architecture: A Practical Guide to Designing Proactive and Resilient Cyber Protection. BCS, The Chartered Institute for IT, 2025. 358 p. URL: https://www.perlego.com/book/4905875/security-architecture-a-practical-guide-to-designing-proactive-and-resilient-cyber-protection-pdf
2. Wireless Communication Security (Advances in Data Engineering and Machine Learning) / edited by Manju Khari et al. Wiley-Scrivener, 2023. 288 p. URL: https://dokumen.pub/wireless-communication-security-advances-in-data-engineering-and-machine-learning-9781119777144-1119777143.html
3. Talukder, M.A., Islam, M.M., Uddin, M.A. et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. Journal of Big Data, 11, 33 (2024). DOI: https://doi.org/10.1186/s40537-024-00886-w
4. Тимощук, В., Ванца, В., Карнаухов, А., Орловська, А., Тимощук, Д. (2024). Порівняльний аналіз підходів до виявлення вторгнень, заснованих на сигнатурах та аномаліях. Матеріали конференції MCND (29 листопада 2024 р.; Житомир, Україна), с. 328–332. URL: https://scholar.google.com/citations?view_op=view_citation&hl=uk&user=sIhfAOgAAAAJ&citation_for_view=sIhfAOgAAAAJ:QIV2ME_5wuYC
5. Thomas L. Case. Enterprise Networks: Infrastructure & Security. Prospect Press, 2025. 558 p. URL:https://books.google.de/books/about/Enterprise_Network_Infrastructure_Securi.html?id=DVMN0AEACAAJ&redir_esc=y
6. Joseph Migga Kizza. Guide to Computer Network Security. Springer Nature Switzerland AG, 2024. 646 p. URL: https://link.springer.com/book/10.1007/978-3-031-47549-8
7. Тимощук, Д., Ясний, О., Митник, М., Загородна, Н., Тимощук, В. (2024). Виявлення та класифікація DDoS-атак методами машинного навчання. CEUR Workshop Proceedings, 3842, с. 184–195. URL: https://ceur-ws.org/Vol-3842/paper11.pdf
8. M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita. Network Traffic Anomaly Detection and Prevention: Concepts, Techniques and Tools. Springer International Publishing AG, 2017. 263 p. URL:https://www.researchgate.net/publication/321502082_Network_Traffic_Anomaly_Detection_and_Prevention_Concepts_Techniques_and_Tools
9. H. A. Salman, A. Kalakech, & A. Steiti. Random Forest Algorithm Overview. Babylonian Journal of Machine Learning, 2024, pp. 69–79. DOI: https://doi.org/10.58496/BJML/2024/007
10. Ahmed, U., Nazir, M., Sarwar, A. et al. Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Scientific Reports, 15, 1726 (2025). DOI: https://doi.org/10.1038/s41598-025-85866-7

11. Parag Deoskar, Ajay Kumar Sachan. Enhancing intrusion detection systems using hybrid deep learning models. International Journal of Cloud Computing and Database Management, 6(1):29–42. DOI: https://doi.org/10.33545/27075907.2025.v6.i1a.82
12. S. A. H. Moamin, M. K. Abdulhameed, R. M. Al-Amri, A. D. Radhi, R. K. Naser, & L. G. Pheng. Artificial Intelligence in Malware and Network Intrusion Detection: A Comprehensive Survey of Techniques, Datasets, Challenges, and Future Directions. Babylonian Journal of Artificial Intelligence, 2025, pp. 77–98. DOI: https://doi.org/10.58496/BJAI/2025/008

REFERENCES

1. J. Green Security Architecture: A practical guide to designing proactive and resilient cyber protection. BCS, The Chartered Institute for IT, 2025. 358 p. URL: https://www.perlego.com/book/4905875/security-architecture-a-practical-guide-to-designing-proactive-and-resilient-cyber-protection-pdf
2. Wireless Communication Security (Advances in Data Engineering and Machine Learning)/ by Manju Khari (Editor) & more. Wiley-Scrivener, 2023. 288 p.
3. Talukder, M.A., Islam, M.M., Uddin, M.A. et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. J Big Data 11, 33 (2024).DOI: https://doi.org/10.1186/s40537-024-00886-w
4. Tymoshchuk, V., Vantsa, V., Karnaukhov, A., Orlovska, A., & Tymoshchuk, D. (2024). Comparative analysis of intrusion detection approaches based on signatures and anomalies. Proceedings of the MCND Conference (November 29, 2024; Zhytomyr, Ukraine), 328–332. URL: https://scholar.google.com/citations?view_op=view_citation&hl=uk&user=sIhfAOgAAAAJ&citation_for_view=sIhfAOgAAAAJ:QIV2ME_5wuYC [in Ukrainian]
5. Thomas L. Case Enterprise Networks: Infrastructure & Security. Prospect Press, 2025. 558 p. URL:https://books.google.de/books/about/Enterprise_Network_Infrastructure_Securi.html?id=DVMN0AEACAAJ&redir_esc=y
6. Joseph Migga Kizza Guide to Computer Network Security. Springer Nature Switzerland AG, 2024. 646 p. URL: https://link.springer.com/book/10.1007/978-3-031-47549-8
7. Tymoshchuk, D., Yasniy, O., Mytnyk, M., Zagorodna, N., Tymoshchuk, V., (2024). Detection and classification of DDoS flooding attacks by machine learning methods. CEUR Workshop Proceedings, 3842, pp. 184 - 195. URL: https://ceur-ws.org/Vol-3842/paper11.pdf [in Ukrainian]
8. M.H. Bhuyan, D. K. Bhattacharyya, J. K. Kalita Network Traffic Anomaly Detection and Prevention. Springer International Publishing AG, 2017. 263 p. URL: https://www.researchgate.net/publication/321502082_Network_Traffic_Anomaly_Detection_and_Prevention_Concepts_Techniques_and_Tools
9. Random Forest Algorithm Overview (H. A. Salman, A. Kalakech, & A. Steiti , Trans.). (2024). Babylonian Journal of Machine Learning, 2024, 69-79. DOI: https://doi.org/10.58496/BJML/2024/007
10. Ahmed, U., Nazir, M., Sarwar, A. et al. Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Sci Rep 15, 1726 (2025). DOI: https://doi.org/10.1038/s41598-025-85866-7
11. Parag Deoskar and Ajay Kumar Sachan Enhancing intrusion detection systems using hybrid deep learning models. International Journal of Cloud Computing and Database Management 6(1):29-42. DOI: 10.33545/27075907.2025.v6.i1a.82
12. Artificial Intelligence in Malware and Network Intrusion Detection: A Comprehensive Survey of Techniques, Datasets, Challenges, and Future Directions (S. A. H. . Moamin, M. K. . Abdulhameed, R. M. . Al-Amri, A. D. . Radhi, R. K. . Naser, & L. G. . Pheng , Trans.). (2025). Babylonian Journal of Artificial Intelligence, 2025, 77-98. DOI: https://doi.org/10.58496/BJAI/2025/008

Блінов Максим Олександрович *студент*
Харківський національний університет ім. В. Н. Каразіна
майдан Свободи 4, 61022, Харків
e-mail: blinov2020kb12@student.karazin.ua;
https://orcid.org/0009-0006-2164-3779

Сватовський Ігор Іванович *к.т.н., доцент*
Харківський національний університет ім. В.Н. Каразіна
майдан Свободи 4, 61022, Харків
e-mail: i.svatowsky@karazin.ua;
https://orcid.org/0000-0002-1836-5599

Аналіз реалізації комбінованої системи виявлення вторгнень Suricata з моделлю машинного навчання

Актуальність. У дослідженні представлено порівняльний аналіз роботи систем виявлення та запобігання вторгненням (IDS/IPS), які функціонують із використанням та без використання технологій штучного інтелекту (ШІ). Традиційні системи, засновані на сигнатурному підході, такі як Suricata, ефективно виявляють відомі загрози, однак часто не здатні розпізнавати нові або модифіковані типи атак. Тому інтеграція технологій ШІ є перспективним напрямом для підвищення адаптивності системи та зменшення кількості хибнопозитивних спрацювань.
Мета дослідження. Метою роботи була оцінка ефективність відкритої системи IDS Suricata у двох конфігураціях: стандартному режимі з використанням сигнатурного виявлення та у модифікованій версії, доповненій модулем машинного навчання. Завданням було визначити, як саме застосування ШІ впливає на точність виявлення, час реагування та достовірність сповіщень за різних сценаріїв кібератак, зокрема DoS та brute-force. Експеримент проводився у віртуалізованому середовищі, що складалось з трьох вузлів: Kali Linux (зловмисник), Windows 10 (цільова машина) та Suricata (система моніторингу).
Методи дослідження. Застосовано методи статистичного моделювання та порівняльного аналізу. У базовій версії Suricata використовувала лише заздалегідь визначені правила, тоді як у варіанті з ШІ аналітичний модуль із застосуванням алгоритму Random Forest обробляв журнали подій для класифікації мережевої активності. Модель навчалась на розмічених наборах даних, що містили нормальний та шкідливий трафік, із використанням статистичних і протокольних ознак.
Результати. Аналіз показав, що базова версія Suricata забезпечила рівень виявлення 87–92% і точність 80–85%, при цьому генерувала надлишкову кількість сповіщень під час DoS-атак. Після інтеграції ШІ кількість сповіщень зменшилася більш ніж утричі, рівень виявлення зріс до 93–96%, а точність — до 90–94%. Середній час реагування скоротився до 1–1,5 секунди.
Висновки. Інтеграція алгоритмів машинного навчання до можливостей IDS Suricata суттєво підвищила ефективність її роботи, зменшила кількість хибних спрацьовувань і покращила здатність системи адаптуватись до нових кіберзагроз. Отримані результати підтверджують, що поєднання сигнатурного підходу з аналітикою на основі ШІ забезпечує більш надійний і розумний підхід до сучасної мережевої безпеки.

Ключові слова: *кібербезпека, система виявлення вторгнень, штучний інтелект, машинне навчання, Suricata, статистичний аналіз, порівняльний аналіз.*

Надійшла у першій редакції 11.08.2025, в останній - 15.09.2025.
The first version has been received on 11.08.2025, the final version – on 15.09.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-02

УДК (UDC) 004.4

Гаврилюк Єгор Андрійович
аспірант кафедри математичного моделювання та аналізу даних
Харківський національний університет імені В.Н. Каразіна,
м. Харків, 61022, Україна, м. Харків, майдан Свободи, 4
e-mail: yehor.havryliuk@karazin.ua
https://orcid.org/0000-0002-4392-2000

Коробчинський Кирил Петрович
Доцент, доцент кафедри математичного моделювання та штучного інтелекту, Національний аерокосмічний університет «Харківський авіаційний інститут», 61070, Україна, м. Харків, вул. Манька Вадима, 17
e-mail: k.korobchinskiy@khai.edu
https://orcid.org/0000-0002-3676-6070

UML-орієнтована інформаційна технологія для неперервних задач максимального покриття з об’єктами довільної форми

Актуальність. Неперервні задачі максимального покриття з об’єктами довільної форми відіграють важливу роль у геоінформаційних системах, моніторингових платформах, логістичних сервісах, системах безпеки, аналізі просторових даних та рішеннях підтримки прийняття рішень. Зростання обсягів даних, динамічність середовищ і висока складність моделей потребують створення формалізованих, модульних і масштабованих інформаційних технологій. UML, як стандарт моделювання, дозволяє формально описати архітектуру програмних рішень, забезпечуючи надійність, повторюваність та прозорість програмної реалізації.
Мета. Розробити UML-орієнтовану інформаційну технологію розв’язання неперервних задач максимального покриття, що включає архітектурну модель, структуру даних, інформаційні потоки, функціональні компоненти та UML-специфікації модулів для реалізації систем покриття.
Методи дослідження. Застосовано методи об’єктно-орієнтованого та структурного моделювання, UML-діаграмування (Use Case, Class, Activity, Sequence, Component, Composite Structure, State Machine, Deployment), методи архітектурного проєктування, принципи модульності, інверсії залежностей, компонентної декомпозиції та підходи до побудови масштабованих інформаційних систем.
Результати. Побудовано повну UML-специфікацію архітектури інформаційної технології для задач максимального покриття: визначено зовнішні сценарії взаємодії, класи, компоненти, послідовності операцій, логіку поведінки і станів системи, інфраструктурні зв’язки та структуру розгортання. Сформовано інтегровану трирівневу архітектуру (рівень представлення, прикладної логіки, даних). Описано принципи формування модулів просторової аналітики, оптимізації, обчислення критерію покриття, управління сценаріями покриття, візуалізації та інтерфейсів даних. UML-моделі забезпечують формалізовану структуру, що дозволяє розробляти масштабовані й відтворювані IT-рішення для задач покриття.
Висновки. Створена інформаційна технологія забезпечує структурну, поведінкову та архітектурну формалізацію системи максимального покриття. UML-орієнтоване моделювання дозволяє підвищити прозорість архітектури, зменшити ризики інтеграційних помилок, забезпечити масштабованість і повторне використання компонентів. Отримані UML-моделі можуть слугувати методологічною основою для побудови інтелектуальних GIS-платформ, оптимізаційних сервісів, систем моніторингу та аналітичних рішень у реальному масштабі.
Ключові слова: *UML, інформаційна технологія, максимальне покриття, архітектура програмних систем, просторові дані, моделювання, оптимізація, GIS.*

Як цитувати: Гаврилюк Є. А., Коробчинський К. П. UML-орієнтована інформаційна технологія для неперервних задач максимального покриття з об’єктами довільної форми. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.18-34. https://doi.org/10.26565/2304-6201-2025-67-02

How to quote: Y. Havryliuk, K. Korobchynskyi, “UML-oriented information technology for continuous maximum coverage problems with arbitrary-shaped objects”, *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 67, pp. 18-34, 2025. https://doi.org/10.26565/2304-6201-2025-67-02 [in Ukrainian]

Вступ

Задачі максимального покриття відіграють ключову роль у широкому спектрі сучасних інформаційних технологій. До типових застосувань належать оптимізація розміщення сенсорів у моніторингових системах, побудова інфраструктурних об’єктів (медичних, транспортних, сервісних), оптимізація роботи безпілотних апаратів, проектування систем відеоспостереження, планування ресурсів та аналіз просторових даних у GIS-системах. Незважаючи на велику кількість

робіт, присвячених математичним методам розв'язання подібних задач, питання проектування інформаційних технологій, архітектурних моделей і програмних систем, здатних практично масштабувати такі рішення, залишаються недостатньо висвітленими.

Створення інформаційної технології охоплює алгоритмічні, архітектурні, інженерні та інтеграційні аспекти, що вимагає застосування стандартизованих методів моделювання. Серед них особливе місце займає UML-моделювання як універсальний засіб проєктування програмних систем. На відміну від математичних формалізацій, UML дозволяє формально описати структуру, поведінку, компоненти та сценарії взаємодії у вигляді стандартизованих діаграм, що значно підвищує надійність і передбачуваність процесу розробки.

У роботі запропоновано комплексну інформаційну технологію розв'язання задач максимального покриття, яка базується на архітектурі модульного типу, формально змодельована за допомогою UML, підтримує інтеграцію просторових даних, містить спеціалізований модуль оптимізації, включає інтелектуальні механізми керування та аналізу. Запропонована технологія може бути використана як основа для створення практичних IT-рішень у сферах телекомунікацій, логістики, міського планування, військової аналітики, кібербезпеки та ін.

Огляд публікацій з тематики дослідження.

Розроблення інформаційних технологій для задач максимального покриття перебуває на перетині програмної інженерії, геоінформаційних систем та оптимізаційного моделювання. У програмній інженерії основу для побудови архітектурних моделей становлять фундаментальні роботи з проектування програмних систем. У класичних джерелах [1–3] описано принципи модульності, інверсії залежностей, абстракції та багатоетапної архітектурної декомпозиції, що формують базу для створення масштабованих IT-рішень. Методи архітектурного дизайну та проєктування високорівневих компонентів, що використовуються у розподілених системах, викладено в сучасних працях з архітектури програмного забезпечення [4–6], які визначають рекомендації щодо структуризації компонентів, сценаріїв взаємодії та управління складністю систем.

UML як міжнародний стандарт моделювання, регламентований OMG, відіграє центральну роль у формалізації програмних систем. У базових та прикладних роботах [7–10] детально описано нотації структурних (Class, Component, Deployment), поведінкових (Activity, State Machine, Sequence) та інтеграційних (Composite Structure, Use Case) діаграм, що забезпечують можливість уніфікованого опису архітектури IT-рішень. Ці джерела формують методологічну основу для моделювання інформаційних технологій, подібних до тієї, що розглядається у цій статті.

Сучасні підходи до роботи з просторовими даними та геоінформаційними системами представлені у фундаментальних джерелах [11, 12], де наведено принципи аналізу геометричних об'єктів, індексації, операцій над геометричними формами та методів обчислення метричних характеристик просторових областей. Ці роботи є базовими для підсистеми просторової аналітики, що є частиною запропонованої інформаційної технології.

Алгоритмічні та прикладні аспекти задач покриття представлені в інтенсивно розвиненій групі робіт з оптимізації, мультимодальних моделей та евристичних методів. Роботи [13–15] містять загальні підходи до оптимізації покриття та низку конкретних алгоритмічних технік, зокрема градієнтні та комбінаторні методи. У системах розподіленої обробки просторових даних широко застосовуються архітектури високопродуктивних обчислень [16, 17], що дозволяють масштабувати обчислення для великих геометричних областей та складних конфігурацій покриття.

Блок досліджень, пов'язаний з математичним моделюванням задач максимального покриття, сформував окрему наукову лінію. У роботах [18–20] представлено математичні моделі неперервного максимального покриття з довільною формою областей, методи оптимізації та дослідження властивостей отриманих оптимальних конфігурацій. Ці праці формують теоретичну основу, на якій базується функціональність оптимізаційного модуля запропонованої інформаційної технології.

Окрему підгрупу становлять дослідження, присвячені застосуванню програмних бібліотек обчислювальної геометрії для практичної реалізації задач покриття. У публікаціях [21, 22] детально описано використання бібліотеки Shapely для моделювання, об'єднання, перетину та метричних оцінок геометричних областей. Ці роботи мають прикладне значення для формування підсистеми геометричних операцій нашої інформаційної технології.

У практично орієнтованих задачах покриття широко використовуються методи прогнозування, аналітики та застосування покриття в реальних сервісах. Такі аспекти висвітлено у роботі [23], де оптимізація розміщення мобільних сервісів інтегрується з прогнозними моделями у кризових сценаріях. Це актуалізує використання покриття в задачах планування, логістики та моніторингу.

Остання група джерел пов'язана з надійністю сенсорних мереж і гібридних систем моніторингу, які також базуються на принципах покриття. Дослідження [24, 25] розглядають архітектуру та надійність конфігурацій сенсорних мереж для екологічного та кризового моніторингу. Вони демонструють, що задачі покриття є ключовими не лише в оптимізації, але і в забезпеченні стійкості систем.

Таким чином, огляд літератури показує, що хоча математичні, алгоритмічні та геометричні аспекти задач максимального покриття досліджені досить широко, архітектурні рішення, побудовані на UML-моделюванні, практично не представлені. Це визначає наукову новизну та практичну значущість запропонованої інформаційної технології.

Формальна постановка неперервної задачі максимального покриття.

У загальному вигляді неперервна задача максимального покриття полягає у знаходженні такої конфігурації геометричних об'єктів (агентів), яка забезпечує максимальне покриття заданої області при виконанні певної системи обмежень на розташування покриваючих об'єктів.

Нехай:

$\Omega \subset R^2$ – компактна область покриття (довільна вимірна множина);

n – кількість покриваючих об'єктів;

$\{S_1,...,S_n\}$ - набір компактних покриваючих об'єктів (агентів) довільної форми;

$p_i = (x_i, y_i)$ – координати трансляції об'єкта S_i;

θ_i – кут повороту об'єкта S_i;

$A(\theta_i)$ – матриця повороту;

$S_i'(p_i, \theta_i) = A(\theta_i)S_i + p_i$ – трансформований об'єкт;

k – кількість заборонених зон Z_j, $j = 1,...,k$ для координат $p_i = (x_i, y_i), i = 1,...,n$.

Потрібно знайти такі $(p_1, \theta_1,..., p_n, \theta_n)$, щоб максимізувати площу покриття області Ω:

$$F(p_1, \theta_1,..., p_n, \theta_n) = area\left(\Omega \cap \bigcup_{i=1}^{n} S_i'(p_i, \theta_i)\right) \to \max \quad (1)$$

за умови

$$p_i \notin \bigcup_{j=1}^{k} Z_j,\ i = 1,...,n.$$

Обчислення значення функції (1) є нетривіальною геометричною операцією. У роботі передбачається використання п'яти підходів, кожен із яких має власний компроміс між точністю та швидкістю:

- *Сітковий метод (Монте–Карло)* - Оцінка покриття за часткою точок, що потрапили у область покриття. Гарантує універсальність та високу швидкодію, але має стохастичну похибку $O(1/N)$.
- *Метод подвійних перетинів (обмежений Inclusion–Exclusion)* - Ураховує площі поодиноких об'єктів та попарні перетини. Для кругів та еліпсів доступні точні формули. Ігнорує перетини третього порядку і вище.
- *Точні геометричні обчислення (Shapely/sympy.geometry)* - Дає аналітично точний результат і дозволяє працювати з об'єктами довільної форми. Недолік — квадратична складність) $O(N^2)$ непридатність на ранніх етапах оптимізації.
- *Метод тріангуляції області покриття* - Розбиття області $\Omega \subset R^2$ на трикутники з точним обчисленням перетинів. Висока точність, низька швидкість.
- *Метод обмежувальних рамок (bounding boxes)* - Дуже швидкий грубий метод для попередніх оцінок на стартових ітераціях оптимізації.

З практичної точки зору технологія повинна дозволяти динамічно перемикати методи обчислення площі залежно від етапу оптимізації, складності конфігурації та потрібної точності.

Зазначимо, що оптимізаційна задача (1) є нелінійною, багатовимірною, багатоекстремальною, з геометричними обчисленнями всередині функції. Тому для її розвязування застосовуються стійкі підходи глобальної оптимізації: метод штрафних функцій для врахування обмежень; ройові алгоритми (PSO, ACO, FA тощо); генетичні алгоритми (GA); меметичні алгоритми (hybrid GA + локальний пошук); локальні методи (Nelder–Mead, Powell, Quasi-Newton) у комбінованих схемах.

У практичних застосуваннях особливо важливо забезпечити: перехід від швидких грубих методів площі до точних повільних методів на фінальних етапах; адаптацію параметрів алгоритмів залежно від поточної конфігурації; стійкість до складної геометрії (неправильні полігони, багаточисельні перетини, вузькі заборонені зони).

Попри наявність значної кількості публікацій щодо методів оптимізації, більшість із них не описують архітектуру ПЗ для реалізації таких моделей, не пропонують цілісних технологічних конвеєрів, не враховують різні методи обчислення площі та перемикання між ними, не формалізують структури даних і модулі системи, не описують UML-моделі, необхідні для практичної реалізації.

У зв’язку з цим у даній статті основна увага приділяється розробленню UML-орієнтованої інформаційної технології, яка формалізує представлення геометричних об’єктів різних типів, інтеграцію методів обчислення покритої площі, керування ройовими, генетичними та меметичними оптимізаторами, підтримку масштабованої та розширюваної архітектури, можливість застосувань у GIS, моніторингу, логістиці та системах безпеки.

Архітектура інформаційної технології розв’язання задач максимального покриття.

Інформаційна технологія розв’язання неперервних задач максимального покриття з об’єктами довільної форми має бути орієнтована на підтримку повного циклу: від формулювання математичної постановки до отримання верифікованого рішення, придатного для практичного використання у ГІС-системах, системах моніторингу та підтримки прийняття рішень.

На основі постановки задачі, наведеної у попередньому розділі, інформаційна технологія повинна забезпечувати:

- завантаження, зберігання та попередню обробку просторових даних (область покриття, заборонені зони, початкові розташування покриваючих об’єктів);
- підтримку різних типів геометричних об’єктів для області, покриваючих об’єктів та зон заборони (багатокутники, кола, еліпси й інші композитні фігури);
- конфігуровані обмеження на розташування покриваючих об’єктів (геометричні обмеження, заборонені зони);
- наявність модуля обчислення критерію покриття, який підтримує щонайменше методи оцінювання площі покритої частини області (Монте-Карло-сітковий, обмежене включення–виключення, бібліотеки обчислювальної геометрії, триангуляція, апроксимація обмежувальними рамками);
- адаптивний вибір методу обчислення покриття залежно від етапу оптимізації, розмірності задачі та доступних обчислювальних ресурсів;
- універсальний оптимізаційний модуль, який підтримує як глобальні метаевристики (ройові алгоритми, генетичні, меметичні алгоритми), так і локальні методи покращення (градієнтоподібні, локальний пошук, hill-climbing);
- механізми зв’язку між вибором оптимізаційного алгоритму та методом обчислення площі, що дозволяють застосовувати швидкі наближені оцінки на ранніх етапах і точні обчислення на фінальних;
- візуалізацію проміжних і фінальних рішень, формування звітів, експорт результатів у ГІС-формати;
- модульну, розширювану та масштабовану архітектуру з чітко визначеними інтерфейсами.

Архітектура реалізується у вигляді класичної *трирівневої моделі:*

1. *Рівень представлення (Presentation Layer***)** – графічний або веб-інтерфейс аналітика, модулі візуалізації покриття, засоби налаштування задачі.

2. *Рівень прикладної логіки (Application / Logic Layer)* – підсистеми геометричного моделювання, оцінювання покриття, оптимізації, керування сценаріями, адаптивного вибору стратегій.

3. *Рівень даних (Data Layer)* – сховище геоданих, параметрів сценаріїв, історії конфігурацій, логів.

UML-діаграми, що описуються далі, послідовно розкривають ці рівні з різних точок зору: функціональної (Use Case), структурної (Class, Component, Composite Structure), поведінкової (Activity, Sequence, State Machine) та фізичної (Deployment).

Діаграма варіантів використання (Use Case Diagram)

UML-діаграма варіантів використання (Рис.1) формалізує зовнішню функціональну специфікацію інформаційної технології з точки зору користувачів та зовнішніх систем. Вона визначає, які задачі може виконувати аналітик, які сервіси надає система та як відбувається інтеграція з ГІС і обчислювальною інфраструктурою.

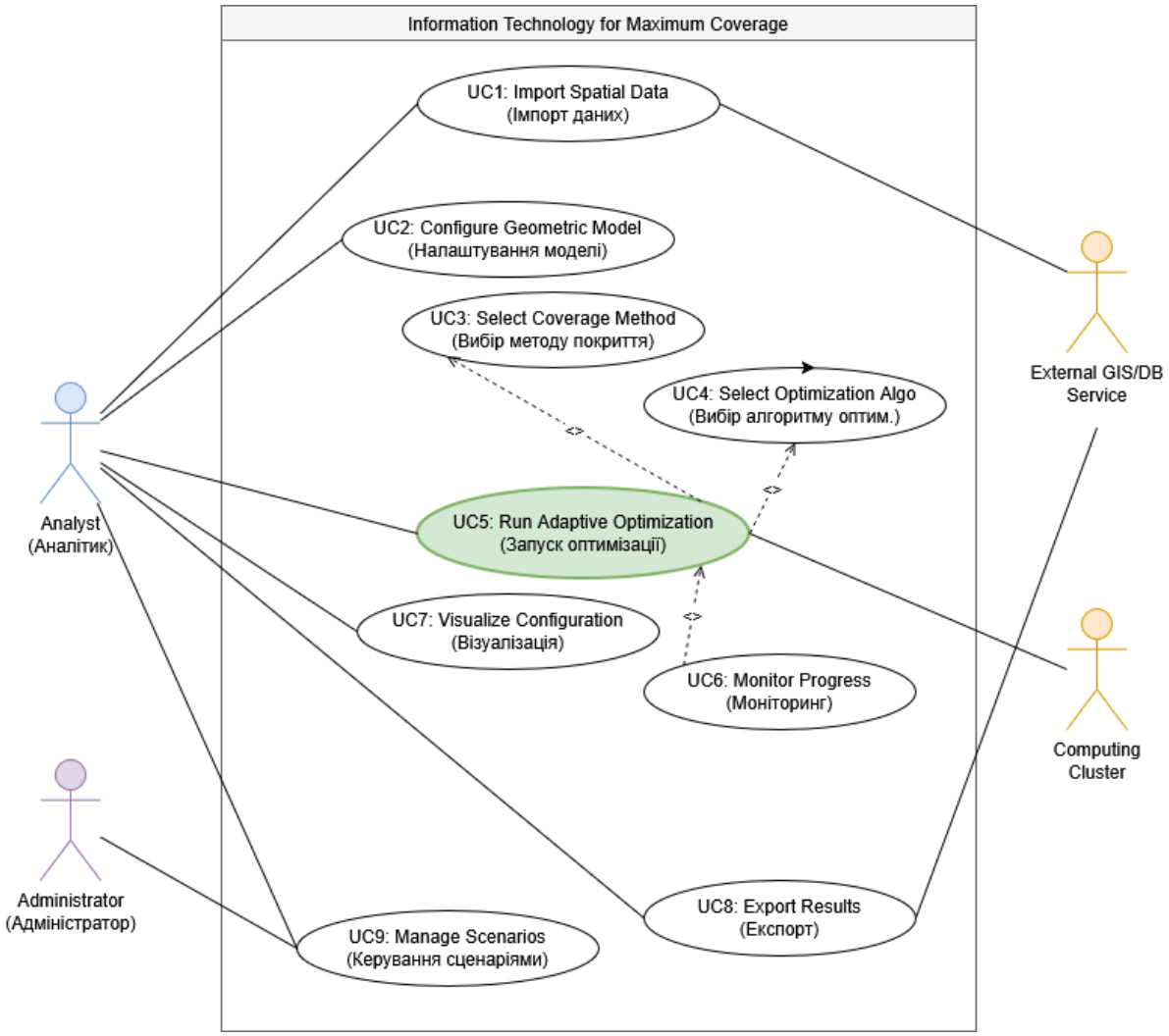


Рис. 1 – Діаграма варіантів використання UML-орієнтованої інформаційної технології для неперервних задач максимального покриття з об'єктами довільної форми
Use Case Diagram of the UML-oriented information technology for continuous maximum coverage problems with arbitrary-shaped objects.

Основні актори:

- Аналітик – формулює задачу, обирає типи геометричних об'єктів, методи обчислення покриття та оптимізації, запускає обчислення, аналізує результати.
- Адміністратор системи – керує конфігураціями, правами доступу, обчислювальними ресурсами.
- Зовнішній GIS/DB-сервіс – постачає та приймає просторові дані (геометрія області, зон заборони, довідкова інфраструктура).
- Обчислювальний кластер покриття – виконує ресурсомісткі операції з оцінювання покриття та оптимізації (може бути абстрагований як зовнішній сервіс).

Ключові варіанти використання:

- UC1 – Імпорт просторових даних (область, заборонені зони, інфраструктура).

- UC2 – Налаштування геометричної моделі (вибір типів об'єктів: багатокутник, коло, еліпс; параметрів форми).
- UC3 – Вибір методу обчислення покриття (доступних стратегій та їх комбінації).
- UC4 – Вибір алгоритмів оптимізації (ройові, генетичні, меметичні, локальний пошук).
- UC5 – Запуск адаптивної оптимізації (з автоматичним перемиканням між грубими та точними методами оцінювання).
- UC6 – Моніторинг прогресу та якості покриття (графіки збіжності, показники якості).
- UC7 – Візуалізація конфігурації покриття (карта покриття, зони прогалин, порушення обмежень).
- UC8 – Експорт рішень та звітів (карти, таблиці, журнали роботи).
- UC9 – Керування шаблонами сценаріїв (збереження й повторне використання постановок задачі).

Між UC3, UC4 та UC5 задаються залежності типу *include/extend*: вибір методів покриття та оптимізації є обов'язковими етапами перед запуском адаптивної оптимізації; деталізовані режими моніторингу та експорту реалізуються як розширення.

На діаграмі наведено основних акторів (аналітик, адміністратор, зовнішні сервіси) та ключові варіанти використання, включаючи вибір геометричної моделі, методів обчислення покриття та алгоритмів оптимізації.

Діаграма класів (Class Diagram)

UML-діаграма класів (Рис.2) задає статичну структуру інформаційної технології: сутності предметної області, їх атрибути, методи та зв'язки. Для нашої задачі виділено чотири взаємопов'язані підсистеми: геометричну, підсистему оцінювання покриття, оптимізаційну та інфраструктурну.

Геометрична підсистема відповідає за представлення області покриття, покриваючих об'єктів та зон заборони з використанням різних типів геометрії. Основні класи:

- AbstractGeometry – абстрактний базовий клас для геометричних об'єктів.
  Атрибути: id, geometryType.
  Методи: area(), contains(point), intersect(another), transform(transformParams).
- Region (наслідує AbstractGeometry) – область покриття $\Omega \subset R^2$.
  Атрибути: boundary : GeometryShape, holes : List<GeometryShape>.
  Методи: clip(geom), toPolygonalApprox().
- ForbiddenRegion (наслідує AbstractGeometry) – заборонені зони.
  Атрибути: severityLevel, penaltyWeight.
  Методи: violatedBy(configuration).
- CoverageObject (наслідує AbstractGeometry) – покриваючий об'єкт.
  Атрибути: shapeType (polygon/circle/ellipse), baseShape : GeometryShape, orientation, params.
  Методи: placeAt(position, orientation), getFootprint().
- PolygonShape, CircleShape, EllipseShape – конкретні класи форм, що інкапсулюють параметри (вершини, радіус, півосі тощо) та реалізують спеціалізовані методи обчислення площі, перетинів тощо (часто через геометричну бібліотеку).

Підсистема оцінювання покриття реалізує методи обчислення площі покритої частини області, організованих за шаблоном Strategy.

- CoverageEstimator – інтерфейс (або абстрактний клас).
  Методи:
  estimateCoverage(region : Region, objects : List<CoverageObject>) : CoverageResult.
- MonteCarloEstimator – сітковий/монте-карло метод.
  Атрибути: numSamples, samplingScheme.
  Особливості: застосовується на ранніх етапах оптимізації як швидкий наближений метод.
- PairwiseInclusionExclusionEstimator – метод з урахуванням подвійних перетинів.
  Атрибути: maxPairs, useCircleIntersectionFormula.
  Підходить для задач з переважно круговими/еліптичними об'єктами.
- GeometryLibraryEstimator – точне обчислення через бібліотеку обчислювальної геометрії (наприклад, Shapely).
  Атрибути: tolerance, backendType.
  Використовується на фінальних етапах, коли критична висока точність.

- TriangulationEstimator – метод на основі тріангуляції області.
  Атрибути: numTriangles, refinementLevel.
  Застосовується для складних форм області $\Omega \subset R^2$, коли потрібна детальна локальна оцінка.
- BoundingBoxEstimator – швидка груба оцінка через обмежувальні рамки.
  Використовується як допоміжний метод для первинного відбору конфігурацій.
- CoverageResult – результат оцінювання.
  Атрибути: coveredArea, coverageRatio, uncoveredRatio, penaltyValue.
  Методи: isFeasible(), toReportMetrics().

Оптимізаційна підсистема керує процесом пошуку конфігурації, що максимізує покриття, з урахуванням обмежень. Основні класи:

- Configuration – конфігурація покриваючих об’єктів.
  Атрибути: objects : List<CoverageObject>, decisionVector, coverageResult, metaInfo.
  Методи: evaluate(estimator), isFeasible().
- ProblemDefinition – постановка задачі.
  Атрибути: region : Region, forbiddenRegions : List<ForbiddenRegion>,
  numObjects, allowedShapeTypes, constraints.
  Методи: generateInitialConfiguration(), repairConfiguration().
- OptimizationAlgorithm – абстрактний базовий клас.
  Методи: initialize(problem), iterate(), getBestSolution().
- PSOAlgorithm, GAAlgorithm, MemeticAlgorithm, LocalSearchAlgorithm, HybridAlgorithm – конкретні реалізації.
  Кожен клас використовує різні схеми генерації, відбору та локального покращення конфігурацій.
- CoverageStrategyManager – керування вибором методу оцінювання покриття.
  Атрибути: currentEstimator, coarseEstimatorList, preciseEstimatorList, switchCriteria.
  Методи: selectEstimator(iteration, stagnationLevel, problemSize).
- OptimizationEngine – фасад для запуску оптимізації.
  Атрибути: algorithm : OptimizationAlgorithm,
  strategyManager : CoverageStrategyManager,
  penaltyManager : PenaltyManager,
  history : OptimizationHistory.
  Методи: run(problemDefinition), step(), refineBestSolution().
- PenaltyManager – реалізує штрафні функції для обмежень (заборонені зони, ліміти кількості об’єктів, мінімальні відстані).
  Методи: computePenalty(configuration), updatePenaltyCoefficients().

У Табл.1 наведено приклад основних класів та методів.

Таблиця 1 – Приклад основних класів та методів / Example of main classes and methods

Клас	*Основні атрибути*	*Основні методи*
Region	id, boundary, holes	area(), clip(), toPolygonalApprox()
CoverageObject	id, shapeType, baseShape, orientation, params	placeAt(), getFootprint()
CoverageEstimator	estimatorType	estimateCoverage()
MonteCarloEstimator	numSamples, samplingScheme	estimateCoverage()
GeometryLibraryEstimator	backendType, tolerance	estimateCoverage()
Configuration	objects, decisionVector, coverageResult	evaluate(), isFeasible()
OptimizationEngine	algorithm, strategyManager, penaltyManager	run(), step(), refineBestSolution()
CoverageStrategyManager	currentEstimator, switchCriteria	selectEstimator()

Інфраструктурна підсистема

- DataImporter / GISAdapter – імпорт даних області, зон заборони та довідкових шарів.

- ScenarioManager – керування сценаріями експериментів.
- ResultExporter – експорт карт, звітів, конфігурацій.
- VisualizationService – візуалізація конфігурацій та динаміки збіжності.
- SystemLogger / MonitoringService – логування та моніторинг.

Зв'язки між класами включають:

- узагальнення (generalization) між AbstractGeometry та його нащадками;
- композицію (composition) між Configuration та CoverageObject;
- агрегацію (aggregation) між Region і ForbiddenRegion;
- асоціацію між OptimizationEngine, CoverageStrategyManager та реалізаціями CoverageEstimator;
- використання шаблонів *Strategy, Factory, Facade.*

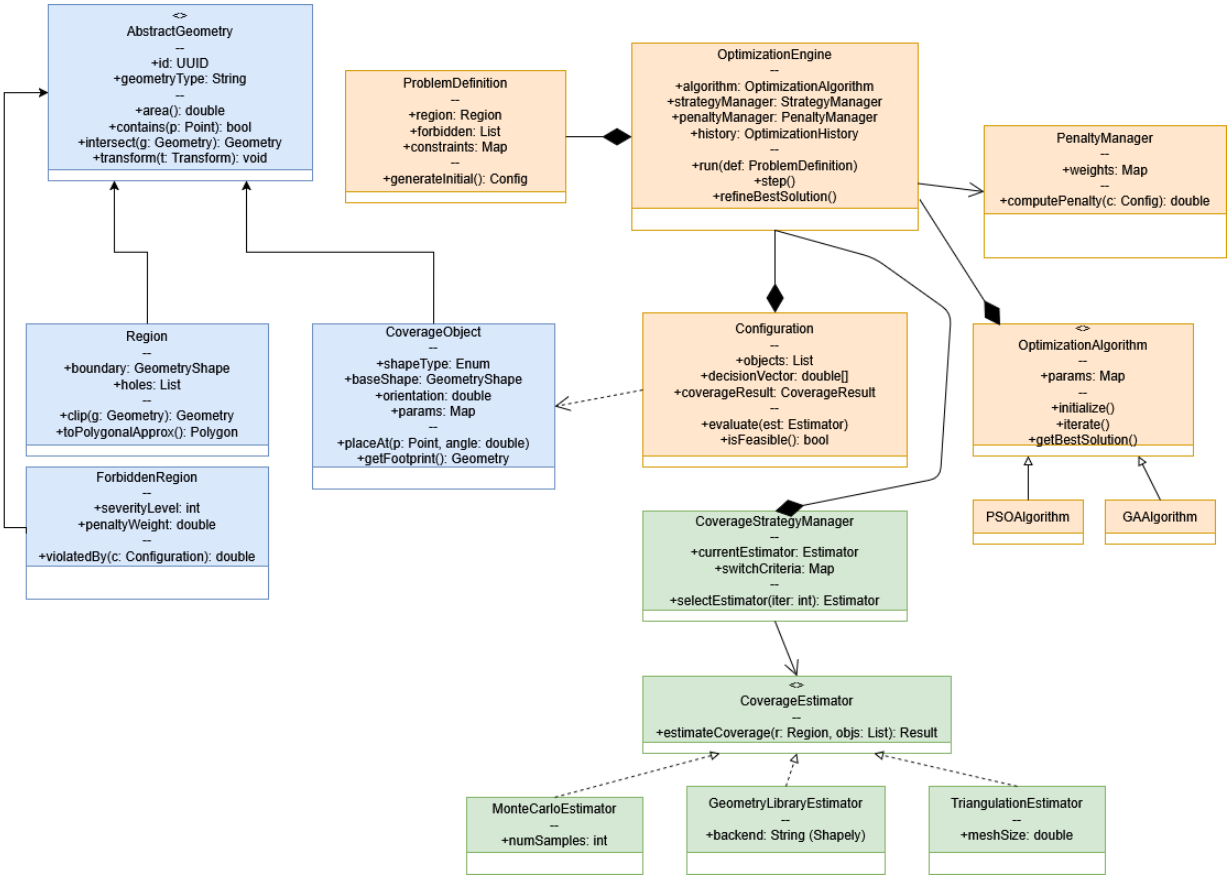


Рис. 2 – Діаграма класів UML-орієнтованої інформаційної технології для неперервних задач максимального покриття з об'єктами довільної форми.

Class Diagram of the UML-oriented information technology for continuous maximum coverage problems with arbitrary-shaped objects.

На діаграмі відображено основні класи геометричної підсистеми, підсистеми оцінювання покриття, оптимізаційного ядра та інфраструктури, а також їхні зв'язки та узагальнення.

Діаграма компонентів (Component Diagram)

Діаграма компонентів, представлена у дослідженні (Рис. 3), відображає високорівневу модульну архітектуру інформаційної технології та визначає інтерфейси взаємодії між її складовими частинами. Дана модель узгоджує логічну структуру системи з класичною трирівневою архітектурою, забезпечуючи чітку декомпозицію на рівень представлення, рівень прикладної логіки та рівень даних.

1. Рівень представлення (Presentation Layer)

Цей рівень відповідає за взаємодію з кінцевим користувачем (аналітиком) та візуалізацію результатів. До його складу входить *Visualization & Reporting Component*, що реалізує інтерфейси IVisualization та IReporting. Цей компонент відповідає за графічне представлення конфігурацій покриття, побудову карт та формування аналітичних звітів .

2. Рівень прикладної логіки (Application Logic Layer)

Це ядро системи, де зосереджена основна обчислювальна логіка. Рівень включає *Optimization Engine Component* - *ц*ентральний керуючий модуль, що реалізує інтерфейс *IOptimizationService*. Він відповідає за запуск глобальних та локальних алгоритмів оптимізації та взаємодіє з модулем оцінювання через адаптивний менеджер стратегій .

3. Рівень даних (Data Layer)

Рівень забезпечує персистентність даних та інтеграцію із зовнішнім середовищем. Рівень включає *Data Access / GIS Integration Component:* Реалізує інтерфейси IDataImport, IDataExport та IGISAdapter. Забезпечує обмін даними із зовнішніми геоінформаційними системами (GIS), базами даних та файловими сховищами .

Міжкомпонентна взаємодія

Взаємодія між рівнями та компонентами реалізується через чітко визначені інтерфейси, що забезпечує слабку зв'язність (low coupling) системи. Наприклад, Optimization Engine використовує Coverage Evaluation для оцінки рішень, який, у свою чергу, делегує геометричні обчислення компоненту Geometry Core. Така архітектура дозволяє масштабувати систему та замінювати окремі модулі без впливу на загальну функціональність.

Ця структурна організація, представлена на діаграмі компонентів (Рис. 3), є основою для побудови масштабованих та відтворюваних IT-рішень у сфері оптимізації покриття .

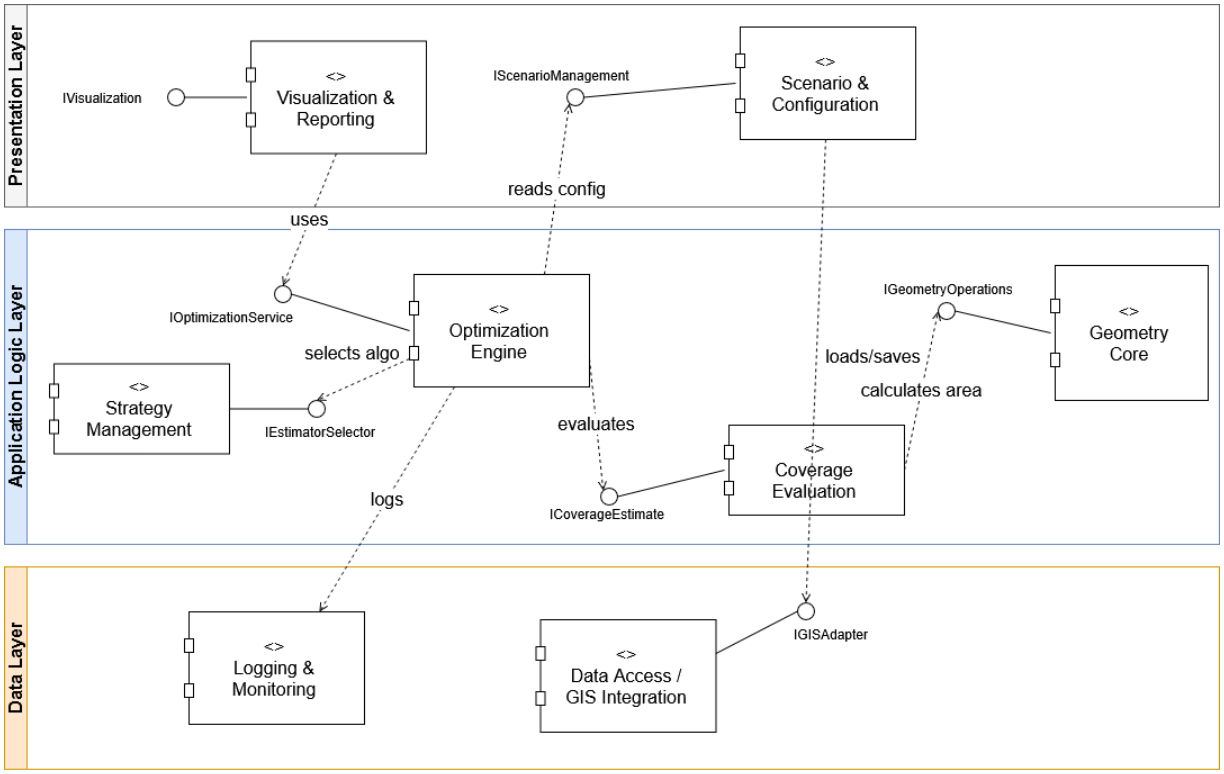


Рис. 3 – Діаграма компонентів UML-орієнтованої інформаційної технології для неперервних задач максимального покриття.

Component Diagram of the UML-oriented information technology for continuous maximum coverage problems.

Діаграма діяльності (Activity Diagram)

Діаграма діяльності (Рис.4) описує робочий процес інформаційної технології від моменту постановки задачі до отримання остаточного рішення.

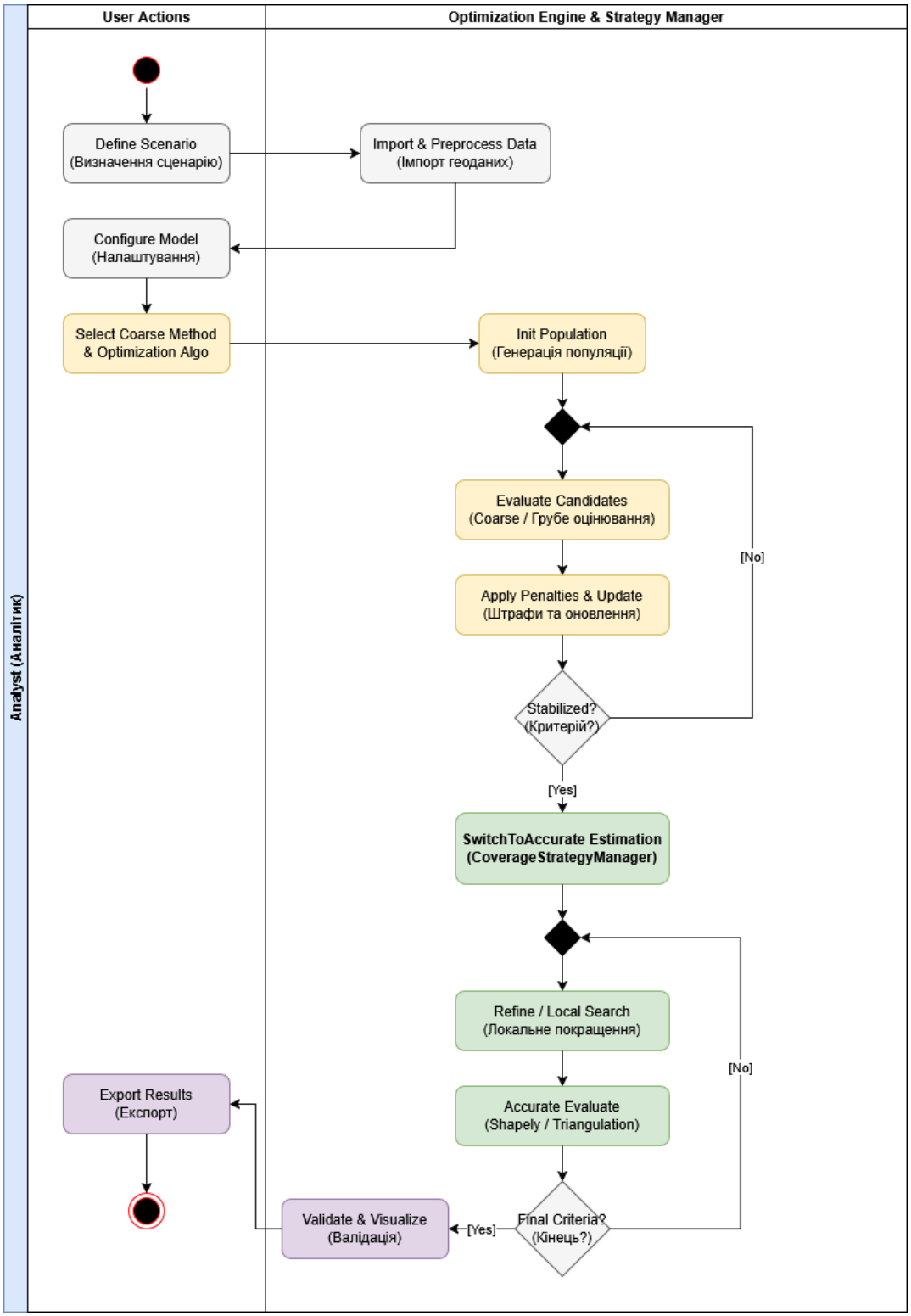


Рис. 4 – Діаграма діяльності процесу адаптивної оптимізації максимального покриття з використанням кількох методів обчислення площі та різних алгоритмів оптимізації.
Activity Diagram of the adaptive maximum coverage optimization process using multiple area-calculation methods and different optimization algorithms.

Основні етапи процесу:

- *Визначення сценарію* - Аналітик задає область покриття, типи об’єктів, обмеження, цільові функції та вимоги до точності.

- *Імпорт та попередня обробка геоданих* - Виклик дій LoadRegion, LoadForbiddenRegions, PreprocessGeometry.
- *Налаштування геометричної моделі* - Вибір типів форм (багатокутник, коло, еліпс), параметрів об’єктів та способів задання заборонених зон.
- *Вибір методу обчислення покриття* - Початково обираються швидкі наближені методи (BoundingBoxEstimator, MonteCarloEstimator) для грубої оцінки рішень.
- *Вибір алгоритмів оптимізації* - Задання глобальних методів (PSO, GA, меметичні) та, за потреби, локального пошуку.
- *Генерація початкової популяції / конфігурацій* - Діяльність InitPopulation або GenerateInitialConfigurations.
- *Основний цикл глобальної оптимізації* – EvaluateCandidates (оцінка покриття для всієї популяції за допомогою обраного грубого методу); ApplyPenalties (врахування порушень обмежень), SelectAndUpdate (відбір кращих рішень і побудова нових конфігурацій), CheckCoarseStoppingCriteria (перевірка критеріїв зупинки для грубої фази).
- *Перемикання на точні методи оцінювання* - Якщо досягнуто стабілізації покриття, активується діяльність SwitchToAccurateEstimation, де CoverageStrategyManager обирає GeometryLibraryEstimator або TriangulationEstimator.
- *Фаза уточнення рішення RefineBestConfigurations* (застосування локального пошуку і більш точних методів оцінювання), AccurateEvaluate (переоцінка кандидатів точним методом), CheckFinalCriteria (контроль фінальних критеріїв: точність, обмеження).
- *Валідація та формування остаточного рішення* - Діяльність ValidateFinalSolution використовує найточніший доступний метод, перевіряючи всі обмеження.
- *Візуалізація та експорт* - Діяльності VisualizeCoverage, GenerateReport, ExportToGIS.
- *Паралельність* - обчислення покриття для різних конфігурацій можуть виконуватися паралельно;

Діаграма відображає послідовність кроків від завдання сценарію до валідації та експорту рішення, включно з перемиканням між грубими та точними методами оцінювання.

Діаграма послідовності (Sequence Diagram)

Діаграма послідовності (Рис. 5) формалізує часову динаміку взаємодії між ключовими архітектурними компонентами інформаційної технології під час виконання адаптивної оптимізації задачі максимального покриття. Діаграма деталізує потік керування та обмін повідомленнями, необхідні для реалізації гібридної стратегії обчислення, яка поєднує грубі та точні методи оцінювання.

Учасники взаємодії (Lifelines) У процесі беруть участь такі активні об'єкти системи :

ScenarioController: Ініціює процес виконання сценарію та керує постановкою задачі.

OptimizationEngine: Виступає центральним координатором, що організовує ітераційний цикл оптимізації.

StrategyManager: Відповідає за адаптивний вибір методу обчислення покриття (Strategy Pattern) залежно від етапу оптимізації.

CoverageEstimator: Абстракція обчислювача, яка делегує виконання конкретним реалізаціям (наприклад, BoundingBoxEstimator або GeometryLibraryEstimator).

GeometryCore: Виконує низькорівневі геометричні операції (перетин, обчислення площі).

Допоміжні сервіси: PenaltyManager (розрахунок штрафів), ResultRepository (збереження рішень) та VisualizationService (відображення результатів).

Алгоритмічна логіка процесу Взаємодія компонентів реалізується у такій хронологічній послідовності:

Ініціалізація та вибір початкової стратегії: Процес розпочинається із запиту аналітика (startOptimization), після чого ScenarioController конфігурує оптимізаційне ядро (configure) . На початковій ітерації (iter=0) OptimizationEngine звертається до StrategyManager, який повертає метод грубої оцінки — BoundingBoxEstimator, що дозволяє швидко обробляти велику кількість конфігурацій .

Ітераційний цикл оцінювання (Coarse Evaluation): Після генерації початкової популяції (generateInitialConfigurations) система входить у цикл оцінювання. Для кожної конфігурації викликається метод estimateCoverage. При цьому CoverageEstimator звертається до GeometryCore для виконання операцій intersect() та area(), повертаючи об'єкт CoverageResult .

Обробка обмежень та збереження: Отримані результати передаються до PenaltyManager для обчислення штрафних функцій (computePenalty), після чого найкращі рішення зберігаються у репозиторії (saveBest).

Адаптивне уточнення (Refinement Phase): Ключовою особливістю алгоритму є динамічна зміна стратегії. При досягненні критеріїв стагнації або певної кількості ітерацій, OptimizationEngine повторно запитує стратегію у StrategyManager. На цьому етапі активується точний метод оцінювання — GeometryLibraryEstimator (на базі бібліотеки Shapely). Це ініціює цикл локального покращення та точного перерахунку метрик (Refinement Loop).

Завершення та візуалізація: Після отримання фінального розв'язку OptimizationEngine повертає результат контролеру, який ініціює його візуалізацію через виклик showFinalCoverage у компоненті VisualizationService.

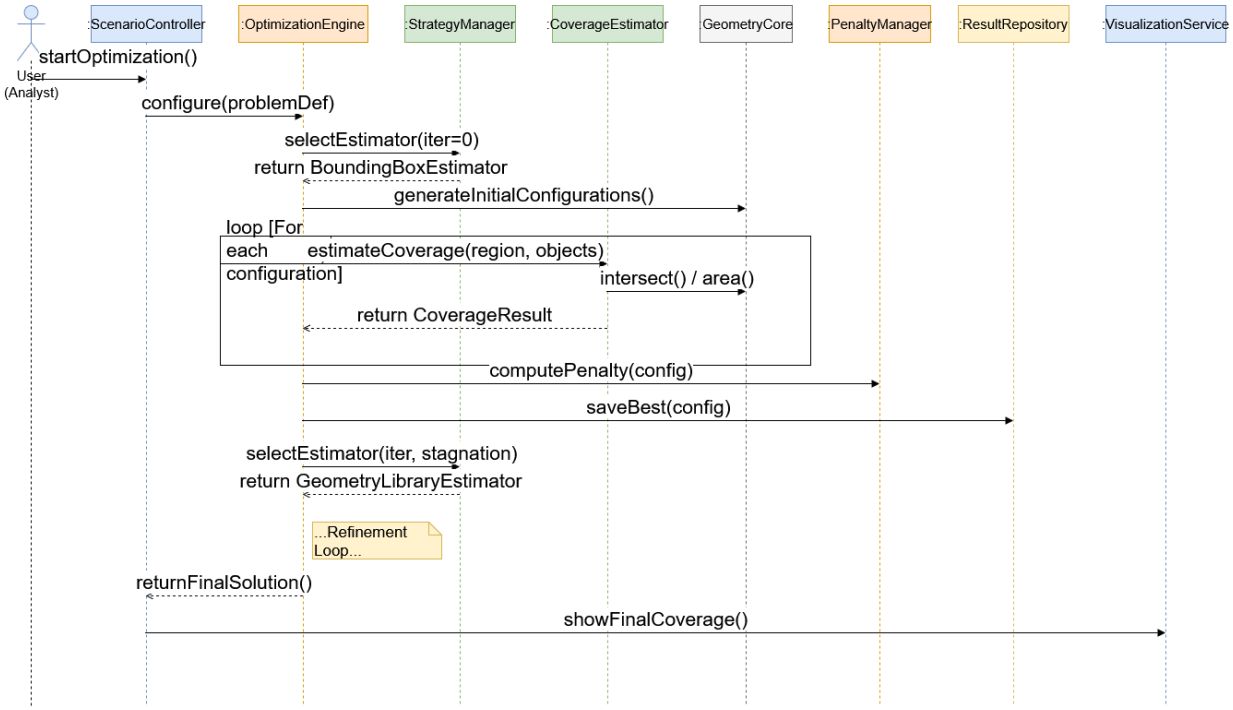


Рис. 5 – Діаграма послідовності взаємодії компонентів під час адаптивної оптимізації неперервної задачі максимального покриття.

Sequence Diagram of the interaction between components during the adaptive optimization of the continuous maximum coverage problem.

Діаграма демонструє динамічний обмін повідомленнями між користувачем, керуючим сценарієм, оптимізаційним ядром, менеджером стратегій покриття, геометричним ядром та сервісом візуалізації.

Діаграма станів (State Machine Diagram)

На Рис.6 представлена загальна характеристика діаграми станів яка формалізує життєвий цикл окремої конфігурації покриваючих об'єктів у процесі роботи адаптивного оптимізаційного алгоритму. Вона визначає логіку переходів між етапами генерації, оцінювання, фільтрації та покращення розв'язків.

Можна виділити наступні етапи життєвого циклу:

Ініціалізація та грубе оцінювання. Життєвий цикл розпочинається зі стану Generated (Згенеровано), в який об'єкт переходить після виклику події init() (випадкова генерація або створення на основі евристик). Далі ініціюється процес швидкого попереднього аналізу (подія coarseEvaluate()), що переводить систему у стан CoarseEvaluating (Грубе оцінювання). Завершення цього процесу фіксується у стані CoarseEvaluated, де конфігурація отримує наближену оцінку якості та штрафів.

Селекція та фільтрація. Ключовим етапом є перевірка перспективності рішення (guard condition isPromising?).

Негативний сценарій: Якщо конфігурація має низькі показники якості або критичні порушення обмежень, вона переходить у стан RejectedByCoarse (Відхилено). Такі об'єкти згодом переміщуються до архіву (Archived) для збереження історії пошуку.

Позитивний сценарій: Перспективні конфігурації переходять у стан SelectedForRefinement (Відібрано), що є вхідною точкою для ресурсомістких обчислень.

Точне оцінювання та локальне покращення. Для відібраних кандидатів запускається процедура точного розрахунку критеріїв (наприклад, на базі геометричних бібліотек або триангуляції), що відповідає стану AccurateEvaluating (Точне оцінювання). Після отримання точних метрик (AccurateEvaluated) до конфігурації застосовуються методи локального пошуку (applyLocalSearch()), переводячи її у стан LocalImprovement (Локальне покращення).

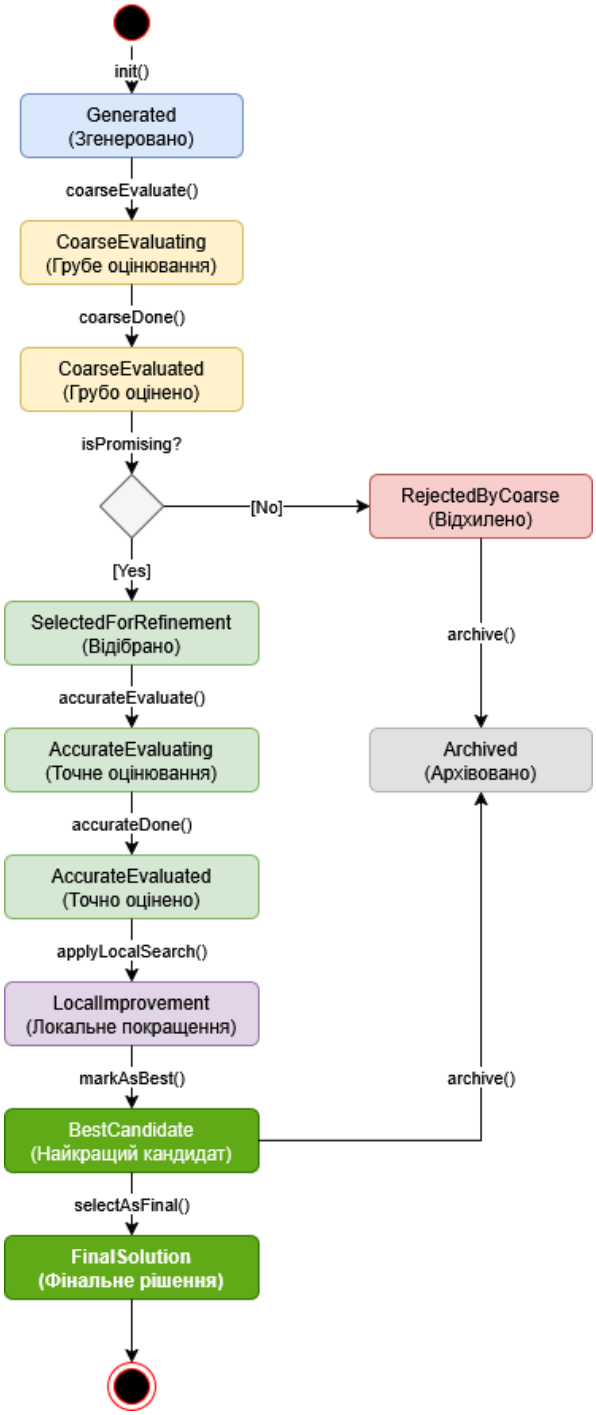


Рис. 6 – Діаграма станів конфігурації покриваючих об'єктів у процесі адаптивної оптимізації задачі максимального покриття.

State Machine Diagram of the configuration of coverage objects during the adaptive optimization of the maximum coverage problem.

Фіналізація рішення. Оптимізовані конфігурації, що продемонстрували найвищі показники ефективності, набувають статусу BestCandidate (Найкращий кандидат). З цієї множини, згідно з критеріями зупинки алгоритму, обирається єдине оптимальне рішення, яке переходить у фінальний стан FinalSolution (Фінальне рішення). Інші кандидати з множини найкращих також підлягають архівації.

Таким чином запропонована модель станів забезпечує ефективне керування обчислювальними ресурсами, дозволяючи відсіювати неперспективні рішення на етапі грубої оцінки та концентрувати обчислювальну потужність на уточненні найбільш якісних конфігурацій. Діаграма відображає переходи між станами генерації, грубого та точного оцінювання, локального покращення, відбору найкращих рішень та формування фінального результату.

Діаграма композитної структури (Composite Structure Diagram)

На Рис. 7 представлена діаграма композитної структури яка деталізує внутрішню архітектурну організацію компонента OptimizationEngine. Вона розкриває інкапсульовану логіку роботи оптимізаційного ядра, демонструючи взаємозв'язки між підмодулями, що відповідають за генерацію рішень, керування популяцією, виконання глобального та локального пошуку, а також адаптивне перемикання стратегій оцінювання .

Структурна декомпозиція компонента Внутрішня архітектура OptimizationEngine складається з наступних функціональних частин (parts):

Модулі пошуку та генерації:

GlobalSearchModule: Реалізує метаевристичні алгоритми глобальної оптимізації (PSO, GA, меметичні алгоритми) для дослідження простору рішень.

CandidateGenerator: Відповідає за процедурну генерацію початкових та проміжних геометричних конфігурацій.

LocalSearchModule: Забезпечує локальне покращення (fine-tuning) відібраних перспективних рішень.

Модулі керування даними:

PopulationManager: Виконує функцію сховища для поточної популяції кандидатів та реалізує логіку їх відбору (selection).

HistoryManager: Накопичує історію обчислень (значення функції пристосованості, метрики покриття) та використовується для детекції стагнації оптимізаційного процесу.

Модулі координації та стратегії:

EvaluationCoordinator: Виступає центральним вузлом для запитів на оцінювання конфігурацій, взаємодіючи із зовнішніми сервісами через порти.

StrategySelector: Реалізує алгоритмічну логіку динамічного перемикання між "грубими" та "точними" методами оцінювання на основі даних про хід оптимізації.

Логіка взаємодії та інформаційні потоки. Зв'язки між внутрішніми компонентами визначають ключові процеси системи:

Генерація та оновлення: GlobalSearchModule взаємодіє з CandidateGenerator та PopulationManager для створення нових особин та оновлення популяції.

Уточнення (Refinement): PopulationManager передає перспективні рішення до LocalSearchModule через конектор refinement для їх локальної оптимізації.

Оцінювання та логування: EvaluationCoordinator приймає запити на оцінку (requests eval), спрямовує їх через CoverageStrategyPort до відповідного оцінювача, а результати передає (logs result) до HistoryManager.

Адаптивне керування: HistoryManager аналізує динаміку збіжності та передає сигнал про стагнацію (stagnation info) до StrategySelector, який, у свою чергу, коригує параметри оцінювання через керуючий вплив (controls) на координатора.

Інтерфейси Взаємодія із зовнішнім середовищем здійснюється через спеціалізовані порти:

CoverageStrategyPort: Забезпечує доступ до поточного реалізатора стратегії оцінювання покриття (Strategy Pattern).

LogPort: Використовується для експорту даних логування.

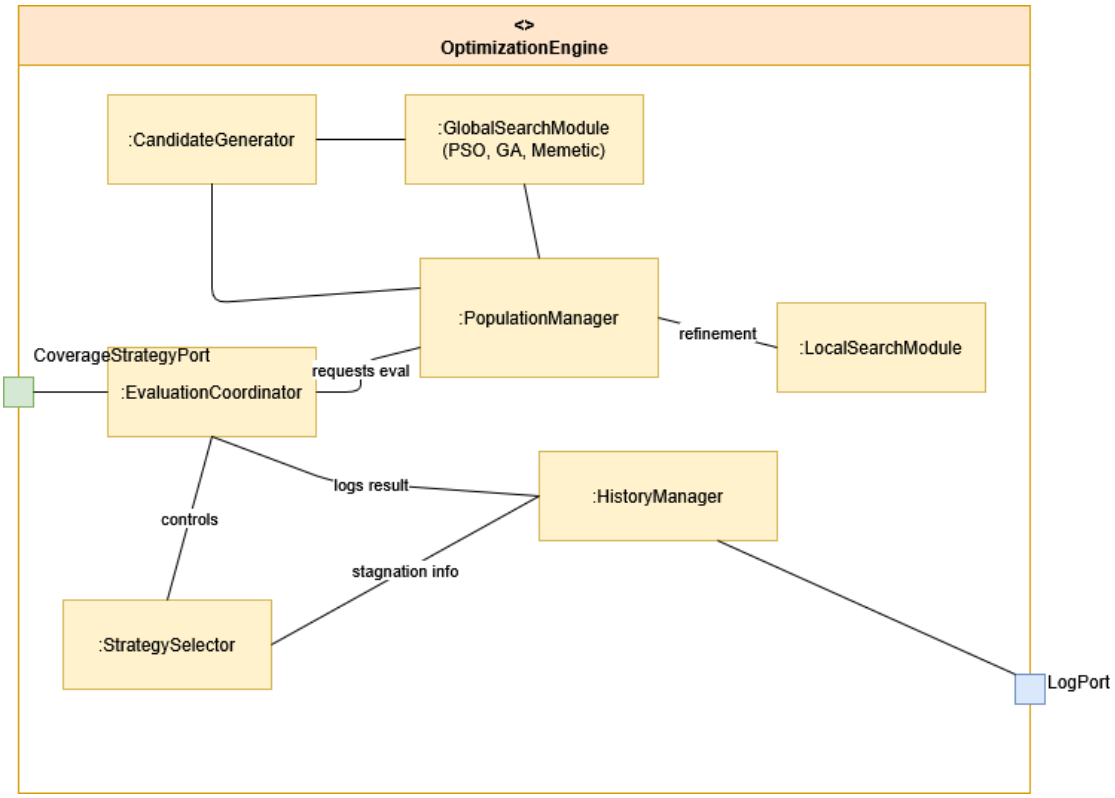


Рис. 7 – Діаграма композитної структури компонента OptimizationEngine в UML-орієнтованій інформаційній технології максимального покриття.
Composite Structure Diagram of the OptimizationEngine *component in the UML-oriented information technology for maximum coverage.*

На діаграмі показано внутрішні частини рушія оптимізації, порти та з'єднання між модулями глобального й локального пошуку, координації оцінювання, вибору стратегій та керування історією.

Висновки

У статті представлено цілісну UML-орієнтовану інформаційну технологію розв'язання неперервних задач максимального покриття з об'єктами довільної форми. Технологія формалізує архітектуру, структури даних, інформаційні потоки та алгоритмічні компоненти системи, забезпечуючи відтворюваність, масштабованість і прозорість процесу проєктування програмних рішень для задач покриття. Розроблений комплекс UML-діаграм охоплює структурні, поведінкові та інтеграційні аспекти системи, що дає змогу однозначно описати її логіку та підтримувати розширюваність.

Наукова новизна виконаного дослідження полягає у створенні уніфікованої UML-орієнтованої інформаційної технології, призначеної для розв'язання неперервних задач максимального покриття з геометричними об'єктами довільної форми.

У роботі запропоновано повну архітектурну формалізацію задач максимального покриття засобами UML. Побудовано комплекс діаграм (Use Case, Class, Component, Activity, Sequence, State Machine, Composite Structure), який формує стандартизований архітектурний каркас системи. На відміну від існуючих досліджень, що концентруються лише на алгоритмах, технологія враховує повний життєвий цикл програмної системи. Введено універсальний метамодуль оцінювання площі покриття, що об'єднує п'ять незалежних стратегій обчислення та підтримує їх адаптивне перемикання. Така інтеграція вперше формалізована UML-моделями та дозволяє ефективно працювати зі складними геометріями та великими конфігураціями. Розроблено новий підхід до адаптивної оптимізації, який поєднує грубі й точні методи оцінювання, глобальні й локальні алгоритми, а також механізм динамічного вибору стратегії. Це створює гібридну оптимізаційну платформу, яка є науково унікальною. Запропоновано оригінальну модель життєвого циклу конфігурації у вигляді діаграми станів, яка формалізує переходи між станами оцінювання, покращення та архівації кандидатних рішень. Сформовано композитну структуру

оптимізаційного ядра, що визначає внутрішню організацію глобального пошуку, локального покращення, моніторингу та стратегічного керування. Це забезпечує масштабованість і відтворюваність архітектурних рішень.

Таким чином, робота формує клас інформаційних технологій, у яких UML виступає не просто засобом документування, а фундаментом архітектурного проєктування систем оптимізації покриття.

Подальші дослідження можуть бути спрямовані на інтеграцію методів машинного навчання для автоматичного вибору стратегій оптимізації, розроблення розподіленої та паралельної реалізації оптимізаційного ядра, підтримку динамічних задач покриття зі змінними у часі геометричними об’єктами, інтеграцію з потоковими сенсорними мережами реального часу.

REFERENCES

1. Object Management Group, “Unified Modeling Language (UML), Version 2.5.1,” formal/17-12-05, Dec. 2017. [Online]. Available: https://www.omg.org/spec/UML/2.5.1

2. J. Arlow and I. Neustadt, *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2005, p. 624.

3. P. Clements, F. Bachmann, L. Bass et al., *Documenting Software Architectures: Views and Beyond*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2010, p. 624.

4. L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2012, p. 624.

5. R. N. Taylor, N. Medvidović, and E. M. Dashofy, *Software Architecture: Foundations, Theory, and Practice*. Hoboken, NJ, USA: Wiley, 2009, p. 736.

6. I. Rauf, M. Z. Iqbal, and Z. I. Malik, “UML based modeling of web service composition—A survey,” *Int. J. Comput. Appl.*, vol. 1, no. 6, pp. 301–307, 2008. doi: 10.5120/324-524.

7. P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Systems and Science*, 3rd ed. Chichester, U.K.: Wiley, 2011, p. 560.

8. S. Gillies, *Shapely: Computational Geometry Library*, ver. 2.0.0. Zenodo, 2021. doi: 10.5281/zenodo.7428463.

9. K. Jordahl et al., “GeoPandas: Python tools for geographic data,” *J. Open Source Softw.*, vol. 9, no. 1083, Art. no. 5660, Mar. 2023. doi: 10.21105/joss.05660.

10. J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proc. IEEE Int. Conf. Neural Netw. (ICNN'95)*, Perth, WA, Australia, 1995, vol. 4, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

11. C. J. A. Bastos-Filho et al., “A novel search algorithm based on fish-school behavior,” *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 39, no. 2, pp. 237–252, Apr. 2009. doi: 10.1109/TSMCC.2009.2030235.

12. X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, 2nd ed. Beckington, U.K.: Luniver Press, 2010, p. 148.

13. J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006, p. 664. doi: 10.1007/978-0-387-40065-5.

14. W. E. Hart, N. Krasnogor, and J. E. Smith, Eds., *Recent Advances in Memetic Algorithms*, vol. 166. Berlin, Germany: Springer, 2005. doi: 10.1007/3-540-32363-5.

15. A. Calvagna, A. Gargantini, and E. Viganò, “An adaptive penalty based parallel tabu search for constrained covering array generation,” *Inf. Softw. Technol.*, vol. 138, Art. no. 106768, Oct. 2021. doi: 10.1016/j.infsof.2021.106768.

16. J. Kallrath, “Cutting circles and polygons from area-minimizing rectangles,” *J. Glob. Optim.*, vol. 43, no. 2–3, pp. 267–298, Jun. 2009. doi: 10.1007/s10898-007-9251-2.

17. Y. Shi, H.-Z. Huang, Y. Liu, Y.-F. Li, and X.-Y. Xiao, “A new reliability analysis method based on the efficient Latin hypercube sampling,” *Struct. Multidiscip. Optim.*, vol. 58, no. 6, pp. 2371–2386, Dec. 2018. doi: 10.1007/s00158-018-1978-3.

18. S. V. Yakovlev, “The concept of modeling packing and covering problems using modern computational geometry software,” *Cybern. Syst. Anal.*, vol. 59, no. 1, pp. 108–119, Jan. 2023. doi: 10.1007/s10559-023-00547-5.

19. S. Yakovlev, O. Kartashov, and A. Mumrienko, “Formalization and solution of the maximum area coverage problem using library Shapely for territory monitoring,” *Radioelectron. Comput. Syst.*, vol. 2, pp. 35–48, 2022. Available: http://nti.khai.edu/ojs/index.php/reks/article/view/reks.2022.2.03.

20. S. Yakovlev, O. Kartashov, and D. Podzeha, “Mathematical models and nonlinear optimization in continuous maximum coverage location problem,” *Computation*, vol. 10, no. 7, Art. no. 119, Jul. 2022. doi: 10.3390/computation10070119.

21. S. Yakovlev, O. Kiseleva, D. Chumachenko, and D. Podzeha, “Maximum service coverage in business site selection using computer geometry software,” *Electronics*, vol. 12, no. 10, Art. no. 2329, May 2023. doi: 10.3390/electronics12102329.

22. S. Yakovlev et al., “Continuous maximum coverage location problem with arbitrary shape of service areas and regional demand,” *Symmetry*, vol. 17, no. 5, Art. no. 676, 2025. doi: 10.3390/sym17050676.

23. S. Yakovlev et al., “Optimization of mobile medical service locations based on predictive analytics in crisis scenarios,” in *Proc. IADIS Inf. Syst. E-Soc.*, 2025, pp. 538–541.

24. K. Leichenko et al., “Assessment of the reliability of wireless sensor networks for forest fire monitoring systems considering fatal combinations of multiple sensor failures,” *Cybern. Syst. Anal.*, vol. 61, no. 1, pp. 137–147, Jan. 2025. doi: 10.1007/s10559-025-00722-w.

25. S. Skorobohatko et al., “Architecture and reliability models of hybrid sensor networks for environmental and emergency monitoring systems,” *Cybern. Syst. Anal.*, vol. 60, no. 2, pp. 293–304, Mar. 2024. doi: 10.1007/s10559-024-00670-x.

Havryliuk Yehor Andriyovych *PhD student of the Department of Mathematical Modeling and Data Analysis Karazin Kharkiv National University, Svobody Sq 4, Kharkiv, Ukraine, 61022 e-mail: yehor.havryliuk@karazin.ua* **https://orcid.org/0000-0002-4392-2000**

Korobchynskyi Kyryl Petrovych *Associate Professor of the Department of Mathematical Modeling and Artificial Intelligence, National Aerospace University "Kharkiv Aviation Institute", Manka Vadyma St. 17, Kharkiv, Ukraine. 61070 e-mail: k.korobchinskiy@khai.edu* **https://orcid.org/0000-0002-3676-6070**

UML-Oriented Information Technology for Continuous Maximum Coverage Problems with Arbitrary-Shaped Objects

Relevance. Continuous maximum coverage problems with arbitrary-shaped objects play a crucial role in geographic information systems, monitoring platforms, logistics services, security systems, spatial data analysis, and decision-support solutions. The growing volume of data, dynamic environments, and high model complexity require formalized, modular, and scalable information technologies. UML, as a modeling standard, enables formal architectural descriptions of software solutions, ensuring reliability, reproducibility, and transparency of implementation.

Purpose. To develop a UML-oriented information technology for solving continuous maximum coverage problems that incorporates an architectural model, data structures, information flows, functional components, and UML specifications of modules supporting coverage-based systems.

Methods. The study employs object-oriented and structural modeling techniques, UML diagramming (Use Case, Class, Activity, Sequence, Component, Composite Structure, State Machine, Deployment), architectural design methods, principles of modularity, dependency inversion, component decomposition, and approaches used in building scalable information systems.

Results. A complete UML specification of the architecture of an information technology for maximum coverage problems has been constructed: external interaction scenarios, classes, components, operation sequences, system behavior and state logic, infrastructural links, and deployment structure have been defined. An integrated three-tier architecture (presentation, application logic, and data layers) has been formed. Principles for constructing modules for spatial analytics, optimization, coverage criterion computation, scenario management, visualization, and data interfaces have been described. The UML models provide a formalized structure that enables the development of scalable and reproducible IT solutions for coverage problems.

Conclusions. The developed information technology provides structural, behavioral, and architectural formalization of a maximum coverage system. UML-oriented modeling improves architectural transparency, reduces risks of integration errors, and ensures scalability and reusability of components. The obtained UML models may serve as a methodological foundation for building intelligent GIS platforms, optimization services, monitoring systems, and real-time analytical solutions.

Keywords: *UML, information technology, maximum coverage, software architecture, spatial data, modeling, optimization, GIS.*

Надійшла у першій редакції 27.08.2025, в останній - 29.09.2025.
The first version has been received on 27.08.2025, the final version - on 29.09.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-03

УДК 539.3+519.60

Гнітько Василь Іванович
к.т.н., старший науковий співробітник
Інститут енергетичних машин і систем ім. А.М. Підгорного НАН України, м. Харків, вул. Комунальників, 2/10, 61023
e-mail gnitkovi@gmail.com
https://orcid.org/0000-0003-2475-5486

Дегтярьов Кирило Георгійович
к.т.н., старший науковий співробітник
Інститут енергетичних машин і систем ім. А.М. Підгорного НАН України, м. Харків, вул. Комунальників, 2/10, 61023
e-mail: kdegt89@gmail.com;
https://orcid.org/0000-0002-4486-2468

Колодяжний Андрій Сергійович
аспірант
Інститут енергетичних машин і систем ім. А.М. Підгорного НАН України, м. Харків, вул. Комунальників, 2/10, 61023
e-mail:7ask7@ukr.net;
https://orcid.org/0000-0008-4026-6715

Крютченко Денис Володимірович
доктор філософії, науковий співробітник
Інститут енергетичних машин і систем ім. А.М. Підгорного НАН України, , м. Харків, вул. Комунальників, 2/10, 61023
e-mail: wollydenis@gmail.com;
https://orcid.org/0000-0003-6804-6991

Стрельнікова Олена Олександрівна
д.т.н., проф., провідний науковий співробітник
Інститут енергетичних машин і систем ім. А.М. Підгорного НАН України, , м. Харків, вул. Комунальників, 2/10, 61023
Харківський національний університет радіоелектроніки
e-mail: elena1@gmx.com;
https://orcid.org/0000-0003-0707-7214

Комп'ютерне моделювання плескань рідини в резервуарах з перегородками

Мета дослідження – розроблення числових методів дослідження стійкості руху в резервуарах за наявності перегородок різного типу.

Актуальність. Дослідження стійкості руху рідини в резервуарах із горизонтальними та вертикальними перегородками має важливе теоретичне та прикладне значення для багатьох галузей - від космічної та авіаційної техніки до морського та наземного зберігання рідин (паливо, технологічні рідини, хімічні реактиви). Наявність перегородок істотно змінює характер плескань: вони впливають на частотний спектр вільної поверхні, структуру вихорів, локалізацію енергії та виникнення резонансних режимів. Неправильне врахування цих ефектів може призвести до зниження безпеки, зростання динамічних навантажень на конструкцію та погіршення експлуатаційних характеристик системи. Експериментальні дослідження таких процесів часто є технічно складними, коштовними та потенційно небезпечними. Випробування на реальних об'ємах рідин потребують великих стендів, високих витрат на матеріали й обладнання, а також обґрунтованих заходів безпеки при роботі з паливно-агресивними або вибухонебезпечними середовищами. У зв'язку з цим розробка точних математичних моделей, чисельних алгоритмів і методів моделювання руху рідини в резервуарах із перегородками набуває особливої актуальності. Комп'ютерне моделювання дозволяє безпечно і відносно недорого дослідити широкий спектр режимів, виконати.

Методи дослідження. В роботі використані методи теорії потенціалу та сингулярних інтегральних рівнянь, методи граничних елементів, метод під-областей та метод заданих нормальних форм.

Результати. Отримані системи одновимірних сингулярних інтегральних рівнянь для визначення потенціалу швидкостей. Знайдені базисні функції, а саме форми коливань вільної поверхні, які надалі використано при розв'язанні задачі дослідження вимушений коливань. Проаналізовано вплив комбінованих горизонтальних і вертикальних навантажень на резервуари різної конструкції - як без перегородок, так і з вертикальними та горизонтальними перегородками. Виявлено області стійкого й нестійкого руху рідини. Встановлено, що наявність перегородок суттєво зменшує амплітуду коливань вільної поверхні рідини

Висновки. Отримані результати показали, що застосування горизонтальних і вертикальних перегородок істотно впливає на стійкість руху рідини в резервуарах, а саме приводить до суттєвого зменшення амплітуди коливань вільної

поверхні. Отримані дані можуть бути використані для підвищення надійності та безпеки резервуарних систем у різних галузях техніки, зокрема в авіаційній, космічній, морській та енергетичній.

***Ключові слова**, плескання рідини, резервуари з перегородками, метод підобластей, системи сингулярних інтегральних рівнянь .метод граничних елементів.*

Як цитувати: Гнітько В. І., Дегтярьов К. Г., Колодяжний А. С., Крютченко Д. В., Стрельнікова О. О. Комп'ютерне моделювання плескань рідини в резервуарах з перегородками. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. т. 67. С.35-44. https://doi.org/10.26565/2304-6201-2025-67-03

How to quote: V.I. Gnitko, K.G. Degtyarev, A.S. Kolodyazhny, D.V. Kriutchenko, and O.O. Strelnikova "Computer modeling of liquid sloshing in tanks with baffles" *Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol.67, pp.35-44, 2025. https://doi.org/10.26565/2304-6201-2025-67-03 [in Ukrainian]

1 Вступ

Динаміка рідини в резервуарах різного призначення — від паливних баків ракет-носіїв і літальних апаратів до ємностей для транспортування та зберігання рідин у промисловості — залишається актуальною науковою і прикладною проблемою. Одним із найбільш суттєвих факторів, що впливають на надійність і безпечну експлуатацію таких систем, є стійкість руху рідини під дією зовнішніх збурень. Коливання вільної поверхні (слосінг) здатні спричиняти додаткові динамічні навантаження на конструкцію, знижувати керованість транспортних засобів та підвищувати ризики аварійних ситуацій. Особливу складність становить аналіз резервуарів із перегородками, адже їхнє розташування та геометрія істотно змінюють картину руху рідини. При цьому експериментальні дослідження є вартісними й нерідко небезпечними, особливо при роботі з великими об'ємами, паливними чи агресивними середовищами. Тому розробка ефективних математичних і чисельних моделей плескань є актуальним науково-технічним завданням. Важливим аспектом забезпечення стійкості є врахування впливу демпфування. Правильний вибір коефіцієнтів затухання дає змогу істотно знизити амплітуди коливань і змістити межі стійкості системи. Практичним інструментом у цьому напрямі є врахування матриці Релея, що дозволяє адекватно описати дисипативні властивості системи. Крім того, одним із перспективних шляхів підвищення надійності є встановлення спеціальних демпфуючих пристроїв, таких як перегородки, плавучі кришки, та інші елементи конструкції, здатних зменшувати інтенсивність коливань і забезпечувати додатковий рівень безпеки. Таким чином, проблема дослідження стійкості руху рідини в резервуарах із горизонтальними та вертикальними перегородками з урахуванням впливу демпфування, вирішенню якої присвячена дана робота має як наукову, так і практичну цінність.

2 Постановка проблеми та огляд сучасного стану питання

Сучасні умови експлуатації техніки та поява нових конструкційних матеріалів суттєво впливають на напружено-деформований стан і вібраційні характеристики елементів конструкцій. Це зумовлює потребу в поглиблених дослідженнях міцносних та динамічних характеристик обладнання, яке працює під дією інтенсивних силових і температурних навантажень, при взаємодії з різними з рідиною або газом. Проблема гасіння коливань рідини в резервуарах стала особливо важливою ще у 1960-х роках у зв'язку з початком космічних польотів, коли неякісне проєктування систем паливних баків призводило до втрати стійкості та руйнування ракет-носіїв. Сьогодні створення потужних сучасних ракет вимагає нових підходів до конструювання паливних баків, які дедалі частіше мають складну або нетрадиційну форму [1-2]. Отже, дослідження стійкості руху рідини, гасіння коливань вільної поверхні в резервуарах і паливних системах залишаються актуальними впродовж останніх десятиліть [3–5]. Для аналізу міцності та вібраційних характеристик конструкцій застосовують сучасні чисельні методи, зокрема метод R-функцій [6], методи сингулярних та гіперсингулярних інтегральних рівнянь [7-8], метод граничних елементів (МГЕ) [9] скінченних елементів (МСЕ) [10], метод скінченних різниць [11], метод скінченних об'ємів [12] і метод поглинання [13]. Під час проєктування паливних баків інженери використовують різні способи зменшення коливань: внутрішні перегородки [14], вставки з піноматеріалів [15], повне [16] або часткове [17] покриття вільної поверхні, застосування новітніх матеріалів [18-19] і систем активного керування [20]. Усі ці рішення спрямовані на

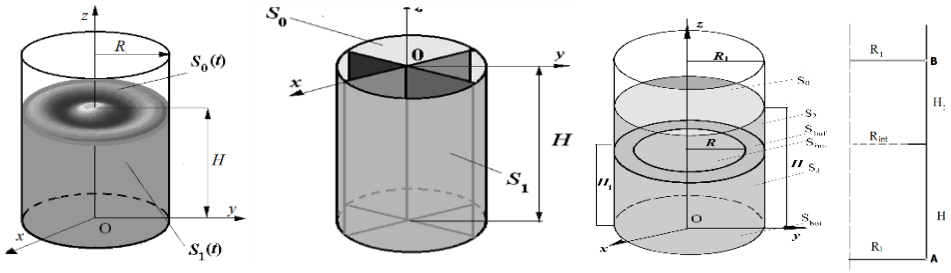
забезпечення достатнього рівня демпфування, що дозволяє зменшити амплітуди коливань і запобігти переходу системи в нестійкі режими. При цьому враховуються обмеження щодо маси конструкції, доступного простору та технологічних особливостей виготовлення, а також вимоги до надійності, довговічності й безпечної експлуатації обладнання в різних режимах роботи.

Таким чином, дослідження стійкості руху рідини з урахуванням демпфування в жорстких оболонках обертання з перегородками є актуальним завданням, результати якого можуть бути використані для підвищення надійності та ефективності роботи технічних систем.

3 Мета дослідження та формулювання задачі

Метою дослідження є побудова комп'ютерної методології для врахування демпфуючих ефектів у задачах аналізу стійкості руху рідини в обмежених областях (резервуарах, паливних баках) під дією періодичних зовнішніх навантажень, зокрема з урахуванням впливу внутрішніх перегородок як додаткових елементів гасіння коливань.

Розглянуто жорсткі оболонки обертання, що мають горизонтальні або вертикальні перегородки, та частково заповнені рідиною, рис.1.



а) б) в) г)

Рис. 3.1. Оболонки обертання з перегородками
Fig. 3.1. Shells of revolution with partitions

Формулювання задачі базується на традиційних припущеннях. Розглядається жорсткий резервуар, заповнений рідиною (густина 997 кг/м³), верхня незаповнена частина зайнята повітрям при сталому атмосферному тиску. Поверхневий натяг не враховується, оскільки для води при кімнатній температурі (густина $\rho_l \approx$ 1000 кг/м³, прискорення вільного падіння $g \approx$ 9.81 м/с², поверхневий натяг $\sigma \approx 0.072$ Н/м, характерна довжина $L \approx 0.5$ м) число Бонда $Bo = \rho g L^2/\sigma \approx 3.5 \times 10^4$. Тобто сили тяжіння значно перевищують сили поверхневого натягу. Вільна поверхня ідеалізується без урахування фазових переходів; випаровування та конденсації. Динамічні газові ефекти, включаючи стисливість, внутрішні потоки та змінний тиск, не розглядаються. Рідина вважається ізотермічною, при цьому теплові ефекти ігноруються, що є прийнятним наближенням за відсутності джерел тепла. В'язкі ефекти обмежуються тонкими прикордонними шарами і мають незначний вплив на власні частоти чи форми коливань, що обґрунтовує використання моделі потенційної течії, стандартної для аналізів першого порядку.

Як моделі резервуарів розглядаються циліндричні оболонки з перегородками та без них, рис.1. Рідина приймається нестисливою і нев'язкою, а її рух, зумовлений коливаннями стінок бака, вважається безвихровим.

Нехай Ω - обмежена область всередині резервуара, заповнена рідиною, S_0 – вільна поверхня рідини, S_1 - змочена поверхня оболонки, S_{baf} – поверхня перегородок. Рух рідини – безвихровий, тому існує потенціал швидкості $\Phi(\mathbf{x},t)$, $\mathbf{x}=(x,y,z)$, такий, що в області Ω задовольняє рівнянню Лапласа $\nabla^2\Phi(\mathbf{x},t) = 0$.

На жорстких нерухомих стінках і перегородках резервуару задано умови непроникності у вигляді

$$\left.\frac{\partial\Phi}{\partial\mathbf{n}}\right|_{S_1} = 0, \qquad \left.\frac{\partial\Phi}{\partial\mathbf{n}}\right|_{S_{baf}} = 0,$$

де $\mathbf{n}$ – одинична зовнішня нормаль до поверхні.

На вільній поверхні задано кінематичну і динамічну умови в лінійному наближенні

$$\frac{\partial\Phi}{\partial\mathbf{n}}\Big|_{S_0} = \frac{\partial\zeta}{\partial t},\ \frac{p-p_0}{\rho_l} = -\frac{\partial\Phi}{\partial t} - \big(g + a_v(t)\big)\zeta + a_h(t)x\big|_{S_0} = 0.$$

Тут p_0 – атмосферний тиск, p – тиск рідини, $a_v(t)$, $a_h(t)$ – прискорення сили, що змушує, в вертикальному та горизонтальному напрямках, $\zeta = \zeta(x,y,t)$ – невідома функція, що описує зміну рівня вільної поверхні з часом.

Для знаходження невідомих функцій Ф та ζ сформульовано таку крайову задачу

$$\nabla^2\Phi = 0, \frac{\partial\Phi}{\partial\mathbf{n}}\Big|_{S_1} = 0,\ \frac{\partial\Phi}{\partial\mathbf{n}}\Big|_{S_0} = \frac{\partial\zeta}{\partial t}, p - p_0|_{S_0} = 0 \tag{3.1}$$

Зобразимо потенціал Ф у вигляді ряду по власним формам коливань рідини в жорсткому резервуарі

$$\Phi = \sum_{k=1}^{M} \dot{d}_k(t)\varphi_k, \tag{3.2}$$

де $d_k(t)$ – невідомі коефіцієнти, які залежать лише від часу; φ_k – базисні функції; M – кількість форм, що утримуються при розрахунках.

Для функцій φ_k формулюємо крайові задачі таким чином:

$$\nabla^2\varphi_k = 0,\ \frac{\partial\varphi_k}{\partial\mathbf{n}}\Big|_{S_1} = 0,\qquad \frac{\partial\varphi_k}{\partial\mathbf{n}}\Big|_{S_0} = \frac{\partial\zeta}{\partial t};\quad \frac{\partial\varphi_k}{\partial t} + g\zeta = 0. \tag{3.4}$$

При цьому на вільній поверхні маємо співвідношення [21]

$$\frac{\partial\varphi_k}{\partial\mathbf{n}} = \frac{\chi_k^2}{g}\varphi_k, \tag{3.5}$$

Зауважимо, що співвідношення (3.4)-(3.5) зображують спектральну проблему [21], для розв'язання якої використано метод сингулярних інтегральних рівнянь [5].

4 Метод заданих форм. Зведення до систем інтегральних рівнянь

Для подальшого аналізу скористаємося представленням шуканих величин у зручній системі координат. Зокрема, розглянемо циліндричні координати, які природно відповідають геометрії задачі. Зобразимо невідомі функції Ф та ζ в циліндричних координатах (r,θ,z) у вигляді рядів:

$$\zeta(r,\theta,t) = \sum_{l=0}^{m}\cos(l\theta)\sum_{k=1}^{n} d_{kl}(t)\zeta_k(r), \tag{4.1}$$

$$\Phi(r,\theta,z,t) = \sum_{l=0}^{m}\cos(l\theta)\sum_{k=1}^{n_2}\dot{d}_{kl}(t)\varphi_k(r,z). \tag{4.2}$$

Тут $\varphi_k(r,z)$, $\zeta_k(r)$- базисні функції, між якими на вільній поверхні існує такий зв'язок [21]:

$$\zeta_{kJ}(r) = \frac{\partial\varphi_{kJ}(r,z)}{\partial z}\Big|_{S_0} = \frac{\chi_{kj}^2}{g}\varphi_{kj}(r,H). \tag{4.3}$$

Далі, згідно з [5], отримуємо розв'язувальну систему сингулярних інтегральних рівнянь для знаходження $\varphi_{kl}(r,z)$. Тут для спрощення індекси *kl* опущені.

$$2\pi\varphi(z_0,R) + \int_\Gamma \varphi(z,R)\Theta(z,z_0)r(z)d\Gamma - \frac{\chi^2}{g}\int_0^R \varphi(\rho,H)\Xi(P,P_0)\rho d\rho = 0,\ P_0 \in S_1, \tag{4.4}$$

$$2\pi\varphi\big(\rho_0,H\big) + \int_\Gamma \varphi(z,R)\Theta(z,z_0)r(z)d\Gamma - \frac{\chi^2}{g}\int_0^R \varphi(\rho,H)\,\Xi(P,P_0)\rho d\rho = 0,\ P_0 \in S_0,$$

Де ядра інтегральних операторів визначені формулами

$$\Theta(z,z_0) = 4/\sqrt{a+b}\left\{\frac{1}{2r}\left[\frac{r^2-r_0^2+(z_0-z)^2}{a-b}E_l(k) - F_l(k)\right]n_r + \frac{z_0-z}{a-b}E_l(k)n_z\right\}, \tag{4.5}$$

$$\Xi(P,P_0) = 4/\sqrt{a+b}\,F_l(k),\ a = r^2 + r_0^2 + (z-z_0)^2,\ b = 2rr_0.$$

Узагальнені еліптичні інтеграли в (4.5) обчислюються таким чином

$$E_l(k) = (-1)^l(1-4l^2)\int_0^{\pi/2}\cos(2l\beta\,\psi)\sqrt{1-k^2\sin^2\psi}\,d\psi,\ F_l(k) = (-1)^l\int_0^{\pi/2}\frac{\cos(2l\beta\psi)d\psi}{\sqrt{1-k^2\sin^2\psi}}. \tag{4.6}$$

Як показано в [5], в співвідношеннях (4.6) обираємо β = 1 при вивченні руху рідини в оболонках без перегородок та β = 2 для оболонок з вертикальними перегородками, рис. 3.1.б).

Після знаходження розв'язку спектральної задачі (4.4)-(4.6) отримуємо базисні функції $\varphi_{kl}(r,z)$ та $\zeta_k(r)$ та власні частоти ω_{kl}.

Далі перейдемо до задачі визначення частот і форм коливань оболонки обертання з горизонтальними перегородками, рис. 3.1в). Розділимо розрахункову область, що зайнята рідиною, на дві під-області. Введемо деякі позначення. Вільну поверхню позначимо як S_0. Під-області позначаємо як $\Omega_k (k = 1,2)$. Вводимо штучну поверхню S_{int}, яку називають також поверхнею інтерфейсу. Поверхні стінок оболонкової конструкції в k-тій під-області позначені як $S_k (k = 1,2)$, S_{bot} є поверхнею днища, та S_{baf} є поверхнею перегородки. Зауважимо, що розроблений підхід дозволяє розглядати довільну кількість перегородок.

Суть методу під-областей полягає в тому, що вплив кожної під-області на сусідню описується за допомогою матриці впливу. Вона встановлює зв'язок між значеннями потенціалів швидкостей на поверхнях інтерфейсу та відповідними потоками. Завдяки цьому розв'язувальна система інтегральних рівнянь формується лише відносно невідомих величин на вільній поверхні. Подальше розбиття межі рідинної області на граничні елементи приводить до системи лінійних алгебраїчних рівнянь для невідомих значень потенціалу при умові, що потоки задані.

Введемо також такі позначення для жорстких поверхонь, що обмежують під-області: $\sigma_1 = S_1 \cup S_{\text{bot}} \cup S_{\text{inf}} \cup S_{\text{baf}}$, $\sigma_2 = S_2 \cup c \cup S_{\text{baf}} \cup S_0$. Межі областей Ω_1 і Ω_2 є такими: $\sum_1 = S_{\text{baf}} \cup S_1 \cup S_{\text{bot}} \cup S_{\text{int}}$ та $\sum_2 = S_{\text{baf}} \cup S_2 \cup S_{\text{bot}} \cup S_0$, рис.3.1г). Позначимо значення потенціалу швидкостей у вузлах S_1, S_2 і S_0 як S_1, S_2, S_0, відповідно, та як w_1, w_2 значення функції $w = (\mathbf{U}, \mathbf{n})$ на границях $\sum_1, \sum_2$, потоки q_1, q_2 відомі з граничної умови непротікання, а на вільній поверхні невідомий потік позначається як q_0. Значення потенціалу швидкостей та потоку на поверхні інтерфейсу S_{int} будуть невідомими функціями

$$q_1 = \frac{\partial\varphi}{\partial\mathbf{n}}\Big|_{S_{\text{int}}}, \; q_2 = \frac{\partial\varphi}{\partial\mathbf{n}}\Big|_{S_{\text{int}}}, \; q_i \in \Sigma_i.$$

Маємо такі умови сумісності

$$\varphi_{2i} = \varphi_{1i}, \; q_1 = -q_2. \tag{4.7}$$

Розглянемо задачу визначення потенціалу швидкостей. Введемо позначення для поверхонь $\tilde{S}_1 = \sigma_1, \tilde{S}_2 = S_{int}, \tilde{S}_3 = \sigma_2, \tilde{S}_4 = S_0$ та матричні оператори

$$A(S,\sigma)\psi = \iint_S \psi \frac{\partial}{\partial\mathbf{n}} \frac{1}{|P-P_0|} d\sigma, \quad B(S,\sigma)\psi = \iint_S \psi \frac{1}{|P-P_0|} d\sigma, \tag{4.8}$$

З використанням (4.8) побудуємо матриці

$$A_{ij} = A(\tilde{S}_i, \tilde{S}_j), \; B_{ij} = B(\tilde{S}_i, \tilde{S}_j), \; i,j = \overline{1,4}.$$

Використовуємо метод під-областей (суперелементів) для визначення потенціалу φ [22].

$$\begin{gathered}
A_{11}\varphi_1 + A_{12}\varphi_{1i} = B_{11}w_1 + B_{12}q_1, \; P_0 \in \sigma_1, \\
A_{21}\varphi_1 + A_{22}\varphi_{1i} = B_{21}w_1 + B_{22}q_1, \; P_0 \in S_{int}, \\
A_{32}\varphi_{1i} + A_{33}\varphi_2 = B_{33}w_2 - B_{32}q_1 + B_{34}q_0, \; P_0 \in \sigma_2, \\
A_{22}\varphi_{1i} + A_{23}\varphi_2 = B_{23}w_2 - B_{22}q_1 + B_{24}q_0, \; P_0 \in S_{int}, \\
A_{42}\varphi_{1i} + A_{43}\varphi_2 = B_{43}w_2 - B_{42}q_1 + B_{44}q_0, \; P_0 \in S_0.
\end{gathered} \tag{4.9}$$

Зауважимо, що умови сумісності (4.7) враховані при отриманні системи (4.9). У результаті розв'язання системи (4.9) здобудемо

$$\varphi = \mathbf{Q}\boldsymbol{w}, \; \varphi = (\varphi_i)_{i=1}^2, \; \boldsymbol{w} = (w_i)_{i=1}^2, \; \mathbf{Q} = (Q_{ij})_{ij=1}^2,$$

де вирази для Q_{ij} отримані в [22].

5 Вільні коливання рідини в циліндричних оболонках.

Дослідимо вільні коливання рідини в жорсткій циліндричних оболонках без перегородок, з горизонтальними та вертикальними перегородками. Спочатку розглянемо жорстку циліндричну

оболонку з плоским дном, що має такі параметри: радіус $R = 1$ м, довжина $L = 2$ м. Нехай H – рівень заповнення рідиною. Горизонтальна перегородка є круглою пластиною з центральним отвором (кільцева перегородка) (див. рис. 3.1). Вертикальну координату перегородки (висоту розміщення перегородки) позначимо як H_1 ($H_1 < H$). Радіус поверхні інтерфейсу позначимо як R_{int}. Будемо мати $H = H_1 + H_2$, $R_{baf} = R - R_{int}$.

Числові результати отримано за допомогою методу граничних елементів. Використано 100 граничних елементів вздовж радіуса днища (N_b), 120 елементів уздовж змочених циліндричних частин (N_w) і 100 елементів вздовж радіусу вільної поверхні (N_0). Подальше збільшення кількості елементів не призводило до суттєвої зміни результатів. На поверхні інтерфейсу та перегородки використано різну кількість елементів в залежності від радіуса перегородки. При числовому моделюванні розглянуто різні значення для R_{int} та H_1. Застосовано аналітичний розв'язок [21] для порівняння та валідації числового розрахунку.

Для тестування суперелементного підходу розраховані власні частоти коливань рідини в резервуарі при встановленні перегородки на рівнях $H_1 = 0.5$ м, $H_1 = 0.9$ м при $R_{int} = 0.7$ м та $H = 1.0$ м. Порівняння результатів з обчислення частот, отриманих запропонованим методом граничних суперелементів, та аналітичних даних з роботи [23] подано в таблиці 5.1.

Таблиця 5.1. Порівняння частот коливань при $l = 0$.
Table 5.1. Comparison of vibration frequencies at l = 0.

Позиція перегородки H, м	Метод	ω_{01}	ω_{02}	ω_{03}	ω_{04}
0.5	під-областей	6.070120	8.293836	9.991324	11.43449
	[23]	6.072544	8.292653	9.989851	11.43277
0.9	під-областей	4.727280	7.798846	9.708979	11.21009
	[23]	4.735574	7.796959	9.708475	11.20921

Дані, наведені в таблиці 5.1, свідчать про достовірність запропонованого методу.

Як відомо [22], найнижчі частоти плескань відповідають хвильовому числу $l = 1$. В таблиці 5.2 наведені частоти коливань рідини при $l = 1$ в циліндричній оболонці без перегородок, а також з вертикальними перегородками та горизонтальними перегородками при H_1= 0.9м.

Таблиця 5.2. Частоти коливань при $l = 1$ для циліндричних оболонок.
Table 5.2. Vibration frequencies at l = 1 for cylindrical shells.

Тип резервуару	ω_{11}	ω_{12}	ω_{13}	ω_{14}	ω_{15}
Без перегородок, МГСЕ [21]	4.1424	7.2286	9.1472	10.7123	11.9624
	4.1424	7.2284	9.1472	10.7112	11.9616
З вертикальними перегородками	5.4582	8.1067	9.8791	11.2574	12.657
З горизонтальними перегородками, МГСЕ [21]	2.6350	6.6446	8.9661	10.6468	11.9876
	2.6350	6.6444	8.9661	10.6467	11.9874

Порівняння результатів свідчить про збіжність та ефективність запропонованого методу.

6 Вимушені коливання рідини в циліндричних оболонках

Нехай базисні функції $\varphi_k(r,z)$ вже визначені. Підставимо їх у формули (4.1) для потенціалу швидкості Ф та (4.2) для підйому вільної поверхні ζ. Отримані вирази використаємо в динамічній умові на S_0. У результаті на вільній поверхні одержуємо таке співвідношення

$$\sum_{l=0}^{m}\cos(l\theta)\sum_{k=1}^{n}\left[\ddot{d}_{kl}(t)+\omega_{kl}^2\left(1+\frac{a_v(t)}{g}\right)d_{kl}(t)\right]\varphi_{kl}(r,z)+a_h(t)r\cos\theta=0, z=\zeta. \quad (6.1)$$

Виконавши скалярне множення рівняння (6.1) на функції $\psi_{lk}(k=\overline{1,n}; l=\overline{0,m}$і застосувавши умову ортогональності власних форм [21], отримуємо таку систему звичайних диференціальних рівнянь другого порядку

$$\ddot{d}_{k0}(t) + \omega_{k0}^2 \left(1 + \frac{a_v(t)}{g}\right) d_{k0}(t) = 0, \tag{6.2}$$

$$\ddot{d}_{k1}(t) + \omega_{k1}^2 \left(1 + \frac{a_v(t)}{g}\right) d_{k1}(t) + a_h(t) F_{k1} = 0,\ F_{k1} = \frac{(r,\varphi_{k1})}{(\varphi_{k1},\varphi_{k1})},$$

$$\ddot{d}_{kl}(t) + \omega_{kl}^2 \left(1 + \frac{a_v(t)}{g}\right) d_{kl}(t) = 0, k = \overline{1,n}; l = \overline{2,m}.$$

Для однозначного розв'язання системи (6.2) необхідно задати початкові умови, тобто

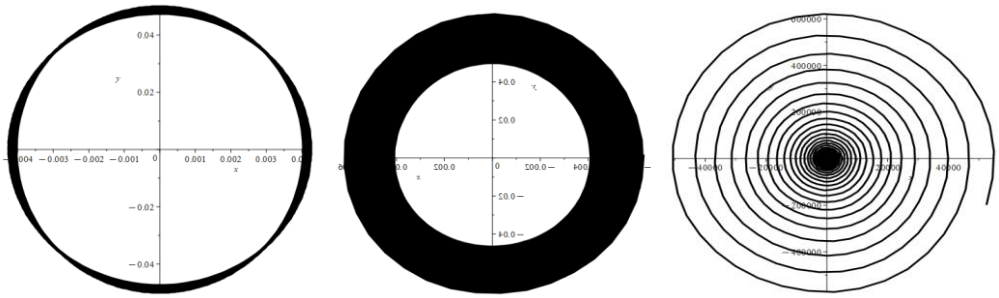
$$d_{kl}(t) = d_{kl}^0,\ \dot{d}_{kl}(t) = d_{kl}^1,\ k = \overline{1,n},\ l = \overline{0,m}. \tag{6.3}$$

Обирались зовнішні навантаження з такими прискореннями

$$a_x(t) = a_h \cos(\omega_h t)\,, a_z(t) = a_v \cos(\omega_v t) \tag{6.4}$$

при різних значеннях параметрів $a_v, \omega_v, a_h, \omega_h$.

Проведено розрахунки руху вільної поверхні за різні значення параметрів a_h, a_v та ω_h, ω_v. Спочатку розглянемо вертикальні навантаження. Фазові портрети рухів в координатах $(\zeta, \dot{\zeta})$ зображені на рис.6.1.



A B C

Рис.6.1. Фазові портрети руху рідини в резервуарі без перегородок при вертикальних навантаженнях
Fig. 6.1. Phase portraits of fluid motion in a tank without partitions under vertical loads

Тут рис. A) відповідає $a_h = 0.1, a_v = 1,\ \omega_v = \omega_\text{h} = 2.3$Гц, для рисунків B) та C) використано параметри $a_h = 0.1, a_v = 1,\ \omega_\text{h} = \omega_v = 4.1424$Гц та $a_h = 0, a_v = 1,\ \omega_v = 8.2848$ Гц, відповідно. З наведених результатів бачимо, що в перших двох випадках рухи є стабільними, але при $\omega_v = 8.2848$ Гц відбувається необмежене зростання амплітуди, що свідчить про параметричний резонанс (частота зовнішнього збурення збігається з подвоєною власною частотою).

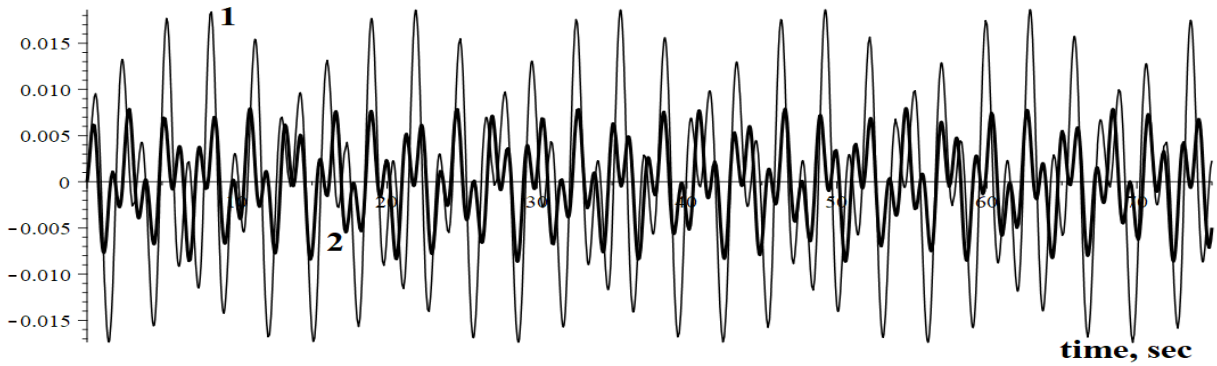


Рис.6.2. Залежність рівня підйому рідини в резервуарах при $\omega_v = \omega_\text{h} = 2.3$Гц
Fig. 6.2. Dependence of fluid level rise in tanks at $\omega_v = \omega_\text{h} = 2.3$ *Hz*

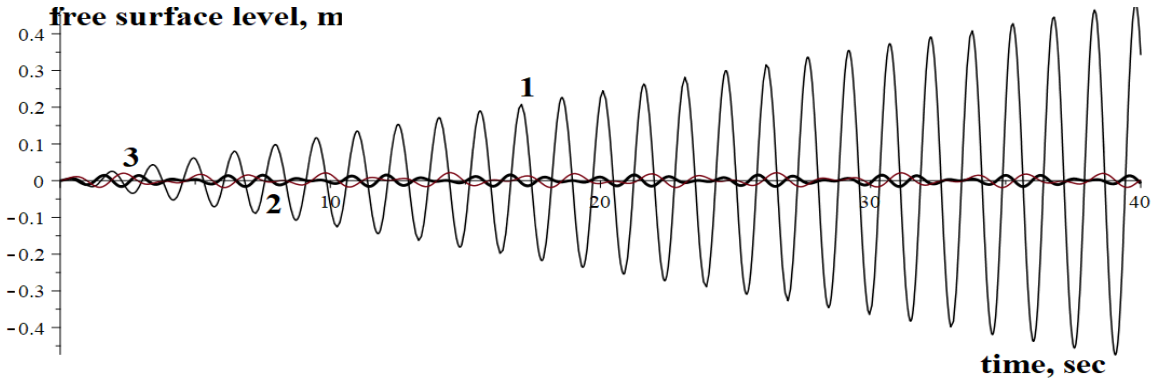


Рис.6.3. Залежність рівня підйому рідини в резервуарах при $\omega_v = \omega_h = 4.1424$Гц
Fig. 6.3. Dependence of fluid level rise in tanks at $\omega_v = \omega_h = 4.1424Hz$

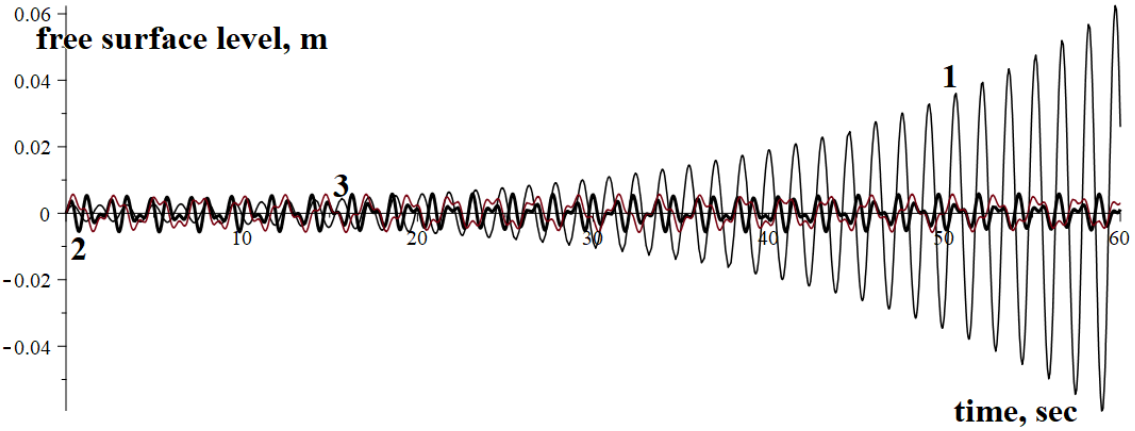


Рис.6.4. Залежність рівня підйому рідини в резервуарах при $\omega_v = \omega_h = 8.2848$ Гц
Fig. 6.4. Dependence of fluid level rise in tanks at $\omega_v = \omega_h = 8.2848$ Hz

На рис. 6.2-6.4 зображено зміну рівня вільної поверхні у точці $(\theta = 0, r = R)$. Лінії, позначені (1), відповідають коливанням вільної поверхні в резервуарі без перегородок, лінії, позначені (2), описують зміну рівня вільної поверхні в резервуарі з вертикальними перегородками, криві, позначені (3), відповідають зміні рівня вільної поверхні за часом для резервуару з горизонтальною перегородкою. Зазначимо, що у всіх випадках відмічається зменшення амплітуди коливань вільної поверхні рідини при встановленні перегородок. Винятком є випадок, коли розглядається резервуар з горизонтальною перегородкою при частоті коливань сили, що змушує, яка дорівнює 2.3Гц, що є досить близькою до першої фундаментальної частоти, яка дорівнює 2.635Гц, але рух рідини при цьому залишається стабільним.

Висновки

Проведене дослідження підтвердило, що наявність горизонтальних та вертикальних перегородок у резервуарах істотно впливає на динаміку рідини, зокрема на стійкість її руху та характер вільної поверхні. Розроблені чисельні методи, зокрема на основі теорії потенціалу, сингулярних інтегральних рівнянь та методу граничних елементів, дозволили отримати точні результати без необхідності проведення складних та дорогих експериментів. Встановлено, що застосування перегородок дає змогу значно зменшити амплітуду коливань вільної поверхні, що сприяє зниженню динамічних навантажень на стінки резервуару та підвищенню стійкості всієї системи. Визначено області стійкої та нестійкої поведінки рідини залежно від. Отримані результати мають високу прикладну цінність і можуть бути використані при проєктуванні резервуарів у таких критичних галузях, як авіація, космонавтика, енергетика та морський транспорт, де важлива надійність та безпечність роботи з рідинами.

REFERENCES

1. L. Liu, J. Li Dynamic (2022). Deformation and Perforation of Ellipsoidal Thin Shell Impacted by Flat-Nose Projectile, *Materials*, Vol. 15(12), 4124, , DOI:10.3390/ma15124124
2. A. Karaiev, E. Strelnikova, (2020). Liquid Sloshing in Circular Toroidal and Coaxial Cylindrical Shells. In: Ivanov, V., Pavlenko, I., Liaposhchenko, O., Machado, J., Edl, M. (eds) Advances in

Design, Simulation and Manufacturing III. DSMIE 2020. Lecture Notes in Mechanical Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-50491-5_1

3. O.-M. Balas C. V. Doicin and E. C. Cipu, (2023). Analytical and Numerical Model of Sloshing in a Rectangular Tank Subjected to a Braking, *Mathematics*, vol. 11, P. 949-955, DOI:10.3390/math11040949
4. E. Gani, S. Öztürk, A. Sari (2025). Effects of Liquid Sloshing in Storage Tanks: An Overview of Analytical, Numerical, and Experimental Studies. *Int J Steel Struct* , vol. 25, pp. 544–556, https://doi.org/10.1007/s13296-025-00946-8.
5. E. Strelnikova, D. Kriutchenko, V. Gnitko, A. Tonkonozhenko, (2020).Liquid Vibrations in Cylindrical Tanks with and Without Baffles Under Lateral and Longitudinal Excitations, *International Journal of Applied Mechanics and Engineering*, Vol. 25, Issue 3, P. 117-132, DOI: 10.2478/ijame-2020-0038.
6. S.M. Lamtiuhova. (2025). Mathematical Modeling of Steady Flow Past Circular Cylinder with Splitter Plates by R-Functions Method, *International Journal of Mathematics and Physics*, DOI: 10.26577/ijmph.202516110.
7. V.I. Gnitko, A.O. Karaiev, K.G. Degtyariov, I.A. Vierushkin, E.A. Strelnikova. (2022). Singular and hypersingular integral equations in fluid–structure interaction analysis. *WIT Transactions on Engineering Sciences*, Vol.134, pp.67 – 79,. DOI:10.2495/BE450061
8. E. Strelnikova, N. Choudhary, K. Degtyariov, D. Kriutchenko, I Vierushkin. Boundary element method for hypersingular integral equations: Implementation and applications in potential theory. *Engineering Analysis with Boundary Elements*, vol. 169, 2024, 105999, https://doi.org/10.1016/j.enganabound.2024.105999
9. T. Medvedovskaya, E. Strelnikova, K. Medvedyeva. (2015). Free Hydroelastic Vibrations of Hydroturbine Head Covers. *Intern. J. Eng. and Advanced Research Technology (IJEART).* 1(1) pp 45 - 50. DOI 10.13140/RG.2.1.3527.4961.
10. N. Smetankina and V. Pavlikov (2021) Mathematical Model of the Stress State of the Antenna Radome Joint with the Load-Bearing Edging of the Skin Cutout, *ICoRSE 2021. Lecture Notes in Networks and Systems*, vol. 305, pp. 287–295. https://doi.org/10.1007/978-3-030-83368-8_28
11. K.Murawski, (2020). Technical Stability of Very Slender Rectangular Columns Compressed by Ball-And-Socket Joints without Friction, *Int. Journal of Structural Glass and Advanced Materials Research*, vol, 4(1), pp. 186-208, DOI: 10.3844/sgamrsp.2020.186.208
12. P. Lampart, A. Rusanov, S. Yershov, S. Marcinkowski, A. Gardzilewicz, (2005).Validation of a 3D BANS solver with a state equation of thermally perfect and calorically imperfect gas on a multi-stage low-pressure steam turbine flow, *Journal of Fluids Engineering, Transactions of the ASME*, vol. 127(1), pp. 83–93,2005. DOI: 10.1115/1.185249.
13. C.Tong, Y. Shao, H. B. Bingham, & FC. W. Hanssen, (2021). An Adaptive Harmonic Polynomial Cell Method with Immersed Boundaries: Accuracy, Stability and Applications. *International Journal for Numerical Methods in Engineering*, , Vol. 122, P. 2945–2980. https://doi.org/10.1002/nme.6648.
14. E. Strelnikova, D. Kriutchenko, V. Gnitko, A. Tonkonozhenko, (2020).Liquid Vibrations in Cylindrical Tanks with and Without Baffles Under Lateral and Longitudinal Excitations, *International Journal of Applied Mechanics and Engineering*, Vol. 25, Issue 3, P. 117-132, DOI: 10.2478/ijame-2020-0038.
15. S. K. Poguluri, Il H. Cho, (2023).Effect of vertical porous baffle on sloshing mitigation of two-layered liquid in a swaying tank, *Ocean Engineering*, vol. 289, Part 1, 115952, https://www.sciencedirect.com/science/article/pii/S0029801823023363
16. N. Choudhary, S.N. Bora and E. Strelnikova, (2021). Study on liquid sloshing in an annular rigid circular cylindrical tank with damping device placed in liquid domain, *J. Vib. Eng. Tech.*, vol. 9, pp. 1–18, DOI:10.1007/s42417-021-00314-w
17. N. Choudhary, N. Kumar, E. Strelnikova, V. Gnitko, D. Kriutchenko, K. Degtyariov, (2021). Liquid vibrations in cylindrical tanks with flexible membranes. *Journal of King Saud University – Science*, vol. 33(8), 101589, doi.org/10.1016/j.jksus.2021.101589.
18. E. Sierikova, E. Strelnikova, V. Koloskov, K. Degtyarev. (2021). The effective elastic parameters determining of threedimensional matrix composites with nanoinclusions. *Problems of Emergency Situations: Proc. of International Scientific-practical Conference. Kharkiv: NUCDU*, pp. 327–328, http://repositsc.nuczu.edu.ua/handle/123456789/13026

19. K. Degtyariov, V. Gnitko, Y. Kononenko, D. Kriutchenko, O. Sierikova, E. Strelnikova. (2022). Fuzzy methods for modelling earthquake induced sloshing in rigid reservoirs. *2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek)*, pp. 1-6, DOI: 10.1109/KhPIWeek57572.2022.9916466
20. M. Konopka, F., De Rose, H. Strauch, C. Jetzschmann, N. Darkow, J. Gerstmann, (2019). Active slosh control and damping - Simulation and experiment, *Acta Astronautica*, vol. 158, pp. 89 - 102, https://doi.org/10.1016/j.actaastro.2018.06.055.
21. I. A. Raynovskyy and A. N. Timokha. (2020). Sloshing in Upright Circular Containers: Theory, Analytical Solutions, and Applications, *CRC Press/Taylor and Francis Group*, DOI: 0.1201/9780429356711.
22. Strelnikova, E., Kriutchenko, D., Gnitko, V., Tonkonozhenko, A.: Liquid Vibrations in cylindrical tanks with and without baffles under lateral and longitudinal excitations. Int. J. Appl. Mech. Eng. **25**(3), 117–132 (2020). https://doi.org/10.2478/ijame-2020-0038

Gnitko Vasyl *PhD, senior researcher*
Anatolii Pidhornyi institute of power machines and systems
vul. Komunalnykiv, 2/10, Kharkiv, 61046, Ukraine
Degtyarev Kirill *PhD, senior researcher*
Anatolii Pidhornyi institute of power machines and systems
vul. Komunalnykiv, 2/10, Kharkiv, 61046, Ukraine
Kolodiazhny Andriy *Post-graduate student*
Anatolii Pidhornyi institute of power machines and systems
vul. Komunalnykiv, 2/10, Kharkiv, 61046, Ukraine

Kriutchenko Denys *PhD, researcher*
Anatolii Pidhornyi institute of power machines and systems
vul. Komunalnykiv, 2/10, Kharkiv, 61046, Ukraine
Strelnikova Elena *Leading researcher*
Anatolii Pidhornyi institute of power machines and systems
vul. Komunalnykiv, 2/10, Kharkiv, 61046, Ukraine

Computer modeling of liquid sloshing in tanks with baffles.

.**Research Objective.** The objective of this study is to develop numerical methods for analyzing the stability of fluid motion in tanks equipped with various types of internal baffles.**Relevance.** The investigation of fluid motion stability in tanks with horizontal and vertical baffles is of significant theoretical and practical importance for many fields — from aerospace and aviation to marine and ground-based liquid storage (e.g., fuels, process fluids, chemical reagents). The presence of baffles substantially alters the sloshing behavior: they affect the frequency spectrum of the free surface, vortex structures, energy localization, and the emergence of resonant modes. Improper consideration of these effects may lead to reduced safety, increased dynamic loads on the structure, and degraded performance of the overall system. Experimental studies of such processes are often technically complex, costly, and potentially hazardous. Testing real liquid volumes requires large-scale facilities, high material and equipment expenses, as well as rigorous safety measures when dealing with flammable, aggressive, or explosive substances. Therefore, the development of accurate mathematical models, numerical algorithms, and simulation methods for fluid motion in baffled tanks is of particular relevance. Computer-based modeling provides a safe and relatively low-cost means to explore a wide range of fluid behavior regimes.

Research Methods. The study employs methods from potential theory and singular integral equations, the boundary element method (BEM), the subdomain method, and the method of prescribed normal forms.

Results. Systems of one-dimensional singular integral equations were derived to determine the velocity potential. Basis functions were obtained, specifically the free surface oscillation modes, which were then used to solve the problem of forced oscillations. The influence of combined horizontal and vertical excitations was analyzed for tanks of various designs — both without baffles and with vertical or horizontal baffles. Regions of stable and unstable fluid motion were identified. It was found that the presence of baffles significantly reduces the amplitude of free surface oscillations.**Conclusions.** The obtained results demonstrated that the use of horizontal and vertical baffles has a significant impact on the stability of fluid motion in tanks, specifically by considerably reducing the amplitude of free surface oscillations. The data obtained may be applied to improve the reliability and safety of tank systems across various engineering domains, particularly in aviation, space, marine, and energy industries.

Keywords: *liquid sloshing, baffled tanks, subdomain method, systems of singular integral equations, boundary element method, damping, Ains–Strett diagram.*

Надійшла у першій редакції 21.08.2025, в останній - 01.10.2025.
The first version has been received on 21.08.2025, the final version – on 01.10.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-04

УДК (UDC) 621.382.002:621.381.821

Horenko Daniiel *Master student of the Institute of Computer Science and Artificial Intelligence, V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv, Ukraine, 61022*
e-mail: horenko2020ku11@student.karazin.ua
https://orcid.org/0009-0004-6910-4622

Kotvytskiy Albert *Candidate of Physical and Mathematical Sciences, Associate Professor, V. N. Karazin Kharkiv National University 4, Svobody Sq., Kharkiv, 61022, Ukraine*
Pavol Jozef Šafárik University in Košice, 2, Šrobárova, Kosice, 041 80, Slovak Republic
e-mail: kotvytskiy@gmail.com;
https://orcid.org/0000-0001-8283-505X

Controlling LEDC timers of the ESP32 microcontroller using registers

Relevance. This paper examines precise generation and control of pulse-width modulation (PWM) signals using the LEDC (LED PWM Controller) subsystem of the ESP32 microcontroller via direct register access. As embedded real-time systems increasingly require fine timing control in LED drivers, motor control and power electronics, standard high-level driver APIs can be insufficient. Direct register manipulation of LEDC enables more precise tuning of frequency, resolution and pulse timing, which is critical for synchronization-sensitive applications.

Objective. To analyze the capabilities of ESP32 LEDC timers when configured through direct register writes, to experimentally evaluate the accuracy and stability of generated PWM signals across representative configurations, and to provide practical recommendations for optimizing LEDC parameters in applied embedded projects.

Methods. The investigation employed low-level register programming under Espressif's ESP-IDF on an ESP32-DevKitC V4 (WROOM-32D). Time-domain characteristics of the PWM outputs were measured with a Logic Analyzer (24 MHz sampling, 8 channels). The study combined theoretical derivations of PWM frequency and period based on clock source, divider (DIV) and counter resolution (RES) with implementation of direct register sequences to configure HSTIMER0 and HS channel 0, and comparative measurements for eighteen distinct configurations covering multiple RES, DIV and DUTY values.

Results. The register-based control method enabled generation of high-frequency PWM in the MHz range with close agreement between calculated and measured values. Across tested configurations the maximum relative deviation did not exceed ±0.03% for frequency and period, and ±0.6% for pulse high-time (duty width). Increasing counter resolution improved duty-cycle granularity, while the prescaler DIV produced a linear change in PWM frequency. The experimental limitations observed at the highest frequencies are attributable to the finite sampling capability of the measurement equipment.

Conclusions. Direct register access to the LEDC allows for obtaining deterministic, high-precision PWM signals with minimal parameter update latency, making them suitable for applications in robotics, power electronics, and other systems with high synchronization requirements. Further research is recommended on the influence of alternative clock sources, low-speed LEDC modes, integration with ISR/FreeRTOS, and extending the approach to other timers and channels.

Keywords: *ATmega, ESP32, LEDC, PWM, register access, logic analyzer*

How to quote: D. V. Horenko, and A. T. Kotvytskiy, "Controlling LEDC timers of the ESP32 microcontroller using registers" *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol.67, pp. 45-55, 2025. https://doi.org/10.26565/2304-6201-2025-67-04

Як цитувати: Horenko D. V., and Kotvytskiy A. T., Controlling LEDC timers of the ESP32 microcontroller using registers. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління.* 2025. 67. С.45-55. https://doi.org/10.26565/2304-6201-2025-67-04

Introduction

In modern technological systems, embedded computing modules are becoming increasingly important, providing intelligent process control, data acquisition, and communication between devices. They are the basis of robotic systems, automated production lines, sensor networks, and consumer IoT solutions. A high level of integration, compactness, and energy efficiency makes such systems a key

element of modern electronics [1], [2]. One of the fundamental tasks for embedded controllers remains the precise generation and control of signals, particularly pulse-width modulated signals, which control LEDs, motors, power switches, and other peripheral devices.

Traditionally, for educational and prototyping purposes, microcontrollers of the ATmega series, particularly the ATmega328P and ATmega2560, which are the foundation of the Arduino Uno and Arduino Mega platforms, are widely used. They are distinguished by their simple architecture, extensive library support, and a user-friendly programming environment. However, the use of the high-level Wiring language and the abstraction layers of the Arduino IDE significantly limits the possibilities for precise configuration of timers and pulse-width modulation [3], [4]. In specialized developments, to achieve maximum configuration flexibility, direct access to the microcontroller's registers is used, which allows changing PWM parameters with minimal delay [5].

In recent years, ESP32 microcontrollers from Espressif Systems have become widespread, combining high computational power, built-in Wi-Fi and Bluetooth interfaces, hardware timer modules, and advanced PWM signal control capabilities.

In the microcontroller platform market, the most common boards are the Arduino UNO (ATmega328P) and Arduino MEGA (ATmega2560), with official prices as of October 2025 of 29.30 EUR and 52.80 EUR, respectively, according to the Arduino Store [6], [7].

In contrast, the ESP32-DevKitC module, which is an official product of Espressif Systems, is available in the manufacturer's official store on the AliExpress platform [8] for a price of 8–15 USD, which is several times cheaper while offering significantly higher computing power and integrated wireless interfaces (Wi-Fi, Bluetooth).

Thus, for a comparable price, the user gets a dual-core 32-bit processor with a clock speed of up to 240 MHz, 16 PWM channels, and an advanced clocking system. This makes the ESP32 a cost-effective choice for developers of real-time systems and researchers in the field of precision electronics.

Software development for the ESP32 is possible both in the Arduino IDE environment and using ESP-IDF – the official SDK from Espressif Systems. However, working through the Arduino layer, which is built on the Wiring language, creates additional delays in function calls and conceals the low-level mechanisms for accessing hardware [9]. That is why for tasks related to high-frequency processes, motor control, or the study of timing characteristics, programming in C with direct writes to peripheral registers is advisable, as it allows for achieving maximum performance and precision in signal control.

Among the peripheral subsystems of the ESP32, a special place is held by the LEDC (LED PWM Controller) – a module for generating pulse-width modulated signals, capable of forming up to 16 independent PWM channels with support for high-speed (up to 40 MHz) and low-speed modes. PWM modulation is a basic tool for controlling LEDs, electric motors, audio modules, and other loads, where the stability and precision of the signal parameters determine the operational quality of the entire system:

- in LED drivers – regulating brightness without flickering;
- motor control systems – smoothness of rotation and torque precision;
- in digital audio interfaces – affecting the level of noise and harmonics;
- in synchronization generators – minimization of time fluctuations of the signal.

The use of standard APIs, particularly the ESP-IDF LEDC driver, significantly simplifies programming but does not allow for full control over the timer registers. This limits the precision of configuring the frequency, duty cycle, and the timing of the signal update. In specialized systems, such as robotic controllers or precision power control circuits, such capabilities are insufficient. In these cases, an effective solution is direct access to the LEDC registers, which allows for flexible modification of all necessary parameters, eliminating the overhead of the driver layer.

The purpose of this study is to analyze the capabilities of controlling the LEDC timers of the ESP32 microcontroller through direct access to its registers, and to experimentally evaluate the accuracy and stability of the generated PWM signals.

2 Architecture and Operating Principle of the ESP32 LEDC

In classic microcontrollers of the ATmega family (specifically, ATmega328P, ATmega2560), the implementation of pulse-width modulation (PWM) is based on linking each timer to strictly defined output channels and pins of the chip [10]. For example, Timer0 in the ATmega328P serves only the OC0A and OC0B outputs, and these pins are hardware-fixed. This approach simplifies the hardware logic but significantly limits the flexibility of channel allocation when designing complex control systems [11].

In contrast, the ESP32 microcontroller implements the principle of hardware decoupling between the timer, the channel, and the GPIO output. Each of the 16 channels of the LEDC subsystem can be independently assigned to any of the four available timers and, in turn, routed to any available pin of the microcontroller. This architecture provides exceptional flexibility when implementing complex control systems, such as multi-channel motor drivers, RGB matrices, or power management systems, where not only timing precision but also the efficient use of GPIOs is critical.

Thus, the ESP32 allows for the programmatic reconfiguration of the logical mapping between channels and timers without hardware changes, which significantly expands design possibilities. Furthermore, unlike in AVR, the LEDC features a separation between the logical control level (channel) and the periodicity generator (timer).

- The timer determines the frequency and resolution of the PWM signal.
- The channel is responsible for the duty cycle and the connection to a specific GPIO.

This allows a single timer to be used for multiple channels that will have the same frequency but different duty cycles. Unlike classic AVRs, where the pulse typically begins upon a counter reset, the LEDC architecture allows setting an arbitrary pulse start time using the special HPOINT register, providing flexible control over the phase shift between channels.

2.1 Brief Hardware Context of the ESP32

The ESP32 [12] microcontroller, developed by Espressif Systems, is a high-performance integrated platform for building embedded systems with support for wireless technologies. Its core is a dual-core Tensilica Xtensa LX6 processor, which operates at a clock frequency of up to 240 MHz. The processing cores feature an advanced power-saving system, allowing for dynamic performance adjustments based on the application's needs.

Fig. 1 ESP32 model Wroom-32D
Рис. 1 ESP32 моделі Wroom-32D

The ESP32 is distinguished by a rich set of peripheral modules, including:

- Communication interfaces (UART, SPI, I²C, I²S, CAN),
- Analog-to-digital (ADC) and digital-to-analog (DAC) conversion blocks,
- General-purpose hardware timers,
- Cryptographic accelerators,
- Specialized modules for signal control.

2.2 Structure of the LEDC Subsystem

LEDC is a hardware PWM controller with 16 independent channels that can generate PWM signals of various frequencies and duty cycles. Key features of the LEDC include:

– Support for high-speed and low-speed modes
– High precision (fractional frequency division)
– Fading (automatic duty cycle change)

LEDC has two logical classes of channels:

High-Speed (HS) — 8 channels designed for high-speed PWM generation (using timers HSTIMER0..3);

Low-Speed (LS) — 8 channels oriented towards low-frequency or power-saving modes (using timers LSTIMER0..3).

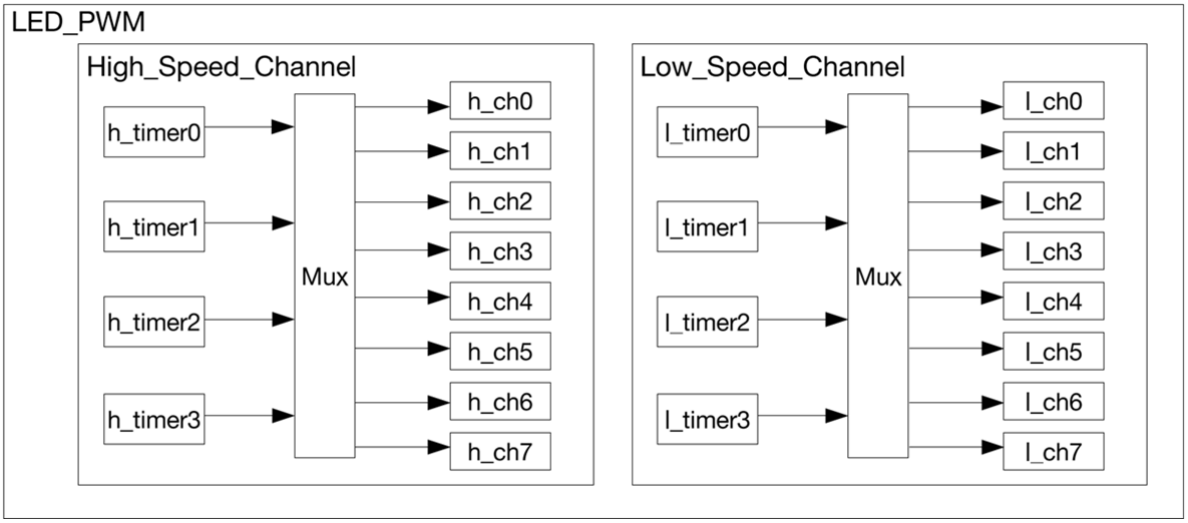


Fig. 2 LED_PWM architecture
Рис. 2 LED_PWM архітектура

As you can see, each output channel (signal), for example h_chn, can operate from any HSTIMERx timer.

2.3 LEDC Registers

The LEDC subsystem of the ESP32 microcontroller manages the pulse-width modulation channels through a set of specialized registers. It is the manipulation at this register level that provides maximum flexibility in configuring signal parameters and minimizing delays.

Let's examine the structure of the high-speed timer and channel.

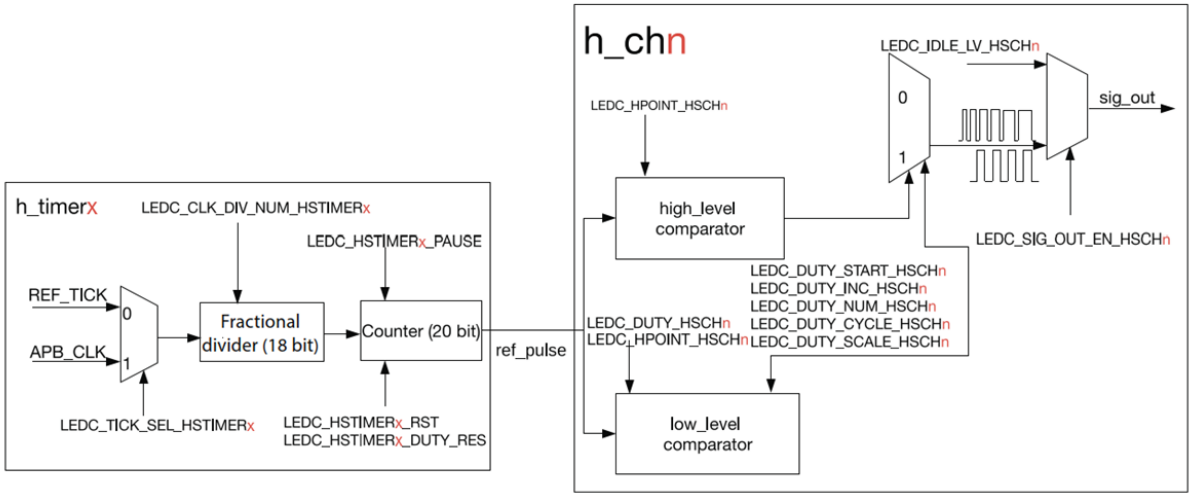


Fig. 3 LED_PWM High-Speed Channel Diagram
Рис. 3 LED_PWM Діаграма високошвидкісного каналу

From this structure, it is clear that the following need to be configured:

1. The h_timerx timer,
2. The h_chn channel,

3. The link between the channel and the timer.

The LEDC_HSTIMERx_CONF_REG register is responsible for configuring the parameters of the high-speed timer and has the following structure:

(reserved)	LEDC_TICK_SEL_HSTIMERx	EDC_HSTIMERx_RST	LEDC_HSTIMERx_PAUSE	LEDC_CLK_DIV_NUM_HSTIMERx		LEDC_HSTIMERx_DUTY_RES
				10 bits of the integer part	8 bits of fractional part	
31 26	25	24	23	22 21 20 19 18 17 16 15 14 13	12 11 10 9 8 7 6 5	4 3 2 1 0
0 0	0	1	0			

Fig. 4 Structure of the LEDC_HSTIMERx_CONF_REG register
Рис. 4 Структура регістру LEDC_HSTIMERx_CONF_REG

LEDC_TICK_SEL_HSTIMERx (bit 25) This bit is used to select the clock source for high-speed timer x, either APB_CLK or REF_TICK.

1: APB_CLK (80 MHz) This is the main peripheral bus of the ESP32. Selecting APB_CLK is typical for tasks requiring the generation of high-frequency PWM signals.

0: REF_TICK (1 MHz) This is a stable but slower reference clock source. REF_TICK is ideal for low-frequency applications.

LEDC_HSTIMERx_RST (bit 24) This bit is used to reset high-speed timer x. The counter value will be "zero" after the reset.

LEDC_HSTIMERx_PAUSE (bit 23) This bit is used to pause the counter in high-speed timer x.

LEDC_CLK_DIV_NUM_HSTIMERx (bits 22–5) This field is used to configure the division factor for the clock divider in high-speed timer x.

The upper 10 bits (22–13): Define the integer part of the divider.

The lower 8 bits (12–5): Define the fractional part of the divider (with a precision of 1/256).

LEDC_HSTIMERx_DUTY_RES (bits 4–0) This field is used to define the bit-width of the counter (from 1 to 20 bits), which is the number of steps in the PWM period for high-speed timer x.

The choice of bit resolution is a trade-off: higher resolution provides smoother control (e.g., of an LED's brightness) but lowers the maximum possible PWM frequency, and vice versa. This relationship is clearly demonstrated by formula (2.1), which integrates all the parameters discussed.

The PWM frequency is calculated by the formula:

$$f_{PWM} = \frac{f_{CLK}}{DIV \cdot 2^{RES}} \quad (2.1)$$

where f_{CRK} – is the clock source (APB_CLK або REF_TICK),

DIV – is the divider,

RES – is the counter resolution.

After configuring the HSTIMER0 timer, the output channel must be configured. The parameters for an individual channel of a high-speed timer x are defined by the LEDC_HSCHn_CONF0_REG register.

(reserved)	LEDC_IDLE_LV_HSCHn	LEDC_SIG_OUT_EN_HSCHn	LEDC_TIMER_SEL_HSCHn
31 4	3	2	1 0
0	0	0	0 0

Fig. 5 Structure of the LEDC_HSCHn_CONF0_REG register
Рис. 5 Структура регістру LEDC HSCHn_CONF0_REG

The register has the following fields:

LEDC_IDLE_LV_HSCHn (bit 3) – determines the signal level on the output when the channel is inactive.

0 — The output is LOW (0 V).

1 — The output is HIGH (3.3 V).

LEDC_SIG_OUT_EN_HSCHn (bit 2) — enables the signal output to the GPIO.

0 — Disables the PWM output (regardless of the timer's operation).

1 — Enables the PWM output (if the timer is running).

LEDC_TIMER_SEL_HSCHn (bits 0-1) — determines which timer controls the PWM channel. The ESP32 has 4 high-speed timers (HSTIMER0 – HSTIMER3).

To describe the relationship between the channel and the timer, it is first necessary to understand how PWM works. The signal is generated as follows:

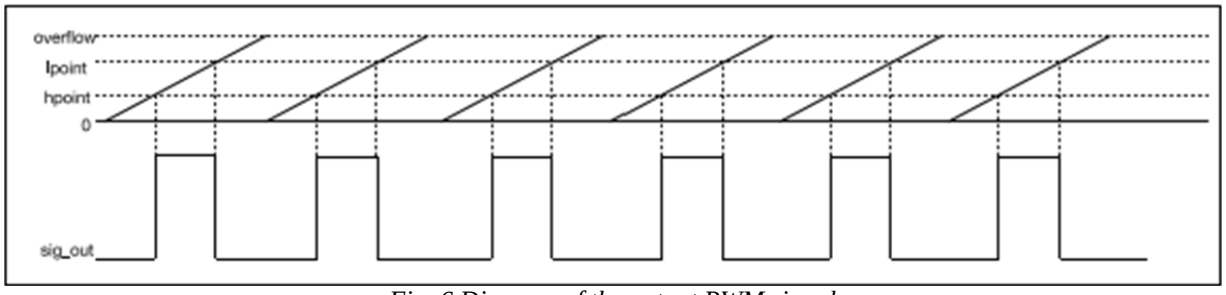


Fig. 6 Diagram of the output PWM signal
Рис. 6 Діаграма вихідного ШІМ сигналу

Each PWM channel receives a 20-bit value from the counter associated with the selected high-speed timer. This value is compared with two registers to generate the signal:

1. LEDC_HPOINT_HSCHn — when the counter reaches this value, the PWM output signal goes HIGH.
2. LEDC_HPOINT_HSCHn + LEDC_DUTY_HSCHn[24:4] — when the counter reaches this sum, the PWM signal returns to LOW.

Thus, HPOINT defines the start time of the pulse, and DUTY sets the duration of the HIGH level, forming the desired duty cycle.

To set the HPOINT value, the LEDC_HSCHn_HPOINT_REG register is used.

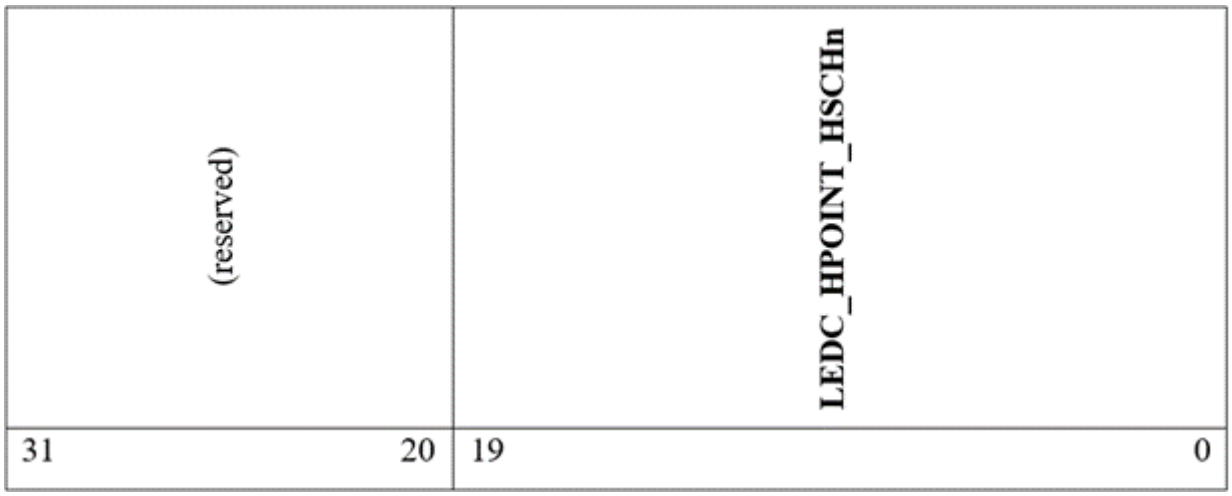


Fig. 7 Structure of the LEDC_HSCHn_HPOINT_REG register
Рис. 7 Структура регістру LEDC_HSCHn_HPOINT_REG

The LEDC_HPOINT_HSCHn field (bits 19 - 0) defines the condition under which the output signal will switch to 1 (HIGH).

This happens when the value of the timer counter, LEDC_HSTIMERx_CNT in the LEDC_HSTIMERx_VALUE_REG register, matches the value of LEDC_HPOINT_HSCHn in the LEDC_HSCHn_HPOINT_REG register. At this moment, a logical one (HIGH) is set on the corresponding PWM output.

To set the DUTY value, the LEDC_HSCHn_DUTY_REG register is used.

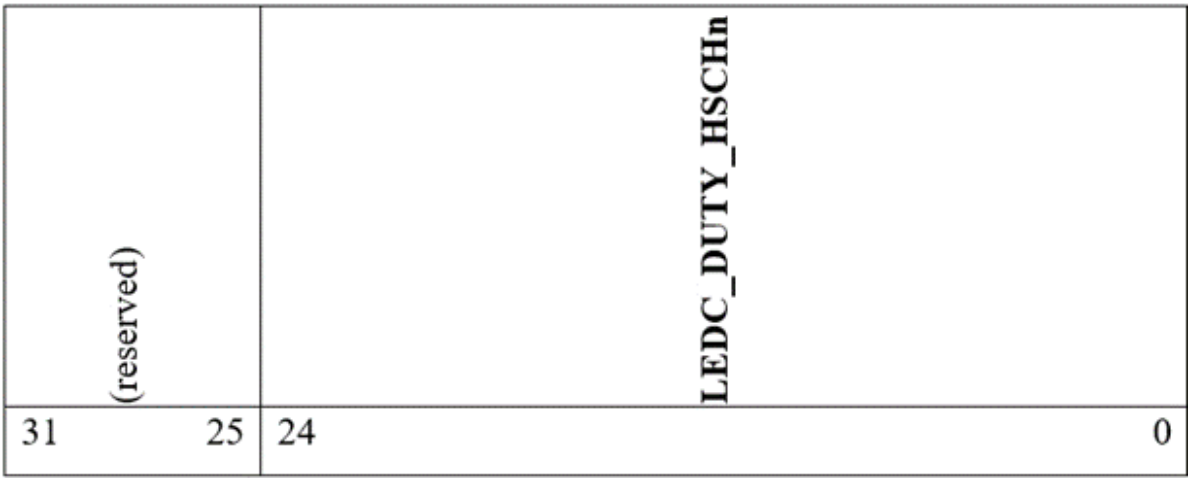


Fig. 8 Structure of the LEDC_HSCHn_DUTY_REG register
Рис. 8 Структура регістру LEDC_HSCHn_DUTY_REG

LEDC_DUTY_HSCHn (bits 24 - 0) defines how long the signal remains high (HIGH) during one PWM period.

When the counter of the hstimerx timer, which is linked to channel n, reaches the value LEDC_LPOINT_HSCHn, the PWM output signal is set to LOW (0).

The duration of the signal's high state is measured in timer ticks. The value of LEDC_LPOINT_HSCHn is calculated by the formula:

$$LEDC\_LPOINT\_HSCHn = LEDC\_HPOINT\_HSCHn1 + LEDC\_DUTY\_HSCHn2 \quad (2.2)$$

or

$$LEDC\_LPOINT\_HSCHn = LEDC\_HPOINT\_HSCHn1 + LEDC\_DUTY\_HSCHn2 + 1 \quad (2.3)$$

depending on the settings. The key point is that the lower 4 bits of the DUTY field are not used

The program implements the enabling of the LEDC clock, the configuration of the HSTIMER0 timer and the HSCH0 channel, as well as outputting the signal to GPIO2:

3 Testing Methodology

3.1 General Methodology

For this research, the ESP32-DevKitC V4 hardware platform, based on the ESP32-WROOM-32D module, was utilized. A Logic Analyzer 24 MHz 8CH served as the measurement tool, allowing for the examination of the timing parameters of pulse signals. Additionally, software control of the ESP32 peripheral clock signal was implemented through the DPORT module, which handles clock enabling/disabling for the LEDC [13].

Timer management was performed via direct access to the microcontroller's registers, using the ESP-IDF libraries:

```
1  #include "soc/ledc_reg.h"
2  #include "soc/io_mux_reg.h"
3  #include "soc/gpio_sig_map.h"  // For LEDC_HS_SIG_OUT0_IDX
4  #include "soc/gpio_reg.h"
5  #include "soc/soc.h"
6  #include "soc/dport_reg.h" // Clock signal control
7  #include "soc/dport_access.h"  //For DPORT_REG_WRITE() and DPORT_REG_READ()
```

Fig. 9 ESP-IDF libraries
Рис. 9 Використані бібліотеки ESP-IDF

The program implements the enabling of the LEDC clock, the configuration of the HSTIMER0 timer and the HSCH0 channel, as well as outputting the signal to GPIO2:

```
11     // Enable LEDC clocking
12     DPORT_REG_WRITE(DPORT_PERIP_CLK_EN_REG, DPORT_REG_READ(DPORT_PERIP_CLK_EN_REG) | DPORT_LEDC_CLK_EN);
13     DPORT_REG_WRITE(DPORT_PERIP_RST_EN_REG, DPORT_REG_READ(DPORT_PERIP_RST_EN_REG) & ~DPORT_LEDC_RST);
14
15     //Connect the LEDC_HS_SIG_OUT0 output signal to GPIO2
16     REG_WRITE(GPIO_FUNC2_OUT_SEL_CFG_REG, LEDC_HS_SIG_OUT0_IDX);//LEDC_HS_SIG_OUT0_IDX = 71 = 0x47
17     REG_WRITE(GPIO_ENABLE_W1TC_REG, (1 << 2));
18
19     //(1<<25) on 80 MHz; (4<<13) integer divider = 4; (6) set PWM to 6 bits
20     REG_WRITE (LEDC_HSTIMER0_CONF_REG, ((1<<25)|(2<<13)|(5)));
21
22     //(1<<2) = 4 sets the ENable enable bit; bits 0 and 1 equal results -> use htimer0
23     REG_WRITE (LEDC_HSCH0_CONF0_REG, (1<<2));
24
25     REG_WRITE (LEDC_HSCH0_HPOINT_REG, 0); // the initial phase is zero
26     REG_WRITE (LEDC_HSCH0_DUTY_REG, (10<<4)); //PWM parameter 10 of 64
27
28     //It is very important to start the timer by writing 0 to the pause bit
29     REG_WRITE(LEDC_HSTIMER0_CONF_REG,REG_READ(LEDC_HSTIMER0_CONF_REG)&~(1<<23));
30     REG_WRITE (LEDC_HSCH0_CONF1_REG, (1 << 31)); //Start updating the duty cycle even though we don't use automatic fade
```

Fig. 10 Main program
Рис. 10 Основна програма

To adjust the configurations, line #19, LEDC_HSTIMER0_CONF_REG, was modified to set the clock source, divider (DIV), and counter resolution (RES). The DUTY value was set in the LEDC_HSCH0_DUTY_REG register on line 25.

During the experiments, the following PWM signal parameters were determined:

- The signal frequency, calculated by formula (2.1), with $f_{CLK} = 80\ MHz$.
- The signal period was calculated by the formula:

$$T_{PWM} = \frac{1}{f_{PWM}} \tag{3.1}$$

- The duration of the high level:

$$t_H = \frac{DUTY}{2^{RES}} * T_{PWM} \tag{3.2}$$

where $DUTY$ is the duty cycle value,
RES is the counter resolution.

3.2 Example of Configuration Analysis

To illustrate the methodology in more detail, let's consider configuration 9 with the following parameters:

RES = 5 bits, **DIV** = 2, **DUTY** = 10

In this configuration, the clock frequency of the LEDC subsystem is $f_{CLK} = 80\ MHz$.. The theoretical frequency is calculated using formula (2.1) and is:

$$f_{PWM} = \frac{f_{CLK}}{DIV \cdot 2^{RES}} = \frac{80 * 10^6 \text{Гц}}{2 * 2^5} = 1.25\ MHz.$$

The signal period is calculated by formula (3.1), and is:

$$T_{PWM} = \frac{1}{f_{PWM}} = \frac{1}{1.25 * 10^6} = 0.8\ \mu s.$$

The duration of the high level is calculated by formula (3.2), and is:

$$t_H = \frac{DUTY}{2^{RES}} * T_{PWM} = \frac{10}{2^5} * 0.8 * 10^{-6} = 0.25\ \mu s$$

Practical results were obtained using a logic analyzer. [14].

Fig. 11 Logic analyzer readings
Рис. 11 Покази логічного аналізатора

The average values of the indicators were calculated over a signal duration of 0.5 seconds and entered into the table.

Table. 1 Configuration 9 Calculations
Табл. 1 Розрахунки конфігурації 9

Configuration 9			
	Frequency, Hz	Period, s	t_H, s
Theoretical	1250000	0.0000008	0.00000025
Practical	1249593.227	0.00000080026	0.0000002495
Error relative to theory	-0.03%	0.03%	-0.18%

The practical results, obtained from the logic analyzer, showed a frequency of 1.2496 MHz, a period of 0.80026 µs, and a high-level duration of 0.2495 µs.

The relative error does not exceed -0.03% for the frequency, 0.03% for the period, and ±0.18% for t_H, which indicates the high accuracy of the implemented register-level control.

4 Results and Analysis

During the experiments, 18 configurations of the LEDC timers were tested with various parameters of resolution (RES = 4, 5, 6 bits), divider (DIV = 1, 2, 4), and duty cycle (DUTY = 10, 33%, 50%). For each configuration, theoretical values of frequency, period, and high-level signal duration were determined, and practical measurements were performed using a logic analyzer. The obtained data demonstrated a high correspondence between the calculations and the experiment, with the maximum error not exceeding ±0.03% for the frequency and period, and ±0.6% for the high-level signal duration.

Analysis of the results showed that as the resolution (RES) increases, the precision of the duty cycle adjustment improves, whereas the PWM frequency is predominantly determined by the value of the divider (DIV).

An increase in DIV inversely proportionally reduces the signal frequency and linearly increases its period. The DUTY parameter directly affects the duration of the high level, t_H, and it was for this indicator that the largest relative deviations were recorded. However, these deviations remain within an acceptable margin of error for practical applications.

5 Conclusions

This work demonstrates the capabilities of direct software control over the registers of the ESP32's LEDC subsystem to generate ultra-high-speed PWM signals. The research confirmed that the signal frequency is inversely proportional to the prescaler value. At the same time, the ratio of the pulse duration

to the period (the duty cycle) remains constant if the DUTY value and the resolution are not changed. Direct register access significantly reduces delays in updating the duty cycle, ensuring more deterministic and stable signal generation, which is crucial for tasks with high requirements for synchronization and error minimization.

The results of this work are of practical significance for embedded systems that require high-speed and precisely regulated PWM signals, particularly in robotics and power electronics, and they also deepen the understanding of the hardware limits of PWM in the ESP32. Further research could be directed towards the application of direct register writes for other LEDC timers and channels, analyzing the impact of different clock sources and low-speed modes on signal quality and power consumption, as well as on the interaction with FreeRTOS cores and ISRs for the synchronous control of multiple channels.

In summary, this work investigated the LEDC architecture, developed an algorithm for the direct configuration of the prescaler, counter resolution, and duty cycle via registers, and conducted a comparative analysis of theoretical and practical calculations which showed a high degree of correspondence. Thus, the scientific problem of controlling LEDC timers has been solved through the development and experimental confirmation of a new approach, which opens up prospects for further optimization and expansion of the ESP32's capabilities in high-speed PWM control.

Funding

This research was supported by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under project No 09I03-03-V01-00119.

REFERENCES

1. Valvano J.W., Yerraballi R. (2014). Embedded Systems — Shape the World: A Cyber-Physical Systems Approach [e-book]. Austin, TX: The University of Texas at Austin. Available at: https://users.ece.utexas.edu/~valvano/Volume1/E-Book/
2. Barr M. (1999). Programming Embedded Systems in C and C++. O'Reilly. 174 p. Available at: https://archive.org/details/programmingembed0000barr
3. Microchip Technology Inc. (2015). ATmega328P — 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash. Datasheet. [Electronic resource]. Available at: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (accessed: 22.09.2025).
4. Microchip Technology Inc. (2014). ATmega640/1280/1281/2560/2561 — 8-bit AVR Microcontroller. Datasheet. [Electronic resource]. Available at: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf (accessed: 22.09.2025).
5. Kotvytskyi, A. T. (2024). Intellectual capital as a basis for innovative development: robotic systems. In Monographic series «European Science» (Book 28, Part 1). Karlsruhe: ScientificWorld-NetAkhatAV. 168 p. ISBN 978-3-98924-041-4. DOI: 10.30890/2709-2313.2024-28-01. Available at: https://desymp.promonograph.org/index.php/sge/issue/view/sge28-01/sge28-01
6. Arduino Store (2025). Arduino UNO Rev3 – Official Product Page. Arduino AG, Italy. [Electronic resource]. Available at: https://store.arduino.cc/collections/uno (accessed: 07.10.2025).
7. Arduino Store (2025). Arduino MEGA 2560 Rev3 – Official Product Page. Arduino AG, Italy. [Electronic resource]. Available at: https://store.arduino.cc/collections/giga (accessed: 07.10.2025).
8. Espressif Systems Official Store (2025). Official Manufacturer Page on AliExpress. Espressif Systems. [Electronic resource]. Available at: https://www.aliexpress.com/store/1100220184 (accessed: 07.10.2025).
9. Arduino (n.d.). Getting Started with Arduino — official documentation (Arduino Docs). [Electronic resource]. Available at: https://docs.arduino.cc/learn/starting-guide/getting-started-arduino/ (accessed: 01.10.2025).
10. All About Circuits (2021). Pulse-width Modulation (PWM) Timers in Microcontrollers. [Electronic resource]. Available at: https://www.allaboutcircuits.com/technical-articles/introduction-to-microcontroller-timers-pwm-timers/ (accessed: 01.10.2025).

11. University of Washington (2010). Lecture 7: ATmega328 Timers and Interrupts (Course CSE P567: Embedded Systems). Seattle: University of Washington. 32 p. Available at: https://courses.cs.washington.edu/courses/csep567/10wi/lectures/Lecture7.pdf
12. Espressif Systems (2020). Technical Reference Manual for ESP32. Version 5.5. 661 p. [Electronic resource]. Available at: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (accessed: 01.10.2025).
13. Espressif Systems (2024). LED Control (LEDC) – Programming Guide for ESP32. ESP-IDF v5.5.1. [Electronic resource]. Available at: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/ledc.html (accessed: 01.10.2025).
14. Saleae Support (2025). Using Logic. [Electronic resource]. Available at: https://support.saleae.com/user-guide/using-logic (accessed: 01.10.2025).

Горенко Даніель Васильович *Магістр ННІ комп'ютерних наук та штучного інтелекту, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022*

Котвицький Альберт Тадеушевич *Кандидат фізико-математичних наук, доцент, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, Україна, 61022*
Pavol Jozef Šafárik University in Košice, 2, Šrobárova, Kosice, 041 80, Slovak Republic

Керування LEDC таймерами мікроконтролера ESP32 за допомогою регістрів

Актуальність. У статті розглядаються питання точного формування та керування широтно-імпульсними (ШІМ) сигналами на базі підсистеми LEDC мікроконтролера ESP32 шляхом прямого доступу до регістрів. Через зростаючі вимоги до точності таймінгу у світлодіодних драйверах, керуванні двигунами та силовій електроніці, а також обмеження високорівневих драйверних інтерфейсів, дослідження є актуальним для розробників вбудованих реального-часу систем.

Мета дослідження. Проаналізувати можливості керування LEDC-таймерами ESP32 через прямий запис у регістри, експериментально оцінити точність та стабільність сформованих PWM-сигналів для ряду конфігурацій і розробити практичні рекомендації щодо оптимізації параметрів.

Методи дослідження. Застосовано регістрове програмування в середовищі ESP-IDF на платі ESP32-DevKitC V4 (WROOM-32D), експериментальні вимірювання часових характеристик вихідних сигналів логічним аналізатором (Logic Analyzer, 24 MHz, 8ch), порівняльний аналіз теоретичних розрахунків (формули частоти, прескейлера, розрядності лічильника) та практичних вимірювань для набора з 18 конфігурацій (RES, DIV, DUTY).

Результати. Розглянуто архітектуру LEDC та структуру відповідних регістрів таймерів і каналів; реалізовано алгоритм налаштування HSTIMER0 та HS-каналу через пряме записування в регістри і виведення сигналу на GPIO. Експериментально підтверджено високу відповідність розрахунків і вимірювань: максимальна відносна похибка частоти і періоду не перевищувала ±0,03%, а тривалості високого рівня — ±0,6%.

Висновки. Прямий регістровий доступ до LEDC дозволяє отримати детерміновані, високоточні ШІМ-сигнали з мінімальною затримкою оновлення параметрів, що є придатними для застосувань у робототехніці, силовій електроніці та інших системах з високими вимогами до синхронізації. Рекомендовано подальші дослідження на тему впливу альтернативних джерел такту, режимів низької швидкості LEDC, інтеграції з ISR/FreeRTOS та розширення підходу на інші таймери й канали.

Ключові слова: *ATmega, ESP32, LEDC, ШІМ, регістрове керування, логічний аналізатор*

Надійшла у першій редакції 02.09.2025, в останній - 05.10.2025.
The first version has been received on 02.09.2025, the final version – on 05.10.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-05

УДК (UDC) 004.051

Зінов'єв Дмитро Володимирович *старший викладач кафедри інтелектуальних програмних систем і технологій, ННІ комп'ютерних наук та штучного інтелекту, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, м. Харків, Україна, 61022*
e-mail: zinoviev@karazin.ua
https://orcid.org/0000-0003-1862-9803

Ткачук Микола Вячеславович *д.т.н., професор; професор кафедри інтелектуальних програмних систем і технологій, ННІ комп'ютерних наук та штучного інтелекту, Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, м. Харків, Україна, 61022*
e-mail: mykola.tkachuk@karazin.ua
https://orcid.org/0000-0003-0852-1081

Архітектура, програмна реалізація та аналіз результатів застосування інтелектуального інструментального засобу для конфігурування мікросервісних застосунків

Актуальність. Розробка застосунків з мікросервісною архітектурою потребує ефективного управління конфігураціями в умовах змінного навантаження, вимог до надійності, відмовостійкості й масштабованості. Це зумовлює потребу в інтелектуальних засобах адаптивного конфігурування, здатних працювати в режимі, близькому до реального часу.
Мета. Створити інтелектуальний інструментальний засіб для адаптивного управління конфігураціями МСА з модулем прийняття рішень на основі Case-Based Reasoning (CBR), спроєктувати його архітектуру, зробити програмну реалізацію, а також експериментально оцінити роботу на тестовому полігоні й порівняти кілька CBR-методів.
Методи дослідження. Уточнено базові поняття процесів конфігурування МСА; спроектовано полігон із трьома сервісами (auth, product, order) і вимогами до продуктивності (≤1000 одночасних запитів, середня затримка ≤200 мс). Адаптивне управління конфігураціями мікросервісів реалізовано як мікросервіс із REST API (FastAPI) та сховищем прецедентів (PostgreSQL); використовуються метрики QoS, ресурсні, «вартісні» та адаптивності. Досліджено п'ять CBR-методів: K-Nearest Neighbors, Weighted KNN, Feature-Based Retrieval, Cluster-Based Retrieval, Indexing & Hashing. Проведено серію вимірювань часу підбору конфігурації для бази прецедентів у 50–1000 записів із усередненням по 100 прогонах.
Результати. Підсистема коректно ідентифікує стани та застосовує релевантні конфігурації для різних сценаріїв (low/medium/high/peak), відповідаючи вимозі часу підбору ≤0,5 с. Найвищу швидкодію продемонстрував метод Indexing & Hashing (≈27,6–50,3 мс для 50–1000 кейсів); KNN має лінійне зростання часу, а Weighted KNN дає керованість за рахунок ваг метрик. Реалізований веб-інтерфейс забезпечує моніторинг і ручний/автоматичний режим застосування змін у реальному часі.
Висновки. Запропонована архітектура та програмна реалізація інструментального засобу з CBR підтверджують практичну доцільність адаптивного конфігурування МСА й створюють підґрунтя для масштабованих даними керованих рішень. Окреслено подальші напрями: еволюція кейс-бази з онлайн навчанням, багатокритеріальна оптимізація (продуктивність/надійність/вартість/енергоефективність), глибша інтеграція з оркестраторами та service mesh, підвищення пояснюваності рішень.

Ключові слова: *програмний мікросервіс, архітектура, управління конфігураціями, інтелектуальний підхід, метод аналізу прецедентів, CBR, інтелектуальний інструментальний засіб, тестування, якість, метрика, модель.*

Як цитувати: Зінов'єв Д.В., Ткачук М.В. Архітектура, програмна реалізація та аналіз результатів застосування інтелектуального інструментального засобу для конфігурування мікросервісних застосунків. *Вісник Харківського національного університету імені В.Н.Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.56-65. https://doi.org/10.26565/2304-6201-2025-67-05

How to quote: D.V. Zinov'ev, M.V. Tkachuk, "Architecture, software implementation and results analyzing of the usage an intelligent tool for configuring microservice applications". *Bulletin of V.N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems,* 2025. vol. 67, pp.56-65. https://doi.org/10.26565/2304-6201-2025-67-05 [in Ukrainian]

Вступ. Актуальність і мета дослідження.

В сучасних умовах стрімкого розвитку інформаційних технологій мікросервісна архітектура (МСА) стала провідною парадигмою проектування розподілених програмних систем. Цей підхід

передбачає розбиття великої та складної системи на набір невеликих, автономних сервісів, кожен із яких виконує свою чітко визначену функцію та взаємодіє з іншими через стандартизовані інтерфейси, такі як REST або gRPC. Це дозволяє створювати масштабовані, гнучкі та легко підтримувані застосунки, що складаються з незалежних сервісів, кожен із яких виконує окрему функцію. Слабка зв'язність між компонентами в МСА забезпечує незалежне розгортання, масштабування та оновлення кожного мікросервісу, що надає суттєві переваги по відношенню до монолітних архітектур [1]. Конфігурування мікросервісів являє собою складний процес, що охоплює налаштування численних параметрів (мережеві адреси, ліміти ресурсів, ключі доступу, налаштування безпеки, тощо) для кожного сервісу. У реальних системах ці параметри можуть часто змінюватися в залежності від навантаження, типу запитів, відмов компонентів або зовнішніх умов. Некоректне конфігурування окремих сервісів може призвести до критичних збоїв у роботі всієї системи - від затримок у відповіді на деякі запити до повної недоступності її функціоналу. Проблеми ще більше ускладнюються, коли система працює в хмарному середовищі з автоматичним масштабуванням, де умови змінюються в режимі реального часу. У такому контексті конфігурації не можуть бути статичними, а мають постійно адаптуватися до змін обчислювального навантаження, доступності сервісів і зовнішніх залежностей [2].

Більшість сучасних рішень у цій сфері, такі як Kubernetes ConfigMaps, HashiCorp Consul або Spring Cloud Config [3], пропонують статичні або шаблонізовані підходи, які не враховують поточний стан системи та вимагають ручного втручання адміністратора. Таким чином, актуальним є науково-технічне завдання розробки інтелектуальних засобів адаптивного управління конфігураціями мікросервісних застосунків. Ця задача стикається з високою складністю технологічних процесів, великою кількістю параметрів та слабкою формалізацією їх взаємозв'язків. Для вирішення цих проблем пропонується використовувати підходи штучного інтелекту, зокрема метод аналізу прецедентів (Case-Based Reasoning - CBR), і розроблена алгоритмічна модель адаптивного управління конфігураціями МСА з його використанням [4].

Мета роботи

Метою цієї роботи є розробка архітектури, програмна реалізація та дослідження результатів застосування інтелектуального інструментального засобу для автоматизованого конфігурування застосунків з МСА на основі методу аналізу прецедентів, що забезпечує пошук та адаптацію конфігурацій окремих мікросервісів з урахуванням змін у середовищі їх функціонування.

2. Архітектура та особливості програмної реалізації інструментального засобу адаптивного управління конфігураціями мікросервісів (АУКМ)

Для організації процесу АУКМ в [3] була запропонована структурно-функціональна схема інструментального засобу, яка містить наступні шари компонентів:

- основний функціонал системи, що передбачає розгортання кожного окремого мікросервісу у відповідному Docker-контейнері, множина яких оркеструється засобами технології Kubernetes;
- модуль прийняття рішень, де спеціальний керуючий мікросервіс, що реалізує методи CBR, приймає метрики через API, обирає адаптивну конфігурацію та формує відповідь;
- компоненти для моніторингу стану МСА, які забезпечують збір даних та розрахунок ключових показників (метрик): завантаження CPU, розмір оперативної пам'яті, час затримки відповідей та ін.) після застосування змін, що забезпечує замкнений цикл адаптації;
- базу даних (БД) прецедентів, кожний з яких включає значення параметрів конфігурації та метрики продуктивності МСА при застосуванні певного алгоритму CBR.

2.1. Функціональні та нефункціональні вимоги до розробки засобу АУКМ

Для визначення вимог до засобу АУКМ був розроблений програмний тестовий полігон, що складався з трьох мікросервісів, які є типовими для застосунку з МСА у предметній області «*Обробка замовлень користувачів при роботі з каталогом продуктів*», а саме: *auth_service* для автентифікації користувачів; *product_service* для керування каталогом продуктів та *order_service* для обробки замовлень. Його функціонал включав:

– генерацію навантаження різного рівня (низьке, середнє, високе) з можливістю ручного налаштування параметрів, таких як розмір пулу підключень до БД та обмеження на кількість запитів;
– API-взаємодію з мікросервісами для отримання метрик (CPU, пам'ять, затримка відповіді, доступність) та застосування змін конфігурації у реальному часі.

Підсистема АУКМ мала виконувати наступні задачі (тобто, функціональні вимоги):
– автоматичний підбір потрібних конфігурацій МСА на основі отриманих метрик за допомогою методів CBR (K-Nearest Neighbors, Cluster-Based Retrieval тощо);
– повний цикл обробки запитів до БД прецедентів: пошук релевантних рішень, адаптація до поточного стану системи, збереження нових випадків для навчання;
– інтеграцію з тестовим полігоном через API для внесення змін без зупинки сервісів**;**
– інтерфейс адміністратора для налаштування параметрів алгоритмів для окремих методів CBR та перегляду результатів знайдених рішень.

Також шляхом вивчення деяких існуючих рішень [5-8], а також експертним шляхом були визначені наступні нефункціональні вимоги до тестового полігону та системи АУКМ, які представлені у Таблиці 1.

Таблиця 1. Узагальнені функціональні та нефункціональні вимоги
Table 1 Generalized functional and non-functional requirements

Вимога	**Тестовий полігон**	**Інструментарій АУКМ**
Функціональні	Симуляція навантаження, збір метрик, API для конфігурування	Автоматичний підбір конфігурацій через CBR, інтеграція з API, збереження прецедентів
Продуктивність	≤200 мс затримки, 1000 запитів	≤0,5 с на підбір конфігурації
Масштабованість	До 10 мікросервісів	До 2000 прецедентів
Надійність	≥95% доступності	Стійкість до збоїв одного мікросервісу
Безпека	OAuth 2.0, AES-256	OAuth 2.0, AES-256
Інтеграція/Розширюваність	RESTful API, модульна архітектура	RESTful API, підтримка нових методів CBR

2.2. Особливості реалізації тестового полігону для дослідження засобу АУКМ

Для ізоляції даних при роботі мікросервісів тестового полігону передбачені окремі таблиці в базі даних PostgreSQL, взаємодія між сервісами здійснюється за допомогою архітектурного стилю RESTful API, і на рисунку 1 наведена ER-діаграма бази даних тестового полігону.

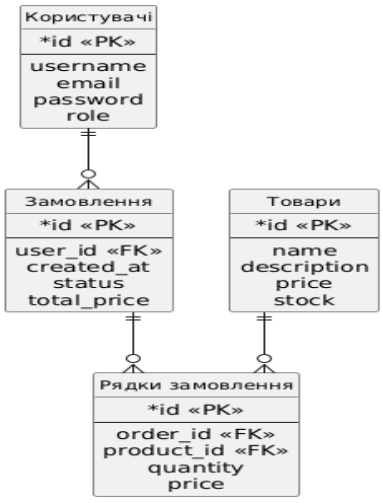


Рис. 1. ER-діаграма бази даних тестового полігону
Fig. 1 ER diagram of the testing ground database

Кожний мікросервіс у складі полігону має набір своїх конфігураційних параметрів, які система АУКМ повинна адаптивно змінювати на основі метрик системи, таких як завантаження процесора чи затримка відповіді мікросервіса. Ці параметри дозволяють адаптувати сервіси до різних сценаріїв навантаження: низьке (до 100 запитів/с), середнє (100-500 запитів/с), високе (500-1000 запитів/с) і пікове (понад 1000 запитів/с). Зміна параметрів відбувається через захищені ендпоінти POST /config/update із подальшим перезапуском бази даних у разі зміни db_connection_type або db_pool_size.

2.3. Розробка підсистеми адаптивного управління конфігураціями мікросервісів

На рисунку 2 наведена діаграма компонентів системи АУКМ.

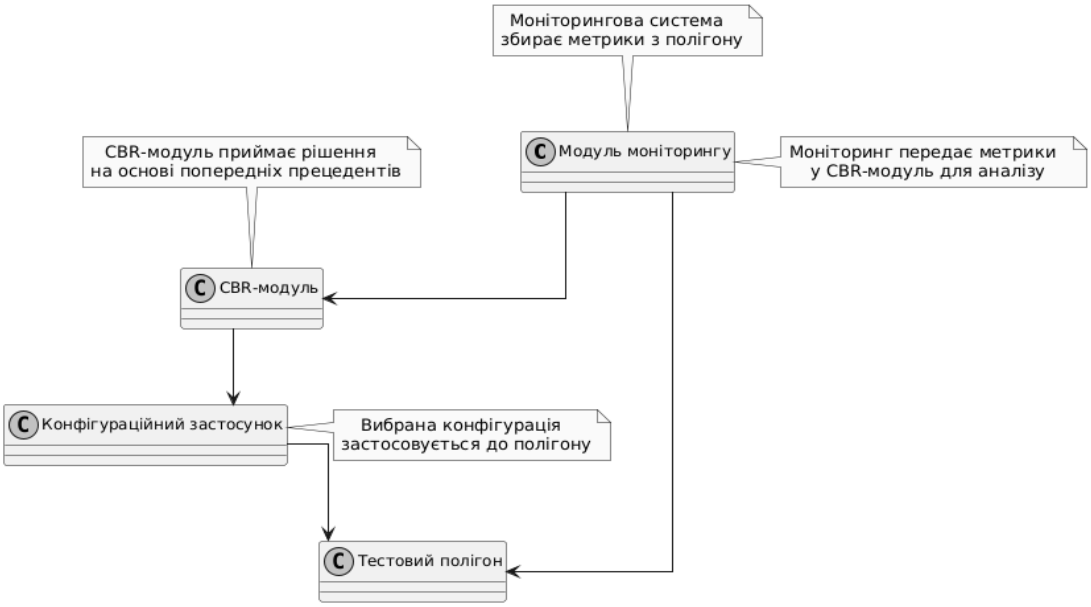


Рис. 2. Діаграма компонентів системи АУКМ
Fig. 2 Diagram of the components of the AMCM system

Система АУКМ визначає потрібні конфігурації МСА тестового полігону за поточними метриками, використовуючи відповідні методи CBR, і має наступні компоненти:

- *модуль моніторингу* збирає наступні метрики з тестового полігону: середня затримка (performance_metric); час відповіді (response_time); використання пам'яті (memory_usage); використання дискового простору (disk_usage); завантаження процесора (cpu_load), мережевий трафік (network_usage), операційні витрати (operational_costs) і доступність (availability). через ендпоінти /health мікросервісів по HTTP.
- *CBR-модуль* зіставляє поточний стан із базою прецедентів і обирає конфігурацію на основі одного з наявних CBR методів: KNN (звичайний та з вагами), Feature-Based, Cluster-Based або Indexing & Hashing.
- *Модуль конфігурування* замінює конфігурації мікросервісів auth_service, product_service, order_service на наново обрані через ендпоінти POST /config/update; також, цей модуль передбачає збереження нових прецедентів у базі даних.

3. Методика та аналіз результатів застосування запропонованого інструментального засобу для конфігурування мікросервісів

3.1. Методика проведення експериментів на тестовому програмному полігоні АУКМ

Стан кожного з мікросервісів тестового полігону описувався вектором метрик: завантаження процесора (CPU), використання оперативної пам'яті (RAM) і дискової пам'яті (DISK), мережева активність, середня затримка/час відповіді s доступність відповідного сервісу. Для відтворення сценаріїв low/medium/high/peak/suboptimal використовувався скрипт генерації значень параметрів конфігурацій МСА у заданих діапазонах. У процедурі порівняння для кожного CBR-методу час обчислення вимірювався 100 разів і потім осереднювався. Експерименти проводилися з різними

розмірами бази прецедентів - 50, 100, 200, 500 і 1000 записів. В якості тестового середовища використовувався Windows 10 22H2; RAM 16 GB DDR4 3200 MHz; CPU Ryzen 5 4600H (6C/12T). Додатково перевірялися результати коригування конфігурацій через інтерфейс системи і їх фактичне застосування до сервісів (див. нижче).

3.2. Інтерфейс користувача підсистеми АУКМ

Підсистема АУКМ реалізована як односторінковий веб-застосунок (single-page application) із двома вкладками. Перша вкладка “Monitoring & Manual Configuration” відображає поточні показники параметрів конфігурацій МСА за різними метриками, які згруповані у такі категорії:

- QoS Metrics - середня затримка, час відповіді, доступність сервісів;
- System Resources - завантаження CPU/RAM/диску та мережі;
- Cost Efficiency Metrics - умовні операційні витрати, що обчислюються як зважена функція ресурсних показників з обраними ваговими коефіцієнтами, які визначалися експертним шляхом: (OC = 0.4·CPU + 0.3·RAM + 0.2·DISK + 0.1·NETWORK);
- Adaptability Metrics - час адаптації та кількість виконаних реконфігурацій.

На рисунку 3 наведений приклад метрик категорії «System Resources», а на рисунку 4 показані поточні конфігурації “Current Configurations” для всіх 3-х мікросервісів тестового полігону.

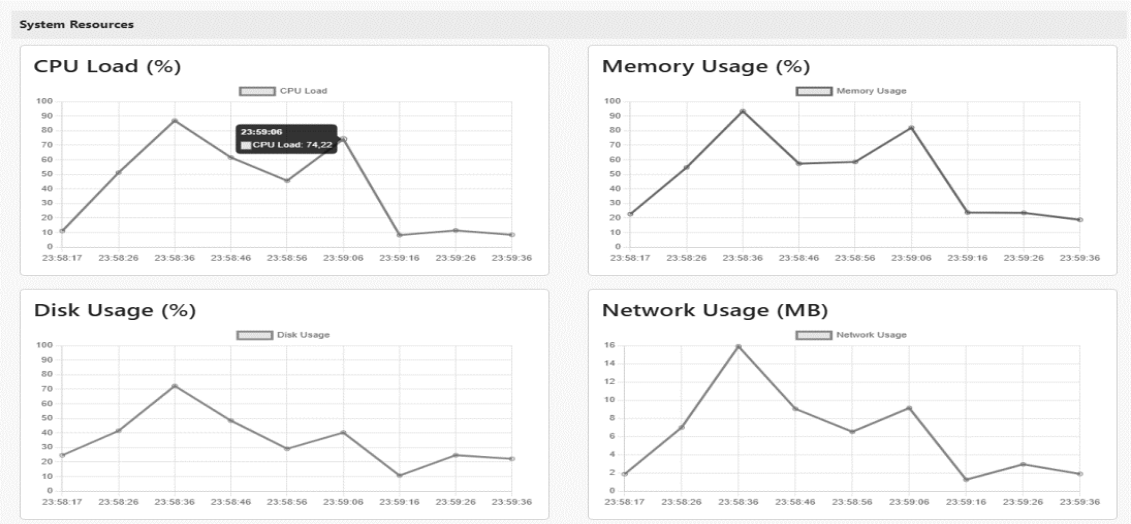


Рис. 3. Категорія «System Resources»
Fig. 3 “System Resources” category

Current Configurations

Service	Config
auth_service	{ "access_token_expire_minutes": 30, "algorithm": "HS256", "db_connection_type": "sync", "db_pool_size": 5, "rate_limit_requests": 100, "response_delay": 0 }
product_service	{ "db_connection_type": "sync", "db_pool_size": 5, "rate_limit_requests": 100, "response_delay": 0, "max_products_per_page": 20 }
order_service	{ "db_connection_type": "sync", "db_pool_size": 5, "rate_limit_requests": 100, "response_delay": 0, "max_orders_per_page": 20, "default_order_status": "pending" }

Рис. 4. Поточні конфігурації мікросервісів тестового полігону
Fig. 4 Current configurations of testing ground microservices

Друга вкладка “Automatic Configuration” (див. рисунки 5 та 6) надає можливість обрати відповідний CBR-метод, увімкнути його авто-конфігурацію параметрів та налаштувати період їх оновлення. Наприклад, у методі Weighted KNN задаються ваги метрик; у методі Feature-Based - набір ознак для поточного прецеденту; у методі Cluster-Based - кількість кластерів) і т.п.

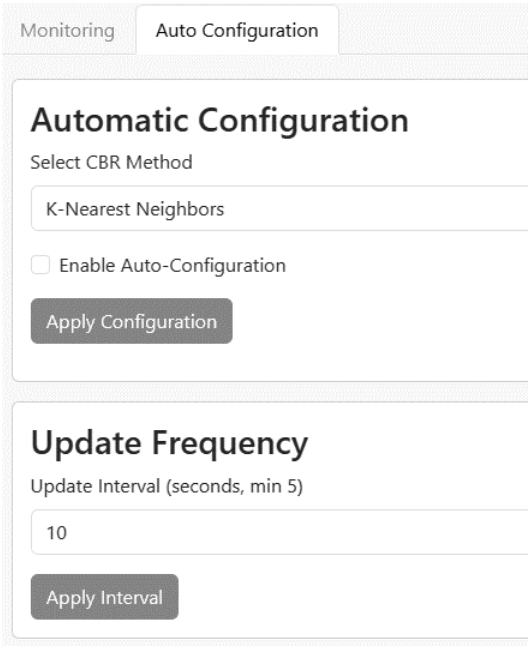


Рис.5. Вкладка «Automatic Configuration»
Fig. 5 Automatic Configuration tab

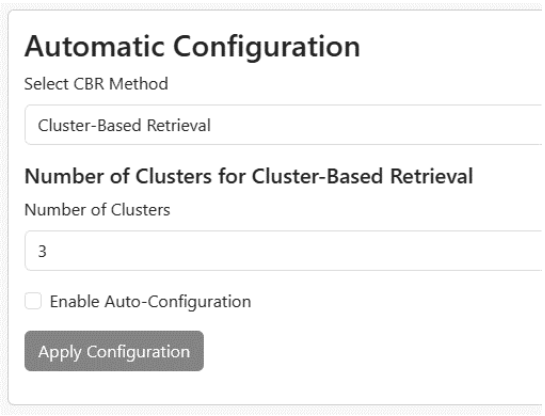


Рис. 6. Параметри для методу Cluster-Based Retrieval

3.3. Результати проведення програмних експериментів та їх аналіз

Для проведення досліджень був розроблений скрипт на мові Python, який генерував значення для параметрів конфігурацій МСА у діапазонах, які відповідали різним типам обчислювального навантаження полігону (вони представлені у Таблиці 2).

Таблиця 2. Інтервали значень конфігураційних параметрів для різних діапазонів навантаження
Table 2 Configuration parameter value ranges for different load ranges

Параметр конфігурації мікросервіса	**Навантаження**				
	Low Load	**Medium Load**	**High Load**	**Peak Load**	**Suboptimal**
cpu_load	5.0-20.0	30.0-50.0	60.0-80.0	80.0-95.0	40.0-70.0
memory_usage	15.0-30.0	40.0-60.0	65.0-85.0	85.0-95.0	50.0-80.0
disk_usage	10.0-25.0	25.0-40.0	40.0-60.0	60.0-75.0	30.0-50.0
network_usage	0.5-3.0 MB/s	3.0-8.0 MB/s	8.0-15.0 MB/s	15.0-25.0 MB/s	5.0-10.0 MB/s
performance_metric	10.0-30.0 ms	30.0-80.0 ms	80.0-150.0 ms	150.0-300.0 ms	200.0-400.0 ms
response_time	5.0-20.0 ms	20.0-50.0 ms	50.0-100.0 ms	100.0-200.0 ms	150.0-300.0 ms
operational_costs	5.0-15.0 units	15.0-30.0 units	30.0-50.0 units	50.0-80.0 units	25.0-60.0 units
availability	95.0-100.0%	90.0-100.0%	85.0-95.0%	80.0-90.0%	70.0-85.0%
adaptation_time	0.1-0.5 s	0.5-1.0 s	1.0-2.0 s	2.0-5.0 s	1.5-3.0 s
reconfigurations	0-2	1-5	3-8	5-10	2-6

Спочатку система АУКМ генерувала поточні значення параметрів мікросервісів тестового полігону, а потім за допомогою одного з CBR методів знаходила в БД прецедент для поточних параметрів системи. На рисунку 7 показаний результат вибору із БД прецедентів нового сценарію на основі поточних параметрів системи.

```
2025-06-05 09:21:54,363 - INFO - Best case selected: performance_metric=50.36
2025-06-05 09:21:54,363 - INFO - Retrieved config: {
  "auth_service": {
    "access_token_expire_minutes": 60,
    "algorithm": "HS256",
    "db_connection_type": "sync",
    "db_pool_size": 7,
    "rate_limit_requests": 150,
    "response_delay": 0.1
  },
  "product_service": {
    "db_connection_type": "sync",
    "db_pool_size": 7,
    "rate_limit_requests": 150,
    "response_delay": 0.1,
    "max_products_per_page": 30
  },
  "order_service": {
    "db_connection_type": "sync",
    "db_pool_size": 5,
    "rate_limit_requests": 100,
    "response_delay": 0.0,
    "max_orders_per_page": 20,
    "default_order_status": "pending"
  }
```


Рис. 7. Результат вибору потрібного прецеденту
Fig. 7 Result of selecting the necessary precedent

На рисунку 8 показаний результат застосування конфігурацій нового сценарію до кожного з мікросервісів тестового полігону.


```
2025-06-05 09:21:54,375 - INFO - Updated configuration for auth_service: {
  "access_token_expire_minutes": 60,
  "algorithm": "HS256",
  "db_connection_type": "sync",
  "db_pool_size": 7,
  "rate_limit_requests": 150,
  "response_delay": 0.1
}
2025-06-05 09:21:54,503 - INFO - Updated configuration for product_service: {
  "db_connection_type": "sync",
  "db_pool_size": 7,
  "rate_limit_requests": 150,
  "response_delay": 0.1,
  "max_products_per_page": 30
}
2025-06-05 09:21:54,618 - INFO - Updated configuration for order_service: {
  "db_connection_type": "sync",
  "db_pool_size": 5,
  "rate_limit_requests": 100,
  "response_delay": 0.0,
  "max_orders_per_page": 20,
  "default_order_status": "pending"
}
```


Рис. 8. Результат застосування конфігурацій нового сценарію до кожного з мікросервісів
Fig. 8 The result of applying the new scenario configurations to each of the microservices

У всіх експериментах з поточними конфігураціями засіб АУКМ коректно ідентифікував поточний стан мікросервісів та застосовував відповідні параметри конфігурації, що підтверджує його працездатність. Важливим чинником для застосування цього підходу на практиці є час, який потрібен для пошуку такого рішення, і у Таблиці 3 та на рисунку 9 наведеніані результати досліджень з порівняння часу виконання алгоритмів п'яти CBR методів в залежності від об'єму бази прецедентів.

Таблиця 3. Порівняння часу виконання CBR-методів в залежності від об'єму бази прецедентів
Table 3. Comparison of execution time of CBR methods depending on the volume of the precedent databases

Метод	**Час виконання алгоритмів (мс)**				
	для 50 записів	**для 100 записів**	**для 200 записів**	**для 500 записів**	**для 1000 записів**
K-Nearest Neighbors	44,8	58,3	68,9	105,2	152,4
Weighted K-Nearest Neighbors	49,6	64,1	75,8	112,7	165,9
Feature-Based Retrieval	47,3	61,9	72,5	109,8	159,2
Cluster-Based Retrieval	53,9	69,2	82,4	126,5	190,8
Indexing and Hashing	27,6	32,1	36,9	43,2	50,3

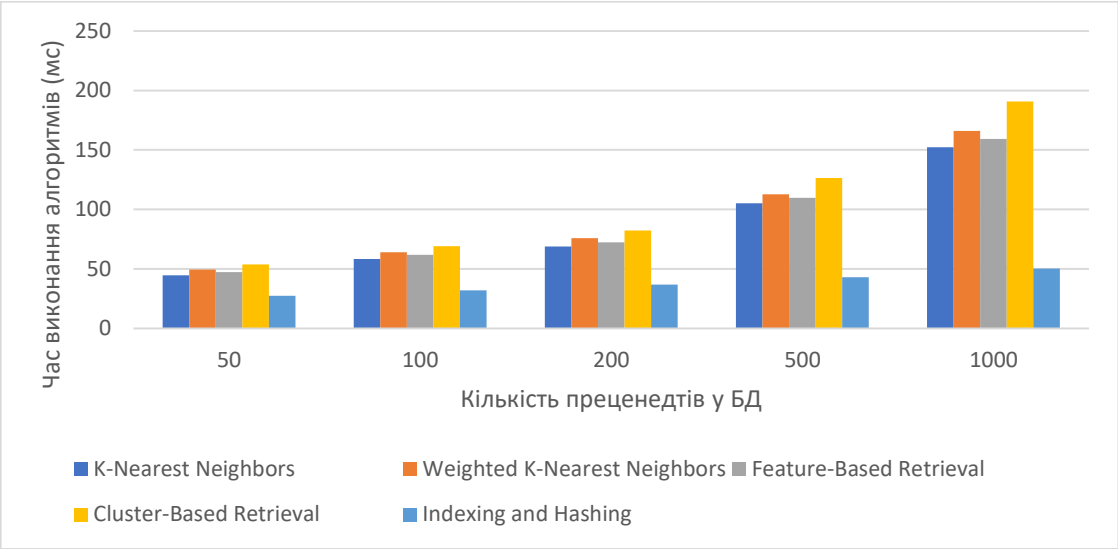


Рис. 9. Стовпчаста діаграма порівняння швидкодії всіх CBR-методів
Fig. 9 Bar chart comparing the speed of all CBR methods

Аналізуючи результати дослідження можна зробити висновки, що за критерієм швидкодії оновлення конфігурацій найкращим виявився CBR метод Indexing and Hashing завдяки попередній індексації БД прецедентів, що суттєво скорочує час пошуку. Метод KNN демонструє лінійне зростання часу з обсягом бази прецедентів і забезпечує стабільність при малих даних, а метод Weighted KNN надає гнучкість за рахунок вагових коефіцієнтів, але працює повільніше за інших. Методи Feature-Based та Cluster-Based є компромісом між точністю, масштабованістю та обчислювальними витратами (див. таблицю 3).

4. Висновки та напрямки подальших досліджень

У цьому дослідженні розв'язано актуальну науково-технічну задачу автоматизації адаптивного конфігурування застосунків з МСА шляхом створення інтелектуального інструментального засобу з модулем прийняття рішень на основі методів CBR. Реалізовано програмний тестовий полігон та розроблено інструментальний засіб АУКМ із чітко визначеними вимогами до

продуктивності, масштабованості, безпеки та інтеграції. Побудовано ER-модель БД прецедентів і реалізовано веб-інтерфейс для моніторингу/ручного та автоматичного конфігурування МСА.

Також в роботі оцінено часові витрати п’яти різних CBR-методів у діапазоні розмірів БД прецедентів у діапазоні [50-1000] записів. Найвищу швидкодію продемонстрував Indexing and Hashing (орієнтовно 27,6–50,3 мс), KNN показав лінійне зростання часу з обсягом кейс-бази, а Weighted KNN забезпечив більшу керованість за рахунок вагових коефіцієнтів для окремих параметрів опису прецедентів.

СПИСОК ЛІТЕРАТУРИ

1. Su R., Li X., Taibi D. From Microservice to Monolith: A Multivocal Literature Review. Electronics. 2024. Vol. 13, No. 8. Art. 1452. DOI: 10.3390/electronics13081452.
URL: https://www.mdpi.com/2079-9292/13/8/1452
2. Pozdniakova O.; Mažeika D.; Cholomskis A. SLA-Adaptive Threshold Adjustment for a Kubernetes Horizontal Pod Autoscaler. *Electronics*. 2024. Т. 13, № 7. 1242. DOI: 10.3390/electronics13071242.
URL: https://www.mdpi.com/2079-9292/13/7/1242 // [2]
3. Зінов’єв Д.В., Ткачук М.В. Аналіз, класифікація та тестування інструментальних засобів для управління конфігураціями програмних мікросервісів. Вісник Харківського національного університету імені В.Н.Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». 2023. вип. 57. С.33-42, DOI: 10.26565/2304-6201-2023-57-03
URL: https://periodicals.karazin.ua/mia/article/view/23251 // [3]
4. Ткачук М. В., Зінов’єв Д.В. Розробка та дослідження алгоритмічної моделі для адаптивного управління конфігураціями програмних мікросервісів. Системи обробки інформації. 2024. № 2(177). - С. 116 –120, DOI:10.30748/soi.2024.177.13
URL: https://doi.org/10.30748/soi.2024.177.12 // [4]
5. Figueira J.; Coutinho C. Developing Self-Adaptive Microservices. *Procedia Computer Science*. 2024. Т. 232. С. 264–273. DOI: 10.1016/j.procs.2024.01.026. // [5]
URL: https://www.sciencedirect.com/science/article/pii/S1877050924000267
6. Ma W.; Wang R.; Gu Y.; Meng Q.; Huang H.; Deng S.; Wu Y. Multi-objective microservice deployment optimization via a knowledge-driven evolutionary algorithm. *Complex & Intelligent Systems*. 2021. Т. 7. С. 1153–1171. DOI: 10.1007/s40747-020-00180-1. // [6]
URL: https://link.springer.com/article/10.1007/s40747-020-00180-1
7. Niswar M.; Safruddin R. A.; Bustamin A.; Aswad I. Performance Evaluation of Microservices Communication with REST, GraphQL, and gRPC. *International Journal of Electronics and Telecommunications*. 2024. Т. 70, № 2. С. 429–436. DOI: 10.24425/ijet.2024.149562.
URL: https://ijet.ise.pw.edu.pl/index.php/ijet/article/view/10.24425-ijet.2024.149562 /
8. Yan A.; Cheng Z. A Review of the Development and Future Challenges of Case-Based Reasoning. *Applied Sciences*. 2024. Т. 14, № 16. 7130. DOI: 10.3390/app14167130.
URL: https://www.mdpi.com/2076-3417/14/16/7130

REFERENCES

1. R. Su, X. Li, and D. Taibi, “From Microservice to Monolith: A Multivocal Literature Review,” *Electronics*, vol. 13, no. 8, p. 1452, Apr. 2024, doi: 10.3390/electronics13081452.
Available: https://www.mdpi.com/2079-9292/13/8/1452
2. O. Pozdniakova, D. Mažeika, and A. Cholomskis, “SLA-Adaptive Threshold Adjustment for a Kubernetes Horizontal Pod Autoscaler,” *Electronics*, vol. 13, no. 7, 1242, 2024, doi: 10.3390/electronics13071242.
Available: https://www.mdpi.com/2079-9292/13/7/1242
3. Zinov’ev, D.V. and Tkachuk, M.V. (2025), “Rozrobka ta doslidzhenniy algoritmichnoi modeli gla adaptivnogo upravlinnya konfiguratsiyami programnuh mikroservisiv” [Development and research of an algorithmic model for adaptive configuration management of software microservices], *Information processing systems*. 2024. № 2(177). - P. 116 –120. [in Ukrainian]
Available: https://doi.org/10.30748/soi.2024.177.13
4. Zinov’ev, D., & Tkachuk, M. (2023). “Analiz, klasyfikatsiia ta testuvannia instrumentiv upravlinnia konfiguratsiiemi dlia programnykh mikroservisiv” [Analysis, classification and testing of

configuration management tools for software microservices] *Bulletin of V.N. Karazin Kharkiv National University, Series «Mathematical Modeling. Information Technology. Automated Control Systems»*, *57*, 32-41, doi: 10.26565/2304-6201-2023-57-03
Available: https://periodicals.karazin.ua/mia/article/view/23251
5. J. Figueira and C. Coutinho, "Developing Self-Adaptive Microservices," *Procedia Computer Science*, vol. 232, pp. 264–273, 2024, doi: 10.1016/j.procs.2024.01.026.
Available: https://www.sciencedirect.com/science/article/pii/S1877050924000267
6. W. Ma, R. Wang, Y. Gu, Q. Meng, H. Huang, S. Deng, and Y. Wu, "Multi-objective microservice deployment optimization via a knowledge-driven evolutionary algorithm," *Complex & Intelligent Systems*, vol. 7, pp. 1153–1171, 2021, doi: 10.1007/s40747-020-00180-1.
Available: https://link.springer.com/article/10.1007/s40747-020-00180-1
7. M. Niswar, R. A. Safruddin, A. Bustamin, and I. Aswad, "Performance Evaluation of Microservices Communication with REST, GraphQL, and gRPC," *International Journal of Electronics and Telecommunications*, vol. 70, no. 2, pp. 429–436, Jun. 2024, doi: 10.24425/ijet.2024.149562.
Available: https://ijet.ise.pw.edu.pl/index.php/ijet/article/view/10.24425-ijet.2024.149562
8. Yan and Z. Cheng, "A Review of the Development and Future Challenges of Case-Based Reasoning," Applied Sciences, vol. 14, no. 16, art. 7130, 2024, doi: 10.3390/app14167130.
Available: https://www.mdpi.com/2076-3417/14/16/7130

Zinov'ev Dmytro *Senior lecturer of the Department of Intelligent Software Systems and Technologies, Education and Research Institute of Computer Sciences and Artificial Intelligence, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine*

Tkachuk Mykola *Doctor of technical sciences, Professor; Professor of the Department of Intelligent Software Systems and Technologies, Education and Research Institute of Computer Sciences and Artificial Intelligence, V. N. Karazin Kharkiv National University, 4 Svobody Sq., Kharkiv, 61022, Ukraine*

Architecture, software implementation and results analyzing of the usage an intelligent tool for configuring microservice applications

Actuality. Developing applications with a microservice architecture requires effective configuration management under varying load conditions, reliability, fault tolerance, and scalability requirements. This creates a need for intelligent adaptive configuration tools that can operate in near-real time mode.
Goal. To create an intelligent tool for adaptive management of MCA configurations with a decision-making module based on Case-Based Reasoning (CBR), design its architecture, make a software implementation, as well as experimentally evaluate the work on a test site and compare several CBR methods.
Research methods. The basic concepts of MSA configuration processes are clarified; a polygon with three services (auth, product, order) and performance requirements (≤1000 simultaneous requests, average latency ≤200 ms) is designed. Adaptive microservice configuration management is implemented as a microservice with REST API (FastAPI) and a precedent database (PostgreSQL); QoS, resource, "cost" and adaptability metrics are used. Five CBR methods are investigated: K-Nearest Neighbors, Weighted KNN, Feature-Based Retrieval, Cluster-Based Retrieval, Indexing & Hashing. A series of measurements of configuration selection time for a precedent database of 50–1000 records with averaging over 100 runs is conducted.
Results. The subsystem correctly identifies states and applies relevant configurations for different scenarios (low/medium/high/peak), meeting the requirement of a matching time of ≤0.5 s. The Indexing & Hashing method demonstrated the highest performance (≈27.6–50.3 ms for 50–1000 precedents); KNN has a linear time growth, and Weighted KNN provides controllability due to metric weights. The implemented web interface provides monitoring and manual/automatic mode of applying changes in real time.
Conclusions. The proposed architecture and software implementation of the CBR tool confirm the practical feasibility of adaptive configuration of the MCA and create a basis for managed solutions that are scaled by data. Further directions are outlined: evolution of the case base with online learning, multi-criteria optimization (performance/reliability/cost/energy efficiency), deeper integration with orchestrators and service mesh and increased explainability of solutions.

Keywords: *microservice, architecture, configuration management, intelligent approach, Case Based Reasoning, CBR, intelligent tool, testing, quality, metrics, model.*

Надійшла у першій редакції 30.08.2025, в останній - 01.10.2025.
The first version has been received on 30.08.2025, the final version – on 01.10.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-06

УДК (UDC) 519.6

Kotenko Dmytro *PhD student,*
Anatolii Pidhornyi Institute of Power Machines and Systems of the NAS of Ukraine, Ukraine, Kharkiv, 2/10 Kommunalnykiv str., 61023
e-mail: dima.kotenko.96@gmail.com
https://orcid.org/0009-0006-7503-3837

Zipunnikov Mykola *Ph.D., senior research associate, department of power machines,*
Anatolii Pidhornyi Institute of Power Machines and Systems of the NAS of Ukraine, Ukraine, Kharkiv, 2/10 Kommunalnykiv str., 61023
e-mail: zipunnikov_n@ukr.net;
https://orcid.org/0000-0002-0579-2962

Application of a genetic algorithm to solve the problem of scaling hydrogen systems

The work aims to develop a robust tool for scaling hydrogen systems and their energy consumption using a genetic algorithm.
Relevance. The most common method of hydrogen production is water electrolysis, which requires a sufficient amount of electricity. If electricity sources are insufficient, this can put additional strain on the power grid, especially during peak consumption periods. Since 87% of hydrogen plants currently use hydrogen on-site (instead of generating it and then transporting it for use), there is a need for optimization in this area to improve energy efficiency and sustainability.
Current research analyzes the improvement of hydrogen systems in terms of the cost-effectiveness of systems using renewable energy sources and the reduction of hydrogen logistics costs by applying linear programming and particle swarm optimization methods.
However, these works are mainly focused on hydrogen production systems based on a single electrolyzer and do not aim to assess the feasibility of using multiple units. As a result, the topic of cost optimization and maintenance strategies for multi-electrolyzer systems remains less explored, as well as the related problem of their dispatching.
Research methods. Stochastic methods were used to solve the problem of finding the best startup queue for electrolysis units, and the effectiveness of the genetic algorithm for solving this problem was tested.
Results. A model for optimizing the peak power consumption of an electrolysis system was built, and the configuration evaluation function and objective function for system optimization were determined. The choice of a stochastic optimization method is justified by checking the objective function for the properties necessary for the effectiveness of traditional optimization methods, namely, continuity, differentiability, smoothness, and convexity. The effectiveness of the genetic method was tested in comparison with the gradient descent method on examples with different configurations of electrolyzers (similar and different types).
Conclusions. These calculations have confirmed that the genetic algorithm has stable results and is effective in finding the global optimum, while the gradient descent may stop at local minima and require additional adjustments to achieve the optimal solution. Using the genetic algorithm method, we obtain results that give an approximate optimal result for a fixed number of steps. This approximate result, as shown in the problem with the placement of 10 electrolyzers, gives significant results — the peak electricity consumption has decreased by almost 40%.
Further research can be aimed at improving the parameters of the algorithm, in particular, adaptive tuning of the mutation and crossover operators to increase the convergence rate.
keywords: *Optimization, Stochastic (non-deterministic) methods, Genetic Algorithm, power consumption, hydrogen systems, electrolysis unit.*

Як цитувати: Kotenko D. A., Zipunnikov M. M. Application of a genetic algorithm for solving the problem of scaling hydrogen systems. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.66-75. https://doi.org/10.26565/2304-6201-2025-67-06

How to quote: D. A. Kotenko, M. M. Zipunnikov "Application of a genetic algorithm to solve the problem of scaling hydrogen systems", *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 67, pp. 66-75, 2025. https://doi.org/10.26565/2304-6201-2025-67-06

1 Introduction

Green hydrogen is one of the most promising sources of clean energy. Growing demand for energy, the need to reduce greenhouse gas emissions, and the desire for sustainable development are driving the active implementation of hydrogen technologies. The most common method of hydrogen production is

the electrolysis of water, which requires sufficient electricity [1]. If electricity sources are not enough, this can cause an additional load on the power system, especially during peak consumption periods.

The simultaneous use of several appliances creates a large electrical and mechanical load on the power system. Unevenly distributed power consumption can lead to an increase in peak power and the occurrence of shock mechanical loads on the power system (which in turn can cause an impact on the turbines of the generating unit and cause their failure).

To avoid such scenarios, installations need to have a controlling entity (controller) that will manage the startup queue in such a way as to minimize the amount of power used simultaneously and avoid shocks during the completion of the installation's cycles. This controller performs the task of finding the best possible startup queue.

2 Problem formulation and literature review

Optimization techniques are crucial in engineering, business, and science because they help improve efficiency, reduce costs, and enhance performance. Optimization techniques ensure better performance, lower costs, and smarter decision-making across industries.

As 87% of existing hydrogen-generating plants currently use hydrogen on-site (instead of generating and then transporting and selling it)[1], there is a need for optimization in this area to improve energy efficiency and sustainability.

The most common method of hydrogen production is the electrolysis of water, which requires sufficient electricity [2]. If electricity sources are scarce, this can put additional strain on the power grid, especially during peak consumption periods.

Optimization helps to reduce energy consumption and carbon footprint.

The most promising method for this is the integration of a smart grid-based control system that optimizes the distribution of electricity. [3, p.1]

Various optimization and computational intelligence techniques has already been incorporated into large-scale grids; for example using artificial intelligence, heuristic, and evolutional optimization to analyze optimal power flow, power flow, SE, stability, and unit commitment.

In his guide to smart grids, James Momoh notes that: The classical optimization tools currently used cannot handle the adaptability and stochasticity of smart grid functions. Thus, the computational tools and techniques required are defined as a platform for assessment, coordination, control, operation, and planning of the smart grid under different uncertainties. [3, p.100]

In modern studies, improvements in hydrogen systems are analyzed from the perspective of cost efficiency in systems utilizing renewable energy sources [4,5] and the reduction of hydrogen logistics costs [6,7] through the application of linear programming and PSO methods.

However, it is important to note that these works primarily focus on hydrogen production systems based on a single electrolyzer and do not aim to assess the feasibility of using multiple units. As a result, the topic of cost optimization and maintenance strategies for multi-electrolyzer systems remains less explored, along with the associated challenge of their dispatching.

If we abstract from the hydrogen-specific context and focus on dispatching as an optimization objective, insights can be drawn from dispatching methodologies applied in power systems [8,9,10], emergency management [11], and construction [12]. These fields offer a well-established foundation for the practical application of stochastic optimization algorithms such as Lyapunov optimization, PSO, and GA in solving complex optimization problems.

3 The research aim and problem statement

The purpose of this study is to develop a mathematical and software tool to minimize the amount of power consumed by a hydrogen-generating system.

An optimization problem is a mathematical task in which it is necessary to find the best (optimal) solution among all possible options, taking into account certain constraints and the optimality criterion (objective function).

The optimality criterion in determining the best start-up shift for electrolysis units is the lowest peak power consumption by the hydrogen generating system.

Task variables:

- n - number of units.
- t - time
- $Af(t)$ - function that describes the voltage change for the electrolysis unit
- $\overline{Af}$ - a vector of functions that describe the voltage change for each unit in the system
- $\overline{I}$- a vector describing the number of amperes used by each unit to produce hydrogen
- $\overline{\varpi}$ - start time offset vector of each unit

Equation (3.1) is a function that characterizes the system costs (power) at a point in time, further Sf. Of(3.2) - an estimation function of a specific system configuration.

$$Sf\ (t, \overline{Af}, \overline{I}, \overline{\varpi}, n\ )\ =\ \sum_{i=0}^{n} |(Af_i\ (t\ +\ \varpi_i) \cdot I_i)| \quad (3.1)$$

$$Of\left(\overline{Af}, \overline{I}\ \overline{\varpi}, n\right) =\ max_{t\in[0,2\pi]} Sf\ (t, \overline{Af}, \overline{I}, \overline{\varpi}, n\ ) \quad (3.2)$$

Sf is a function that estimates a specific system configuration at a specific shift. The configuration consists of three main components. First, a set of functions $\overline{Af}$ describes the voltage change for each unit. Second, a vector $\overline{I}$ represents the number of amperes each unit uses to produce hydrogen. Finally, a vector $\overline{\varpi}$ defines the start time offset for each unit.

The functions describing the voltage change $\overline{Af}$ and the vector describing the number of amperes $\overline{I}$ are defined as the input conditions of the problem, and the start time offset vector $\overline{\varpi}$ is a parameter generated from the optimal solution space.

- M - a matrix of start time offset vectors of each unit or a function that generates a start time offset vector
- K - is the number of shift vectors.

F$(\overline{Af}, \overline{I}, M, n, k)$ (3.3) - is the estimation function by which the optimization is performed, min F$\left(\overline{Af}, \overline{I}, M, n, k\right)$ - objective function.

$$\mathrm{F}\left(\overline{Af}, \overline{I}, M, n, k\right) =\ min_{i\in[0,k]}\ Of\ (\overline{Af}, \overline{I}, M_i, n\ ) \quad (3.3)$$

When minimizing, we are interested in the amount of power consumed, since it is this amount that determines the restrictions on the grid, so we use the modulo power consumption in the following. Next, we need to define an estimating function that measures the amount of consumption.

The estimation function for this task is the maximum power value during the operation of the electrolysis system - the peak amount of power consumed. Accordingly, the objective function of optimization is the smallest peak power consumption.

4 The research aim and problem statement

The task of choosing an optimization method is to determine the most efficient approach for a particular class of problems, taking into account their mathematical properties and computing resources. Since different optimization methods have their limitations and peculiarities, choosing the right method depends on the characteristics of the objective function and constraints. In general, optimization methods can be classified into the following two types: Traditional (deterministic) methods and Stochastic (non-deterministic) methods.

Traditional (deterministic) methods are not always able to solve optimization problems efficiently. They are usually based on such properties as continuity, differentiability, smoothness, and convexity of the objective function and constraints (if any). The absence of at least one of these properties makes it difficult to apply traditional optimization methods [13]. Therefore, to further search for a solution to this problem, we checked these properties.

Continuity. The functions that describe the power consumed by the electrolyzer are periodic and without discontinuities. They are represented as the sum of sinusoidal functions (sines and cosines) with different frequencies and amplitudes. Since sines and cosines are continuous functions, their sum also retains this property. In addition, the set of possible values for the maximum of the approximation functions is compact (closed and bounded), which confirms the continuity of the corresponding function.

Differentiability. The minimum function of the target function is not differentiable since it may have a fracture at the minimum point. For example, if a pure sinusoidal signal models the behavior of an electrolyzer, then when the startup is shifted by 90 degrees, the problem equivalent to max(sin(x), cos(x)) arises. At the points where these functions are equal, a sharp transition occurs, making it undifferentiable.

Because of this, traditional optimization methods cannot be applied to this problem.

Using a direct search of possible shift operations is also inefficient because it generates numerous variants to be checked. To solve the problem in this way, it is necessary to check all possible combinations of startup time shifts of n units with k number of shifts. Accordingly, the number of such combinations is the number of placements with repetitions of n elements by k elements [14 p.14]. $A_n^k = n^k$.

Therefore, for 3 units and 21 offset options (from 0 to 60 minutes in increments of 3), the number of combinations will be ., for 4 units 194481, and for 5 units 4084101. When the quality and number of units change, the complexity of the execution time increases significantly. In this case, the complexity is $\Theta(\phi) = n^k \cdot t$ (where t is the number of steps required to calculate the estimation function).

Therefore, one of the stochastic algorithms should be chosen instead. The choice of a stochastic optimization method depends on the characteristics of the problem, such as the dimensionality of the solution space, the differentiability of the function, the constraints, and the required accuracy.

It is worth noting that due to the ability to work with complex, multidimensional or discrete optimization problems with many local optima, evolutionary algorithms are often used to solve scheduling problems. [15, p.4 Table 1].

That is why it was decided to select a genetic algorithm to solve this problem.

5 Genetic algorithm

A genetic algorithm is an evolutionary search algorithm used to solve optimization and modeling problems by sequentially selecting, combining, and varying the desired parameters using mechanisms that resemble biological evolution. The specific feature of the genetic algorithm is the emphasis on the use of the "crossover" operator, which performs the recombination of candidate solutions, the role of which is similar to the role of crossing in living nature [16].

```
population = INIT() //Initialize the population using.
best_solution = None
best_fitness = negative infinity.
FOR Number_of_generations:
    // the fitness of each individual in the population
    fitnesses = [FITNESS(population)]
    if max(fitnesses) > best_fitness:
        best solution, best_fitness = max(fitnesses)
    new_population = []

    FOR population_size / 2:
        parent1, parent2 = SELECT(population)
        child1, child2 = CROSSOVER(parent1, parent2)
        child1, child2 =MUTATE(child1, child2)
        new_population.add[ child1, child2]
    // Replace the old population with the new population.
    population = new_population
// Return the best solution and its fitness.
return best_solution, best_fitness
```

Scheme of the genetic algorithm in the form of pseudo-code

The main stages of the genetic algorithm:

Creation of the initial population. The first step is to create an initial set of solutions (chromosomes) that can be generated randomly or based on certain assumptions. In our case, it is assumed that the values are generated randomly in the range from 0 to 60 minutes (from 0 to 2). The number of chromosomes in each group corresponds to the number of electrolysis units, and the total number of solution groups is set manually and can be increased to improve search efficiency.

Performing iterations until the stop criterion is reached. The process is repeated until the algorithm's stopping criterion is met (in this case, reaching a certain number of generations or steps).

Evaluating the suitability of solutions (fitness function). For each element of the population, a fitness function value is calculated that reflects the quality of the solution in the context of the problem. In this case, the estimation function $Of(\overline{Af}, \overline{I}\,\overline{\omega}, n)$ is used.

Selecting individuals for the next generation ("selection")

The chromosomes that will be used to create the next generation are selected. Tournament selection is used in this process: several chromosomes are selected, and the best one moves on.

Crossover and/or mutation

In this implementation, both mechanisms are used.

- **Crossover:** new chromosomes are formed by combining pairs of initial solutions. Universal crossing is used (5.1), in which each gene (the offset of a particular unit i) is inherited from the parents in proportion to a random value within [0;1]:

$$\varpi_{ni} = \alpha \cdot \varpi_{1i} + (1-\alpha) \cdot \varpi_{2i} \quad (5.1)$$

- **Mutation:** a random introduction of minor changes to the genes of a chromosome. In this case, a Gaussian mutation is used, which involves changing the value of a gene within the permissible range.

Formation of a new population. A new population is created, consisting of the resulting descendants (the results of crossing and mutation) that replace the previous population.

6 Numerical results

To validate the proposed method, we will test the proposed solution to the problem of producing 354.538m^3 of hydrogen per hour. The function $Af(t)$ describing the voltage change for the electrolysis unit is given in Table 1, and the approximation based on this table $Af(t)$ (6.1).

$$Af(t) = -\frac{0.58}{2} - 0.46\cos(t) + 1.63\sin(t) + 0.19\cos(2t) - 0.15\sin(2t) + 0.44sin(3t) \quad (6.1)$$
$$+4cos(4t) + 0.03six(4t) + 0.06cos(5t) + 0.2 * sin(5t)$$

Table 1. time series of voltage changes of the full cycle of hydrogen and oxygen production during electrolysis using the Fe electrode assembly (sponge). Current density: I = 0.015 A/cm²

*Таблиця 1. Зміна напруги повного циклу виділення водню і кисню під час електролізу з використанням електродної збірки Fe (губчасте). Щільність струму: I = 0,015 А/*см²

T	0	1,5	3	4,5	6	7,5	9	10.5	12	13,5	13,5	15	16,5
U	0	0.31	0.37	0.41	0.47	0.51	0.61	0.68	0.77	0.88	0.88	1.01	1.2
T	18	19,5	21	22,5	24	25,5	27	28,5	30	31,5	33	34,5	36
U	1.31	1.42	1.51	1.57	1.71	1.4	0	-0.43	-0.78	-1.13	-1.43	-1.62	-1.71
T	37,5	39	40,5	42	43,5	45	46,5	48	49,5	51	52,5	54	
U	-1.76	-1.8	-1.8	-1.8	-1.8	-1.8	-1.8	-1.8	-1.8	-1.8	-1.8	0	

To produce 1m^3 of hydrogen, our electrolyzer consumes 4.24 kW of electricity [17]. Therefore, to produce 354.538m^3 of hydrogen, we need to spend 1503.244 kW of electricity.

In order to verify the suitability of the genetic algorithm, the first exercise compares its results with the results that can be obtained using the gradient descent algorithm. Consider a situation in which 10 identical electrolyzers produce the required amount of hydrogen, each of which has a plate area of 75162.2cm^2 . The use of the direct search method is not advisable since 60^{10} possible combinations need to be checked to calculate qualitative results (with a time step of at least 1 minute). The results of the calculations are shown in Table 2.

Table 2: Comparison of the peak power obtained by 5 rounds of optimization using the genetic algorithm and the gradient descent method.

Таблиця 2: Порівняння пікової потужності, отриманої за 5 раундів оптимізації з використанням генетичного алгоритму та методу градієнтного спуску.

genetic algorithm	gradient descent method
915.43	1011.83
903.64	953.05
918.53	952.18
913.32	1000.05
924.14	984.40

The best result obtained with gradient descent in this configuration has a maximum peak power of 952 kW (startup queue: 46.69 min, 7.09 min, 37.81 min, 32.92 min, 53.29 min, 22.43 min, 20.36 min, -0.26 min, 11.62 min, 39.56 min). The genetic algorithm provided the best result with a peak power of 903 kW (start-up queue: 15.26 min, 27.31 min, 6.55 min, 18.07 min, 30.72 min, 51.77 min, 2.54 min, 42.52 min, 53.78 min, 38.82 min).

We also investigated the algorithm's effectiveness in two more cases. The first is a configuration of three identical electrolyzers, each with a plate area of $125270.3cm^2$, and the first electrolyzer with a plate area of $375811cm^2$. The second half of the production is covered by two identical electrolyzers, given by Table 1 and Equation 1, and two PEM electrolyzers, given in Equation below (6.2).

$$Af(t) = 8.13 - 2 + 1.00 * \cos(1 * t) + 0.34 * \sin(1 * t) - 4.61 * \cos(2 * t)$$
$$-4.37 * \sin(2 * t) - 0.94 * \cos(3 * t) - 1.70 * \sin(3 * t) + 0.27 * \cos(4 * t)$$
$$+ 2.73 * \sin(4 * t) + 0.13 * \cos(5 * t) + 0.85 * \sin(5 * t) + 0.01 * \cos(6 * t)$$
$$-0.33 * \sin(6 * t) - 1.04 * \cos(7 * t) - 0.23 * \sin(7 * t) + 0.17 * \cos(8 * t)$$
$$+ 0.02 * \sin(8 * t) + 0.90 * \cos(9 * t) + 1.01 * \sin(9 * t) + 0.01 * \cos(10 * t)$$
$$+ 0.34 * \sin(10 * t) + 0.21 * \cos(11 * t) - 0.64 * \sin(11 * t)$$
$$+ 0.35 * cos(12 * t) - 0.35 * sin(12 * t) \quad (6.2)$$

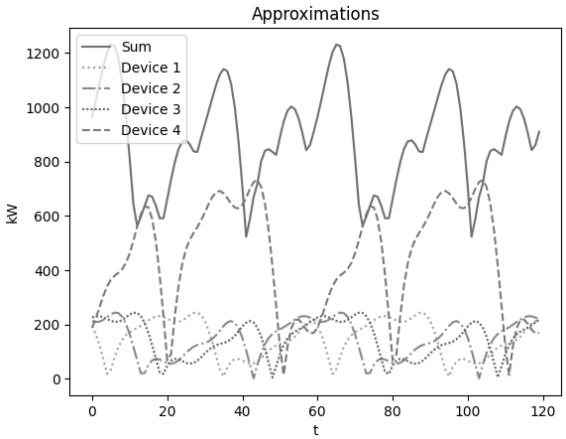


Figure 1. Optimization of a configuration of 3 identical electrolyzers, each with a plate area of $125270.3cm^2$ and an electrolyzer with a plate area of $375811cm^2$ using a genetic algorithm.

Рисунок 1. Оптимізація конфігурації 3 однакових електролізерів, кожен з яких має площу пластини 125270,3 〖см〗 ^2 , та електролізера з площею пластини 375811 〖см〗 ^2 за допомогою генетичного алгоритму.

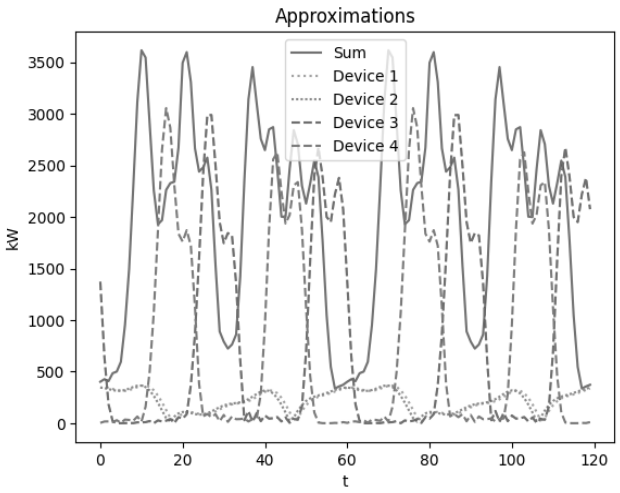


Figure 2. Optimization of a configuration of 2 identical membrane-less electrolyzers and 2 PEM electrolyzers using a genetic algorithm.

Рисунок 2. Оптимізація конфігурації 2 ідентичних безмембранних електролізерів та 2 PEM електролізерів за допомогою генетичного алгоритму.

Table 3: Numerical results of optimization of the startup queue for configurations of systems with several electrolyzers using the genetic algorithm and the gradient descent method.

Таблиця 3: Числові результати оптимізації черги запуску для конфігурацій систем з декількома електролізерами з використанням генетичного алгоритму та методу градієнтного спуску.

Configuration	Genetic algorithm	Gradient descent
3 identical electrolyzers, each with a plate area of 125270.3 cm^2, and 1 electrolyzer with a plate area of 375811 cm^2	903.64	926.81
2 identical membrane-less electrolyzers and 2 PEM electrolyzers	3256.94	3315.41

The analysis of the numerical results (presented in Table 3 and Figures 1 and 2) confirms that the genetic algorithm demonstrates higher efficiency than the gradient descent method. This is observed regardless of the number of electrolysis units in the configuration and the type of units used.

In particular, the genetic algorithm consistently provides the best values of optimized parameters, which indicates its ability to effectively find global optimal solutions, even in cases with high dimensionality of the search space and complex dependence of input parameters.

7 Conclusion

The study substantiated the feasibility of using a genetic algorithm to solve the optimization problem of calculating the effective start queue of electrolyzers in a hydrogen production system. The analysis of its effectiveness in comparison with the gradient descent showed that the genetic algorithm demonstrated better quality of the obtained solutions, especially in conditions when the function is non-uniform, has local minima, or is not differentiable.

These calculations have confirmed that the genetic algorithm has stable results and is effective in finding the global optimum, while the gradient descent may stop at local minima and require additional adjustments to achieve the optimal solution. The results confirm the feasibility of using a genetic algorithm to solve similar optimization problems, where traditional gradient methods may be less effective due to their sensitivity to local minima.

The results confirm that the genetic algorithm is a promising approach to solving optimization problems in cases where traditional methods have limitations.

By using the genetic algorithm method, we obtain results that give an approximate optimal result for a fixed number of steps. This approximate result, as shown in the problem with the placement of 10 electrolyzers, gives significant results — the peak power consumption decreased by almost 40%.

Further research can be aimed at improving the parameters of the algorithm, in particular, adaptive tuning of the mutation and crossover operators to increase the convergence rate.

СПИСОК ЛІТЕРАТУРИ

1. European Hydrogen Observatory. The European Hydrogen Market Landscape: Report 01, November 2023. November 2023. [Електронний ресурс]. Режим доступу: https://observatory.clean-hydrogen.europa.eu/sites/default/files/2023-11/Report%2001%20-%20November%202023%20-%20The%20European%20hydrogen%20market%20landscape.pdf .
2. El-Shafie M. Hydrogen production by water electrolysis technologies: A review // Results in Engineering. – 2023. – Vol. 20. – P. 101426. – DOI: 10.1016/j.rineng.2023.101426.
3. Momoh J. A. Smart Grid: Fundamentals of Design and Analysis. – Hoboken, NJ: Wiley-IEEE Press, 2012. – 250 p.
4. Almutairy N. Bidding optimization for hydrogen production from an electrolyzer // Proceedings of International Conference. – 2024. – P. 214–222. – DOI: 10.21741/9781644903216-28.
5. Yu D., Yang P., Zhu W. Capacity optimization of photovoltaic storage hydrogen power generation system with peak shaving and frequency regulation // Sustainable Energy Research. – 2025. – Vol. 12. – DOI: 10.1186/s40807-024-00141-z.
6. Liu Q., Zhou Z., Chen J., Zheng D., Zou H. Optimization operation strategy for comprehensive energy system considering multi-mode hydrogen transportation // Processes. – 2024. – Vol. 12. – P. 2893. – DOI: 10.3390/pr12122893.
7. Alamir N., Kamel S., Abdelkader S. Stochastic multi-layer optimization for cooperative multi-microgrid systems with hydrogen storage and demand response // International Journal of Hydrogen Energy. – 2025. – Vol. 100. – P. 688–703. – DOI: 10.1016/j.ijhydene.2024.12.244.
8. Research on distributed optimization scheduling and its boundaries in virtual power plants // Electronics. – 2025. – Vol. 14. – P. 932. – DOI: 10.3390/electronics14050932.
9. Xu J., Chen W., Dai H., Xu L., Xiao F., Liu L. Wireless charging scheduling for long-term utility optimization // ACM Transactions on Sensor Networks. – 2024. – Vol. 21. – DOI: 10.1145/3708990.
10. Zhang S., Chen S., Lin F., Zhao X., Li G. Collaborative optimization and scheduling of source, load and storage of distribution networks considering distributed energy and load uncertainty // Journal of Physics: Conference Series. – 2024. – Vol. 2823. – P. 012031. – DOI: 10.1088/1742-6596/2823/1/012031.
11. Guo H., Huang R., Cheng S. Scheduling optimization based on particle swarm optimization algorithm in emergency management of long-distance natural gas pipelines // PLOS ONE. – 2025. – Vol. 20. – DOI: 10.1371/journal.pone.0317737.
12. Dynamic optimization of tunnel construction scheduling in a reverse construction scenario // Systems. – 2025. – Vol. 13. – P. 168. – DOI: 10.3390/systems13030168.
13. Bansal J. C., Bajpai P., Rawat A., Nagar A. K. Sine Cosine Algorithm for Optimization. – Singapore: Springer, 2023. – (SpringerBriefs in Computational Intelligence). – DOI: 10.1007/978-981-19-9722-8.
14. Тумбрукакі А. В., Кушнірук А. С., Нєдялкова К. В. Елементи комбінаторики та біном Ньютона : методичні рекомендації для організації самостійної роботи та дистанційного навчання за курсом «Елементарна математика» здобувачів вищої освіти за першим (бакалаврським) рівнем спеціальності 014 Середня освіта (Математика). – Одеса : ФОП Бондаренко М. О., 2020. – 35 с. – Режим доступу: http://dspace.pdpu.edu.ua/handle/123456789/9904 .

15. Koop L., do Valle Ramos N. M., Bonilla-Petriciolet A., Corazza M. L., Voll F. A. P. A review of stochastic optimization algorithms applied in food engineering // International Journal of Chemical Engineering. – 2024. – Vol. 2024. – P. 3636305. – DOI: 10.1155/2024/3636305.
16. Alam T., Qamar S., Dixit A., Benaida M. Genetic algorithm: Reviews, implementations, and applications // International Journal of Engineering Pedagogy (iJEP). – 2020. – Vol. 12. – P. 57–77. – DOI: 10.3991/ijep.v10i6.14567.
17. Соловей В. В., Зіпунніков М. М., Шевченко А. А. Дослідження ефективності електродних матеріалів в електролізних системах з роздільним циклом генерації газів // Проблеми машинобудування. – 2015. – Т. 18, № 2. – С. 72–76.

REFERENCES

1. European Hydrogen Observatory, The European hydrogen market landscape, Report 01–November 2023, Nov. 2023. [Online]. Available: https://observatory.clean-hydrogen.europa.eu/sites/default/files/2023-11/Report%2001%20-%20November%202023%20-%20The%20European%20hydrogen%20market%20landscape.pdf
2. M. El-Shafie, "Hydrogen production by water electrolysis technologies: A review," Results Eng., vol. 20, p. 101426, Dec. 2023, doi: 10.1016/j.rineng.2023.101426.
3. J. A. Momoh, Smart Grid: Fundamentals of Design and Analysis. Hoboken, NJ, USA: Wiley-IEEE Press, 2012.
4. N. Almutairy, "Bidding optimization for hydrogen production from an electrolyzer," in Proc. Int. Conf., 2024, pp. 214–222, doi: 10.21741/9781644903216-28.
5. D. Yu, P. Yang, and W. Zhu, "Capacity optimization of photovoltaic storage hydrogen power generation system with peak shaving and frequency regulation," Sustain. Energy Res., vol. 12, 2025, doi: 10.1186/s40807-024-00141-z.
6. Q. Liu, Z. Zhou, J. Chen, D. Zheng, and H. Zou, "Optimization operation strategy for comprehensive energy system considering multi-mode hydrogen transportation," Processes, vol. 12, p. 2893, 2024, doi: 10.3390/pr12122893.
7. N. Alamir, S. Kamel, and S. Abdelkader, "Stochastic multi-layer optimization for cooperative multi-microgrid systems with hydrogen storage and demand response," Int. J. Hydrogen Energy, vol. 100, pp. 688–703, 2025, doi: 10.1016/j.ijhydene.2024.12.244.
8. "Research on Distributed Optimization Scheduling and Its Boundaries in Virtual Power Plants," Electronics, vol. 14, p. 932, 2025, doi: 10.3390/electronics14050932.
9. J. Xu, W. Chen, H. Dai, L. Xu, F. Xiao, and L. Liu, "Wireless charging scheduling for long-term utility optimization," ACM Trans. Sensor Netw., vol. 21, 2024, doi: 10.1145/3708990.
10. S. Zhang, S. Chen, F. Lin, X. Zhao, and G. Li, "Collaborative optimization and scheduling of source, load and storage of distribution networks considering distributed energy and load uncertainty," J. Phys.: Conf. Ser., vol. 2823, p. 012031, 2024, doi: 10.1088/1742-6596/2823/1/012031.
11. H. Guo, R. Huang, and S. Cheng, "Scheduling optimization based on particle swarm optimization algorithm in emergency management of long-distance natural gas pipelines," PLOS ONE, vol. 20, 2025, doi: 10.1371/journal.pone.0317737.
12. "Dynamic Optimization of Tunnel Construction Scheduling in a Reverse Construction Scenario," Systems, vol. 13, p. 168, 2025, doi: 10.3390/systems13030168.
13. J. C. Bansal, P. Bajpai, A. Rawat, and A. K. Nagar, Sine Cosine Algorithm for Optimization, SpringerBriefs in Computational Intelligence. Singapore: Springer, 2023, doi: 10.1007/978-981-19-9722-8.
14. A. V. Tumbrukaki, A. S. Kushniruk, and K. V. Nedialkova, Elements of combinatorics and Newton's binomial: Methodical recommendations for the organization of independent work and distance learning in the course “Elementary Mathematics” for higher education students at the first (bachelor's) level of specialty 014 Secondary Education (Mathematics). Odesa: Bondarenko M. O., 2020. [Online]. Available: http://dspace.pdpu.edu.ua/handle/123456789/9904 [in Ukrainian]
15. L. Koop, N. M. do Valle Ramos, A. Bonilla-Petriciolet, M. L. Corazza, and F. A. P. Voll, "A review of stochastic optimization algorithms applied in food engineering," Int. J. Chem. Eng., vol. 2024, p. 3636305, 2024, doi: 10.1155/2024/3636305.
16. T. Alam, S. Qamar, A. Dixit, and M. Benaida, "Genetic Algorithm: Reviews, Implementations, and Applications," Int. J. Eng. Pedagogy (iJEP), vol. 12, pp. 57–77, 2020, doi: 10.3991/ijep.v10i6.14567.

17. V. V. Solovey, M. M. Zipunnikov, and A. A. Shevchenko, "Investigation of the efficiency of electrode materials in electrolysis systems with a separate gas generation cycle," Probl. Mech. Eng., vol. 18, no. 2, pp. 72–76, 2015. [in Ukrainian]

Котенко Дмитро Анатолійович *Аспірант*
Інститут енергетичних машин і систем ім. А. М. Підгорного НАН України, вул. Комунальна, 2/10, Харків, 61046, Україна.

Зіпунніков Микола Миколайович *к.т.н., с.н.с. відділу енергетичних машин;*
Інститут енергетичних машин і систем ім. А. М. Підгорного НАН України, вул. Комунальна, 2/10, Харків, 61046, Україна.

Застосування генетичного алгоритму для розв'язання задачі масштабування водневих систем

Метою роботи є розроблення надійного інструменту для масштабування водневих систем та їх енергоспоживання за допомогою генетичного алгоритму.

Актуальність

Найпоширенішим методом виробництва водню є електроліз води, який вимагає достатньої кількості електроенергії. Якщо джерела електроенергії є недостатніми, це може створити додаткове навантаження на енергосистему, особливо в періоди пікового споживання. Оскільки 87% водневих станцій наразі використовують водень на місці (замість того, щоб генерувати його, а потім транспортувати для використання), існує потреба в оптимізації в цій галузі для підвищення енергоефективності та сталого розвитку. У сучасних дослідженнях вдосконалення водневих систем аналізуються з погляду економічної ефективності систем, що використовують відновлювані джерела енергії, та зниження витрат на водневу логістику шляхом застосування методів лінійного програмування та оптимізації рою частинок. Однак важливо зазначити, що ці роботи в основному зосереджені на системах виробництва водню на основі одного електролізера і не ставлять за мету оцінити доцільність використання декількох установок. Як наслідок, тема оптимізації витрат і стратегій технічного обслуговування багатоелектролізерних систем залишається менш дослідженою, а також пов'язана з цим проблема їх диспетчеризації.

Методи дослідження

Для розв’язання задачі пошуку найкращої черги запуску для електролізних установок використані стохастичні методи, та перевірено ефективність генетичного алгоритму для розв’язку цієї задачі.

Результати

Побудована модель оптимізації пікового споживання електроенергії електролізною системою, визначено оціночну функцію конфігурації та цільову функцію для оптимізації системи. Вибір стохастичного методу оптимізації аргументовано за допомогою перевірки цільової функції на властивості які необхідні для ефективності традиційних методів оптимізації, а саме — неперервність, диференційованість, гладкість та опуклість. Ефективність генетичного методу перевірено у порівнянні з методом градієнтного спуску на прикладах з різними конфігураціями електролізерів (однотипних та різнотипних).

Висновки

Ці розрахунки підтвердили, що генетичний алгоритм має стабільні результати і є ефективним для пошуку глобального оптимуму, в той час, як градієнтний спуск може зупинятися на локальних мінімумах і вимагати додаткових налаштувань для досягнення оптимального розв'язку. Використовуючи метод генетичного алгоритму, ми отримуємо результати, які дають наближений оптимальний результат за фіксовану кількість кроків. Цей наближений результат, як показано в задачі з розміщенням 10 електролізерів, дає значні результати — пікове споживання електроенергії зменшилося майже на 40%. Подальші дослідження можуть бути спрямовані на покращення параметрів алгоритму, зокрема, адаптивне налаштування операторів мутації та кросовера для збільшення швидкості збіжності.

Ключові слова: *оптимізація, стохастичні (недетерміновані) методи, генетичний алгоритм, енергоспоживання, водневі системи, електролізер.*

Надійшла у першій редакції 28.08.2025, в останній - 30.09.2025.
The first version has been received on 28.08.2025, the final version – on 30.09.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-07

УДК (UDC) 004.056.53:004.032.26

Ланін Євген Сергійович *студент магістратури ННІ комп'ютерних наук та штучного інтелекту, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, м. Харків, 61022*
e-mail:lanin2020ki12@student.karazin.ua;
https://orcid.org/0009-0003-2639-6218

Бакуменко Ніна Станіславівна *к.т.н., доцент, доцент кафедри комп'ютерних систем та робототехніки, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, м. Харків, 61022*
e-mail: n.bakumenko@karazin.ua ;
https://orcid.org/0000-0003-3496-7167

Застосування методів машинного навчання для детекції зловмисного програмного забезпечення в дампах оперативної пам'яті

Актуальність. У сучасних умовах постійного зростання кіберзагроз особливу актуальність набуває проблема виявлення зловмисного програмного забезпечення, яке може функціонувати приховано в оперативній пам'яті, використовуючи техніки безфайлових атак. Традиційні антивірусні рішення, що базуються переважно на сигнатурному підході, виявляються неефективними проти сучасних advanced persistent threats (APT) та нових модифікованих загроз. Це робить актуальною розробку інноваційних підходів до детекції зловмисного програмного забезпечення на основі аналізу поведінкових патернів в оперативній пам'яті з використанням методів машинного навчання.
Мета роботи: розробка та апробація системи автоматизованого виявлення зловмисного програмного забезпечення шляхом аналізу дампів оперативної пам'яті з використанням методів машинного навчання, а також порівняльна оцінка ефективності різних алгоритмів класифікації для багатокласової детекції типів загроз.
Методи дослідження: порівняльний аналіз алгоритмів машинного навчання, статичний аналіз дампів пам'яті, багатокласова класифікація, експериментальна апробація.
Результати. Створено технологічний конвеєр (pipeline) для автоматизованої обробки та класифікації дампів оперативної пам'яті. Проведено порівняльний аналіз 13 алгоритмів машинного навчання, який продемонстрував, що найкращі результати для задачі багатокласової класифікації ЗПЗ показує Random Forest з точністю 85.49% та F1-score 85.52%. Розроблена система реалізована на мові Python з використанням бібліотек scikit-learn (для класичних ML моделей), TensorFlow/Keras (для нейронних мереж) та pandas (для обробки даних).
Висновки. Дослідження підтвердило високу ефективність класичних методів машинного навчання, зокрема ансамблевих алгоритмів, для виявлення зловмисного програмного забезпечення в дампах оперативної пам'яті. Створена модель на основі Random Forest забезпечує оптимальний баланс між точністю класифікації (85.52% F1-score), швидкістю навчання (1.3 с) та обчислювальною ефективністю, демонструючи значні переваги над нейронними мережами у даному контексті. Розроблена система має високу практичну значущість і може бути інтегрована у форензічні платформи, системи моніторингу інцидентів кібербезпеки та експертні системи для автоматизованого виявлення загроз і прискорення процесу аналізу інцидентів. Результати дослідження підтверджують доцільність використання методів машинного навчання для створення систем захисту від сучасних кіберзагроз, що функціонують виключно в оперативній пам'яті.

Як цитувати: Ланін Є. С., Бакуменко Н. С. Застосування методів машинного навчання для детекції зловмисного програмного забезпечення в дампах оперативної пам'яті. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.76-82. https://doi.org/10.26565/2304-6201-2025-67-07

How to quote: Y. Lanin, and N. Bakumenko, "Machine Learning Approaches to Malware Detection in RAM" *Bulletin of V. N. Karazin Kharkiv National University, series "Mathematical modelling. Information technology. Automated control systems*, vol. 67, pp. 76-82, 2025. https://doi.org/10.26565/2304-6201-2025-67-07

Вступ

У сучасних умовах постійного зростання кіберзагроз особливу актуальність набуває проблема виявлення зловмисного програмного забезпечення (ЗПЗ), яке може функціонувати приховано, зокрема, тільки в оперативній пам'яті. Сучасні кібератаки стають все частішими, витонченішими та результативнішими, ставлячи під загрозу конфіденційну інформацію та об'єкти критичної інфраструктури. Додаткове занепокоєння викликають нові вектори атак, що експлуатують вразливості та ризики, пов'язані з новітніми технологіями, зокрема зі штучним інтелектом [1].

Зловмисники все частіше використовують техніки безфайлових атак, коли шкідливий код завантажується безпосередньо в оперативну пам'ять, не залишаючи слідів у файловій системі. Це, а також поширення програм-вимагачів (ransomware) [6], значно ускладнює виявлення традиційними засобами захисту.

Традиційні антивірусні рішення, що базуються переважно на сигнатурному підході, виявляються неефективними проти сучасних advanced persistent threats (APT) та нових модифікованих загроз [6]. Це робить актуальною розробку інноваційних підходів до детекції ЗПЗ. Особливу увагу слід приділити методам, що базуються на аналізі поведінкових патернів в оперативній пам'яті, таким як аналіз викликів API та бібліотек DLL за допомогою моделей машинного навчання.

1. Огляд літератури та існуючих методів

Метою дослідження є розробка комп'ютерної моделі, яка дозволяє виявляти ознаки присутності ЗПЗ шляхом аналізу дампів оперативної пам'яті. Основні завдання включають побудову системи збору дампів, їх попередню обробку, а також розробку алгоритмів ідентифікації підозрілих структур.

У процесі дослідження проаналізовано сучасні інструменти, зокрема Volatility та Rekall, а також підходи до використання машинного навчання для виявлення зразків ЗПЗ. Порівняльний аналіз основних інструментів аналізу пам'яті наведений в таблиці 1.

Табл.1 Порівняльний аналіз інструментів для аналізу дампів пам'яті
Table. 1 Comparative Analysis of Memory Dump Analysis Tools

Інструмент	Переваги	Недоліки	Продуктивність
Volatility	Широкий функціонал, активна спільнота	Повільна обробка великих дампів	2-5 ГБ/год
Rekall	Швидша обробка, модульна архітектура	Обмежена підтримка ОС	5-8 ГБ/год
MemProcFS	Автоматизований аналіз, низьке споживання ресурсів	Обмежений функціонал	10-15 ГБ/год

Сучасні методи виявлення шкідливого програмного забезпечення можна класифікувати на статичні, динамічні та методи машинного навчання. Статичні методи аналізують код програм без їх виконання та мають точність 85-90% для відомих загроз, але лише 30-40% для нових варіантів. Динамічні методи спостерігають за поведінкою програм під час виконання, досягаючи точності 70-80% з високою кількістю помилкових спрацювань. Методи машинного навчання використовують алгоритми класифікації для виявлення нових загроз, демонструючи найкращі результати з точністю 90-95% [7].

Аналіз останніх досліджень показує, що найперспективнішими є гібридні підходи, які поєднують різні методи детекції. Зокрема, дослідження Li et al. (2024) [8] продемонстрували ефективність використання графових нейронних мереж для аналізу структури процесів у пам'яті, досягнувши точності 96.2%. Aljabri et al. (2024) [2] запропонували спеціалізований підхід для детекції ransomware на основі аналізу memory features з точністю 94.8%.

Основними недоліками існуючих рішень є їх висока залежність від сигнатурних баз, що не дозволяє ефективно виявляти нові та модифіковані типи ЗПЗ. Більшість аналізаторів пам'яті працюють у пост-інцидентному режимі і потребують ручного аналізу експертами. Комерційні рішення, такі як FireEye та CrowdStrike, хоча й демонструють високу ефективність, мають значну вартість та потребують спеціалізованої інфраструктури.

2. Структура pipeline для детекції malware методами машинного навчання

Запропонована модель ґрунтується на принципах статичного аналізу дампів оперативної пам’яті із застосуванням методів машинного навчання, що забезпечує можливість виявлення ознак зловмисної активності без необхідності виконання коду в контрольованому середовищі. Архітектура системи побудована за модульним принципом і включає кілька послідовних етапів обробки даних, кожен з яких виконує окрему функцію у загальному процесі аналізу та класифікації.

На першому етапі реалізується модуль попередньої обробки дампів пам’яті, який відповідає за завантаження первинних даних, їх очищення, нормалізацію та стандартизацію. На цьому етапі також здійснюється екстракція ключових ознак, що характеризують поведінку процесів у пам’яті, таких як кількість потоків виконання, активні дескриптори, використані бібліотеки та інші структурні параметри. Метою цього етапу є підготовка однорідного та придатного до аналізу набору даних, який може бути безпосередньо використаний у подальших модулях системи.

Другий етап представлений компонентом feature engineering, що забезпечує поглиблене опрацювання та розширення простору ознак. У межах цього процесу здійснюється виділення статистичних характеристик процесів, аналіз поведінкових патернів, а також кодування категоріальних змінних для узгодження їх з вимогами алгоритмів машинного навчання. Застосування цього підходу дозволяє зменшити вплив надлишкових або корельованих параметрів, підвищуючи точність і стійкість побудованих моделей.

На третьому етапі функціонує модуль машинного навчання, у якому реалізовано порівняльний аналіз різних алгоритмів класифікації. Дослідження охоплює як класичні методи машинного навчання, так і архітектури штучних нейронних мереж. Для кожного алгоритму проводиться оцінювання продуктивності, точності, стабільності та часу навчання, що дозволяє визначити оптимальні підходи до вирішення задачі багатокласової класифікації зразків зловмисного програмного забезпечення. Особлива увага приділяється аналізу ефективності ансамблевих методів, які продемонстрували високу точність при помірних витратах обчислювальних ресурсів.

Завершальним компонентом є оцінювання та звітності, яка відповідає за інтерпретацію результатів класифікації. На цьому етапі здійснюється багатокласова класифікація типів загроз із використанням метрик якості, таких як accuracy, precision, recall та F1-score, що дає змогу об’єктивно оцінити ефективність кожної моделі. Крім того, система генерує детальні звіти з візуалізацією отриманих результатів, що спрощує аналіз та подальше вдосконалення алгоритмів.

Запропонована модульна архітектура системи забезпечує комплексний підхід до виявлення зловмисного програмного забезпечення через статичний аналіз дампів оперативної пам'яті. Послідовна реалізація чотирьох основних етапів – попередньої обробки даних, інженерії ознак, машинного навчання та оцінювання результатів створює ефективний технологічний конвеєр для автоматизованої детекції та класифікації загроз, що забезпечує повний цикл обробки даних – від збору та підготовки дампів пам’яті до формування підсумкових аналітичних звітів, що робить систему гнучким і масштабованим інструментом для дослідження та виявлення зловмисного програмного забезпечення.

3. Тестовий набір даних

У дослідженні використовується датасет Obfuscated-MalMem2022 [3], який містить понад 58 000 записів з 58 ознаками, що характеризують поведінку процесів Windows в оперативній пам'яті. Ключовими перевагами датасету є збалансованість класів, наявність розширеної категоризації типів шкідливого ПЗ та відсутність пропущених значень, що спрощує процес побудови та валідації моделей машинного навчання.

Набір даних охоплює різноманітні метрики: кількість потоків виконання (threads), список завантажених бібліотек (loaded modules), використані дескриптори ресурсів (handles), ознаки ін'єкції коду (code injection indicators), характеристики служб (services) та модулів ядра (kernel modules). Набір даних є збалансованим відносно класів з можливістю багатокласової класифікації типів ЗПЗ. Дослідження реалізовано як задачу мультикласової класифікації з чотирма основними категоріями[4]:

- Benign (легітимні процеси);
- Trojan (троянські програми);
- Spyware (шпигунське ПЗ);

- Ransomware (програми-вимагачі).

4. Використання методів машинного навчання для класифікації ЗПЗ

Формально задачу багатокласової класифікації можно сформулювати так:

$$f : X \to Y \tag{1}$$

де $X \in \mathbb{R}^{58}$ – вектор ознак процесу, $Y \in \{0,1,2,3\}$ - мітка класу (0 - Benign, 1 - Ransomware, 2 - Spyware, 3 - Trojan).

Для багатокласової класифікації використовуються метрики weighted averaging :

$$Precision_{weighted} = \frac{\sum_{i=1}^{K} n_i Precision_i}{\sum_{i=1}^{K} n_i} \tag{2}$$

де K – кількість класів, n_i – кількість зразків класу i.

$$Recall_{weighted} = \frac{\sum_{i=1}^{K} n_i * Recall_i}{\sum_{i=1}^{K} n_i} \tag{3}$$

де K – кількість класів, n_i – кількість зразків класу i.

Для класифікації випадків були застосовані методи випадкового лісу (Random Forest), градієнтного бустінгу (Gradient Boosting), метод дерев рішень (Decision Tree), k найближчих сусідів (k-Nearest Neighbors) та нейромережеві методи [5]. Для навчання та оцінки моделей використовувалася стратифікована розбивка даних з співвідношенням 80/20 для навчальної та тестової вибірок. Для забезпечення відтворюваності результатів та об'єктивності порівняння, у всіх експериментах використовувалися фіксовані параметри ініціалізації генератора псевдовипадкових чисел. Метрики оцінювання включали: accuracy, precision, recall, F1-score з weighted averaging для коректної обробки багатокласової задачі. Також враховувався час навчання кожної моделі для оцінки практичної застосовності. Результати порівняльного аналізу роботи алгоритмів наведені в таблиці 2.

Табл. 2 Результати порівняльного аналізу алгоритмів машинного навчання
Table. 2 Comparative Analysis Results of Machine Learning Algorithms

Алгоритм	Тип	Accuracy	Precision	Recall	F1-Score	Час навчання (с)
Random Forest	Classical ML	0.8549	0.8558	0.8549	0.8552	1.3
Gradient Boosting	Classical ML	0.8457	0.8463	0.8457	0.8458	220.62
Decision Tree	Classical ML	0.8445	0.8458	0.8445	0.8449	1.65
K-Nearest Neighbors	Classical ML	0.8119	0.8131	0.8119	0.8117	6.7
Extra Trees	Classical ML	0.8044	0.8117	0.8044	0.8046	0.59
Wide & Deep Network	Neural Network	0.7707	0.7780	0.7707	0.7688	29.99
Deep NN (with BatchNorm)	Neural Network	0.7654	0.7737	0.7654	0.7581	45.86
Feedforward NN	Neural Network	0.7591	0.7778	0.7591	0.7533	27.84

Серед класичних алгоритмів машинного навчання найкращу продуктивність показали Random Forest з оптимальним співвідношенням точності та швидкості, Decision Tree з високою інтерпретовністю при хорошій точності, та Extra Trees з найшвидшим часом навчання 0.59 секунди при прийнятній точності. Нейронні мережі продемонстрували стабільні, але менш вражаючі результати, де Wide & Deep Network показала найкращі результати серед нейронних мереж з F1-score 76.88%, проте всі нейронні мережі потребували значно більше часу на навчання від 27 до 46 секунд і мають можливості для покращення через оптимізацію архітектури та гіперпараметрів. Дослідження показало, що для задачі багатокласової класифікації зловмисного програмного забезпечення на основі аналізу дампів пам'яті найефективнішими є класичні алгоритми машинного навчання, зокрема Random Forest, що узгоджується з результатами інших досліджень у галузі кібербезпеки, де ансамблеві методи часто демонструють кращу продуктивність.

5 Аналіз результатів

Найкращі результати продемонстрував алгоритм Random Forest з точністю 85.49% та F1-score 85.52% при мінімальному часі навчання 1.3 секунди. Це підтверджує ефективність ансамблевих методів для задач багатокласової класифікації зловмисного ПЗ.

Gradient Boosting показав дуже близькі результати (F1-score 84.58%), але з значно більшим часом навчання (220.62 секунди), що робить його менш практичним для масштабного використання.

Нейронні мережі показали помірні результати з F1-score в діапазоні 76-77%, що може бути пов'язано з відносно невеликим розміром датасету або потребою в додатковому налаштуванні гіперпараметрів.

6. Висновки та напрямки подальших досліджень

Результати проведеного дослідження засвідчили високу ефективність використання методів машинного навчання для розв’язання задачі багатокласової класифікації зловмисного програмного забезпечення на основі аналізу дампів оперативної пам’яті. Запропонований підхід довів доцільність застосування інтелектуальних алгоритмів для автоматизованого виявлення та ідентифікації загроз, що функціонують у пам’яті системи, без необхідності прямого втручання експерта.

У межах виконаного дослідження реалізовано комплексний технологічний конвеєр (pipeline), який забезпечує проведення порівняльного аналізу тринадцяти алгоритмів машинного навчання для задачі класифікації зразків зловмисного програмного забезпечення. Проведене моделювання дало змогу визначити алгоритм Random Forest як найбільш ефективний серед протестованих моделей, що підтверджується досягнутими показниками точності (85,49%) та метрики F1-score (85,52%). Отримані результати демонструють переваги класичних методів машинного навчання над нейронними мережами у контексті даного типу задач, зокрема завдяки меншій обчислювальній складності, стабільності результатів і високій швидкості навчання. Крім того, створено практичну систему, орієнтовану на статичний аналіз дампів пам’яті, яка може бути використана як базовий компонент для побудови модулів автоматичного моніторингу загроз.

Подальший розвиток дослідження доцільно спрямувати на вдосконалення архітектур нейронних мереж і оптимізацію їх гіперпараметрів з метою підвищення точності та узагальнювальної здатності моделей. Перспективним напрямом є розширення навчального набору даних за рахунок нових типів зловмисного програмного забезпечення, що дозволить підвищити стійкість системи до нових і модифікованих варіантів загроз. Значний потенціал має також інтеграція розробленої моделі з іншими джерелами інформації — зокрема, з аналізом мережевого трафіку, системними журналами (лог-файлами) та телеметричними даними, що сприятиме формуванню комплексної системи кіберзахисту.

Особливої уваги потребує дослідження можливостей застосування методів пояснюваного штучного інтелекту (Explainable AI), які забезпечують прозорість процесу прийняття рішень і дозволяють інтерпретувати внутрішні механізми класифікації. Крім того, перспективним є використання підходів трансферного навчання для адаптації вже навчених моделей до нових типів загроз без потреби у повному перенавчанні.

СПИСОК ЛІТЕРАТУРИ

1. Гайдук О., Звєрєв В. Аналіз кіберзагроз в умовах стрімкого розвитку інформаційних технологій. *Кібербезпека: освіта, наука, техніка*. 2024. Т. 3, № 23. С. 225–236. URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/552.
2. Aljabri M., Al. E. Ransomware detection based on machine learning using memory features. *Egyptian informatics journal*. 2024. Vol. 25. P. 100445. URL: https://doi.org/10.1016/j.eij.2024.100445.
3. Canadian Institute for Cybersecurity. Malware memory analysis. URL: https://www.unb.ca/cic/datasets/malmem-2022.html.
4. Dhanya K. A., Al. E. Detection of network attacks using machine learning and deep learning models. *Procedia computer science*. 2023. Vol. 218. P. 57–66. URL: https://doi.org/10.1016/j.procs.2022.12.401.
5. Géron A. Hands-On machine learning with scikit-learn, keras, and tensorflow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Incorporated, 2022. 483 p.
6. Impact, vulnerabilities, and mitigation strategies for cyber-secure critical infrastructure / H. Riggs et al. *MDPI*. URL: https://www.mdpi.com/1424-8220/23/8/4060 (Last accessed: 29.10.2025).
7. Kumar S., Al. E. Malware classification using machine learning models. *Procedia computer science*. 2024. Vol. 235. P. 1419–1428. URL: https://doi.org/10.1016/j.procs.2024.04.133.
8. Li Q., Al E. MDGraph: a novel malware detection method based on memory dump and graph neural network. *Expert systems with applications*. 2024. P. 124776. URL: https://doi.org/10.1016/j.eswa.2024.124776.

REFERENCES

1. O. Haiduk, V. Zvieryev, "Analysis of cyber threats in the context of rapid development of information technologies", Cybersecurity: education, science, technology, vol. 3, no. 23, pp. 225–236, 2024. [in Ukrainian]. URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/552.
2. M. Aljabri, et al., "Ransomware detection based on machine learning using memory features", Egyptian Informatics Journal, vol. 25, p. 100445, 2024. DOI: 10.1016/j.eij.2024.100445.
3. Canadian Institute for Cybersecurity, "Malware memory analysis". URL: https://www.unb.ca/cic/datasets/malmem-2022.html.
4. K. A. Dhanya, et al., "Detection of network attacks using machine learning and deep learning models", Procedia Computer Science, vol. 218, pp. 57–66, 2023. DOI: 10.1016/j.procs.2022.12.401.
5. A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Incorporated, 2022, 483 p.
6. H. Riggs, et al., "Impact, vulnerabilities, and mitigation strategies for cyber-secure critical infrastructure", MDPI. URL: https://www.mdpi.com/1424-8220/23/8/4060 (Last accessed: 29.10.2025).
7. S. Kumar, et al., "Malware classification using machine learning models", Procedia Computer Science, vol. 235, pp. 1419–1428, 2024. DOI: 10.1016/j.procs.2024.04.133.
8. Q. Li, et al., "MDGraph: a novel malware detection method based on memory dump and graph neural network", Expert Systems with Applications, p. 124776, 2024. DOI: 10.1016/j.eswa.2024.124776.Гайдук О., Звєрєв В. Аналіз кіберзагроз в умовах стрімкого розвитку інформаційних технологій. *Кібербезпека: освіта, наука, техніка*. 2024. Vol. 3, no. 23. P. 225–236. URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/552.

Lanin Yevhen *Master student of the Education and Research Institute of Computer Sciences and Artificial Intelligence, V.N. Karazin Kharkiv National University, 6 Svobody sq., Kharkiv, Ukraine, 61022*

Bakumenko Nina *Ph.D, associate professor of the Department of Computer Systems and Robotics, Education and Research Institute of Computer Sciences and Artificial Intelligence, V.N. Kharkiv National University, 6 Svobody sq., Kharkiv, Ukraine, 61022;*

Machine Learning Approaches to Malware Detection in RAM

Relevance. In the current context of constantly growing cyber threats, the problem of detecting malicious software that can operate covertly in RAM using fileless attack techniques has become particularly relevant. Traditional antivirus solutions based primarily on signature-based approaches prove ineffective against modern advanced persistent threats (APT) and new modified threats. This makes it essential to develop innovative approaches to malware detection based on behavioral pattern analysis in RAM using machine learning methods.

Goal. Development and testing of an automated malware detection system through RAM dump analysis using machine learning methods, as well as comparative evaluation of the effectiveness of various classification algorithms for multi-class threat type detection.

Research methods: comparative analysis of machine learning algorithms, static analysis of memory dumps, multi-class classification, experimental validation on the Obfuscated-MalMem2022 dataset containing over 58,000 records with 58 Windows process features. Models were evaluated using accuracy, precision, recall, and F1-score metrics with weighted averaging.

Results. A fully functional technological pipeline was created for automated processing and classification of RAM dumps, including modules for data preprocessing, feature engineering, machine learning, and results evaluation. A comparative analysis of 13 machine learning algorithms was conducted, including classical methods (Random Forest, Gradient Boosting, Decision Tree, k-NN, SVM) and neural network architectures (Wide & Deep Network, CNN). It was established that the Random Forest algorithm demonstrates the best results for the multi-class malware classification task with an accuracy of 85.49% and F1-score of 85.52% at a training time of 1.3 seconds. The developed system is implemented in Python using scikit-learn libraries (for classical ML models), TensorFlow/Keras (for neural networks), and pandas (for data processing).

Conclusions. The study confirmed the high effectiveness of classical machine learning methods, particularly ensemble algorithms, for malware detection in RAM dumps. The developed Random Forest-based model provides an optimal balance between classification accuracy (85.52% F1-score), training speed (1.3 s), and computational efficiency, demonstrating significant advantages over neural networks in this context. The developed system has high practical significance and can be integrated into forensic platforms, cybersecurity incident monitoring systems, and expert systems for automated threat detection and accelerated incident analysis. The research results confirm the feasibility of using machine learning methods to create defense systems against modern cyber threats that operate exclusively in RAM.

Keywords: *machine learning, memory dump analysis, malware detection, Random Forest, multi-class classification, pipeline, digital forensics, cybersecurity, Python. Keywords: machine learning, memory dump analysis, malware detection, Random Forest, classification, pipeline, forensics, Python.*

Надійшла у першій редакції 08.09.2025, в останній - 12.10.2025.
The first version has been received on 08.09.2025, the final version – on 12.10.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-08

УДК 004.415.53

Мелкозьорова Ольга Михайлівна *кандидат технічних наук, доцент кафедри кібербезпеки інформаційних систем, мереж і технологій, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: olha.melkozerova@karazin.ua https://orcid.org/0000-0002-1134-2925*

Нарєжній Олексій Павлович *кандидат технічних наук, доцент кафедри кібербезпеки інформаційних систем, мереж і технологій, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: o.nariezhnii@karazin.ua; https://orcid.org/0000-0003-4321-0510*

Математичні моделі модуляції простих сигналів для алгебраїчного відокремлення перешкоди у системах передачі інформації

Стаття є подовження роботи [1] про сепарацію корисного сигналу від перешкоди та робіт [2,3], у яких пропонувався метод вирішення систем лінійних алгебраїчних рівнянь з використання QR розкладання на базі методу Грама Шмідта. Робота є **актуальною**, тому що на частотній осі систем передачі інформації неможна знайти ділянку, вільну від перешкод, завжди треба розраховувати на випадок, що перешкода є у всьому доступному діапазоні частот, опис деяких джерел цих перешкод наведено у вступі цієї статті. Розробка сучасних інформаційно-комунікаційних систем неможлива без використання математичних моделей, тому що це впливає на вартість дослідження та є передумовою створення дослідницьких стендів. Отже, **метою цієї роботи** є побудова моделей уявлення корисних сигналів, важливим напрямком при цьому є дотримання критеріїв математичних моделей: адекватності, гнучкості, прийнятної складності. Користь від моделювання можна отримати лише за умов, коли забезпечується правильне (адекватне) відображення властивостей оригіналу, а також відбувається видалення проблеми складності досліджень на реальних об'єктах. Тому робота подовжується у напрямку побудови аналітичних математичних моделей простих сигналів з **використанням методів модуляції**: амплітудної, частотної, фазової. У роботі є графіки з часовою розгорткою простих сигналів, формули побудови та параметри, до яких належить частота, швидкість передачі символів і період передачі одного символу, а також наведено словесний опис процесу демодуляції для оцінки правильності графіків модуляції. Отже, **результатом роботи** є аналітичні математичні моделі, які мають адекватність та прийнятну складність, також їх можна використати для побудови складніших моделей, наприклад, побудови моделі квадратурної модуляції, де спостерігається зміна вже двох параметрів: амплітуди та початкової фази. За результатом роботи можна зробити **висновки**, що робота є актуальною, має мету, результат і напрямок подальшого дослідження, що буде визначатися математичними моделями побудови системи перешкод на базі рядів Фур'є та sinc функцій, їх адитивним додаванням до корисного сигналу, з подальшим використанням матриць систем лінійних алгебраїчних рівнянь (СЛАР) і порівнянням отриманих результатів зі звичайними методами процесу демодуляції, які побудовані на використанні кореляційних інтегралів.
Ключові слова: Модуляція, демодуляція, амплітудна модуляція, фазова модуляція, частотна модуляція, системи передачі інформації, сепарація, перешкода.

Як цитувати: Мелкозьорова О. М., Нарєжній О. П. Математичні моделі модуляції простих сигналів для алгебраїчного відокремлення перешкоди у системах передачі інформації. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.83-90. https://doi.org/10.26565/2304-6201-2025-67-08

How to quote: O. Melkozerova, and O. Nariezhnii "Mathematical models of simple signals modulation for algebraic separation of noise in information communication systems", *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems,* vol. 67, pp. 83-90, 2025. https://doi.org/10.26565/2304-6201-2025-67-08 [in Ukrainian]

Вступ

Технічні проблеми реалізації фізичних ліній зв'язку є основою виникнення фундаментальних меж швидкості та надійності передачі [4, 5, 6]. Перешкоди, шуми і потоки помилок, що породжуються ними, у системах передачі інформації (СПІ) є свого роду бекграундом, тобто фоном, на якому будуються оптимальні сигнальні конструкції.

До технічних проблем реалізації СПІ можна віднести наступні перешкоди:

1) Джерелами спотворень і перешкод у СПІ є природні перешкоди. Це радіовипромінювання, що надходять із космосу, природні явища, землетруси, виверження вулканів, магнітні бурі.

2) Шуми електронних приладів – це апаратна основа, всіх пристроїв передачі та прийому, їм властиві власні теплові шуми. Навіть, якби не було зовнішніх перешкод, все одно прийом та передача інформації здійснювалася за наявності цих шумів, що описуються моделлю Гаусового каналу.

3) Взаємний вплив різних ліній. Дуже багато зараз абонентів, які бажають використовувати один і той самий фізичний ресурс, неминуче з'являється взаємний шкідливий вплив один на одного.

4) Детерміновані спотворення. Мається на увазі неідеальність частотних та енергетичних характеристик у лініях зв'язку, неоднорідності у маршруті поширення сигналу та інше

5) Розмноження сигналу при їх поширенні кількома маршрутами. Це особливо актуально при глобальному короткохвильовому зв'язку, коли радіо хвилі можуть відбиватися від різних шарів атмосфери, проходити різну тривалість маршруту від передавача до приймача і надходити на вхід приймача зі зміщенням за часом. Те саме спостерігається при поширенні ультра коротких хвиль в умовах міста. Випромінювання відбивається від будівель, автомобілів і досягають до приймача по кількох маршрутах зі зсувом за часом.

6) Збої, відмови електронних пристроїв та тимчасові розриви з'єднань у системах комутацій. Усі технічні пристрої характеризуються неідеальною надійністю. Цю причину виключити не можна.

7) Навмисні перешкоди. Станція радіо протидії. Це військової сфери застосування, поліції або це може бути організація недобросовісних, нечесних, незаконних заходів конкурентної боротьби.

Також є практичні обмеження на фізичні ресурси в лінії зв'язку, вони завжди матеріальні, до них відносяться обмеження:

1) потужності передавача, це особливо доречно для мобільних пристроїв;

2) обмеження на доступну ширину смуги частот, так як сигнали подаються у вигляді коливань, вони характеризуються певною частотою і чим більше ми хочемо передавати сигнали, тим більшу смугу частот ми повинні задіяти, але вона одна на всіх. І навіть при обережному регулюванні та розподілі радіо частот наш ефір дедалі більше стає непридатним для екстенсивного розвитку систем комунікацій.

Винахід у роботі [1] пропонує метод, який пов’язано з системами для покращення передачі та чистого відділення шуму та корисного сигналу. Базова ідея, яка підкреслюється у роботі для нового шляху розробки теорії та техніки комунікації, це відхилення методу, який побудовано на імовірності, для оцінки сигналу згідно з правилом найбільшої ймовірності. Це математична процедура для абсолютно чіткого відокремлення сигналу та шуму та доказ відсутності будь-яких фундаментальних теоретичних обмежень на ефективність комунікації, включно відсутність обмежень ємності каналу. Такий підхід розглядає нову концепцію та технічні аспекти імплементації інформаційно-комунікаційних систем та використовує системи алгебраїчних рівнянь (СЛАР) для того, щоб відфільтрувати сигнал від шуму. Матриця СЛАР – це лінійна алгебраїчна матриця, що сепарує та виділяє правдиві значення інформативних параметрів сигналу.

У роботах [2,3] запропоновано рішення системи лінійних рівнянь, для такого математичного сепаратора, оскільки матриці, які для цього використовуються мають прямокутну форму, звичайні методи вирішення системи рівнянь не будуть ефективними, у цих роботах пропонувався метод ортогонального розкладення. Для отримання матриць з ортогональним розкладенням можна використовувати метод Грама Шмідта для матриць з будь-яким розміром навіть якщо є стовпці або строки, що повторюються у матриці. Метод для вирішення СЛАР містить повний опис рішення та придатний для довільного розміру матриць. У роботі є приклад з вирішення з малим розміром матриці. Також є приклад імплементації з матрицею набагато більшого розміру у середовищі MathCad Prime. Імплементація містить функції, які можна використовувати для інших

мов програмування. Отримане рішення має мінімальну норму та придатне для лінійних алгебраїчних матриць, що сепарують сигнал від шуму.

У цій роботі пропонується розглянути найпростіші моделі сигналів, їх модуляції та демодуляції для передачі двійкових чисел, а саме: амплітудну, частотну та фазову. Ці параметри – ступені свободи, які можна змінювати відповідно до повідомлення, яке передається. Під модуляцією будемо називається процес в результаті якого параметр одного сигналу, який називається переносником, змінюється за законом, що задається іншим сигналом, який називають сигналом повідомлення [6]. Звісно, що тут треба додати ще і перенесення спектру частот, але для простоти ми не розглядаємо цей процес. Саме таке математичне моделювання процесів побудови сигналів можна використати для побудови процесу прийняття рішень при демодуляції процесу, що описаний у роботах [1,2,3].

У якості подальшого дослідження будуть розглядатися математичні моделі побудови системи перешкод та їх адитивне додавання до корисного сигналу, з подальшим використанням матриць СЛАР та їх порівняння зі звичайними методами процесу демодуляції, які побудовані на використанні кореляційних інтегралів.

1. Амплітудна модуляція (АМ) та демодуляція системи передачі інформації

Першу модель позначимо, як амплітудну модуляцію (АМ) із двома градаціями амплітуди, сигнал з пасивною паузою, тобто одна з амплітуд - це 0, друга амплітуда при передачі – 1, яка визначається з умови забезпечення середньої потужності або середньої амплітуди:

$$S_{AM}(t,T_m) = \sqrt{2}\cdot X_{\left\lfloor \frac{T_m - t}{T} \right\rfloor} \cdot \sin(\omega_0 (T_m - t)), \tag{1.1}$$

ω_0 - колова частота, що дорівнює:

$$\omega_0 = 2\cdot\pi\cdot\omega \tag{1.2}$$

ω - частота, що переносить;

T – період, за який переноситься один біт повідомлення;

X – бітовий потік, який є повідомленням, передається;

t – час, на якому можна розглянути процес модуляції;

$\lfloor . \rfloor$ - округлення у бік найближчого цілого значення.

T_m – цей параметр введено для цілей моделювання, ця величина має бути кратною періоду T.

Амплітудний множник $\sqrt{2}$ тут для забезпечення середнього значення квадрата амплітуди рівного 1, що відповідає середній потужності сигналу 0,5, оскільки будь-яка гармонійна функція в квадраті, проінтегрована на цілій кількості періодів дає величину 0,5.

Також можна визначити параметр швидкість модуляції V – це число елементарних символів, що передаються у одиницю часу.

$$V = \frac{1}{T}. \tag{1.3}$$

На рисунку 1.1 зображено модульований сигнал з використанням амплітудної модуляції. Задані параметри цього сигналу: частота 1600 герц, що переносить, швидкість маніпуляції 200 символів/с, відповідно тривалість одного канального символу – це 1/200=0,005 с. На графіку розгорнуто в інверсному часі передача перших 32 (4 байта) символів. Перший символ 0 тут йде справа, оскільки він був сформований раніше, у канал вийшов першим, далі йде 2 одиниці, це перший та другий символ. З часової розгортки видно, що байтовий код ASCII буде 116, 115, 111, 104, це символи, які відповідаю повідомленню ‘host’.

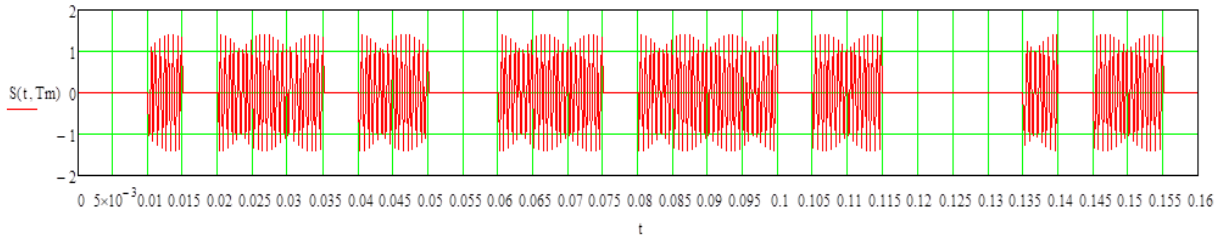


Рис.1.1 Сигнал, що модульовано з використанням АМ
Fig. 1.1 Signal with an amplitude modulation

Щоб отримати зворотній сигнал, демодулятор обчислює ступінь близькості в метриці Гілберта між прийнятою реалізацією сигналу на одному інтервалі модуляції і можливими еталонами. Нижче наведено скріншот блоку виконання процесу демодуляції з використанням MathCad (рис. 1.2). У циклі n – це кількість байтів у повідомленні.

$$\begin{aligned}&Xp := \left|\begin{array}{l}\text{for } i \in 0..n\cdot 8-1\\ \quad \left|\begin{array}{l} x_i \leftarrow \frac{2}{T}\cdot\int_{i\cdot T}^{(i+1)\cdot T} SS(t+z\cdot T)\cdot\sqrt{2}\cdot\sin(\omega 0\cdot t)\,dt \\ xx_i \leftarrow \left|\begin{array}{ll}1 & \text{if } x_i \ge 1\\ 0 & \text{otherwise}\end{array}\right.\end{array}\right.\\ xx\end{array}\right.\end{aligned}$$

Рис.1.2 Скріншот виконання АМ демодуляції
Fig. 1.2 Screenshot of an amplitude demodulation

2. Частотна модуляція (ЧМ) та демодуляція системи передачі інформації

Формування простого сигналу з частотною маніпуляцією визначається наступним описом. Масив вектор двійкових символів передачі керує зміною частоти дискретно на кожному інтервалі модуляції при заданій швидкості V, змінюючи цю частоту щодо середнього значення на величину $\pm\Delta\omega$ (при передачі 1 та 0 відповідно), яку називають девіацією частоти [4]:

$$\omega = \begin{cases}\omega_0 - \Delta\omega \\ \omega_0 + \Delta\omega\end{cases}. \tag{2.1}$$

Величина $\Delta\omega$ обирається з принципу максимальної відмінності сигналу 0 та 1 на інтервалі T. Для визначення ступеня схожості можна обчислити кореляції двох гармонійних коливань із частотами, що відрізняються на $\Delta\omega$:

$$K(\Delta\omega) = \frac{1}{E}\int_0^T \sin\left[(\omega_0+\Delta\omega)\right]\cdot\sin\left[(\omega_0-\Delta\omega)\right]dt, \tag{2.2}$$

$$E = \int_0^T \sin^2(\omega_0 t)\,dt \tag{2.3}$$

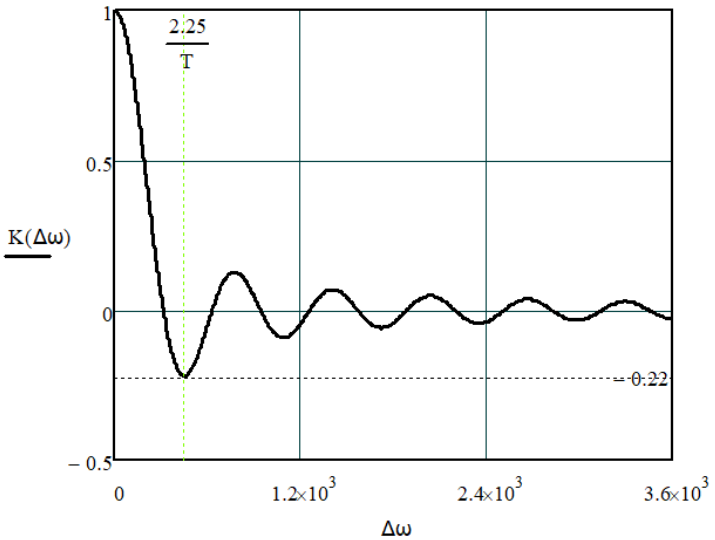


Рис.2.1 Графік взаємної кореляції двох функції, що відрізняються на $\Delta\omega$
Fig. 2.1 The graph of the cross-correlation of two functions that differ by $\Delta\omega$

Тоді процес частотної модуляції задаємо формулою [4]:

$$S(t,T_m)=\sin\left[\left(\omega_0+\left(2\cdot X_{\left\lfloor\frac{T_m-t}{T}\right\rfloor}-1\right)\cdot\Delta\omega\right)\cdot\left(T_m-t-T\cdot\left\lfloor\frac{T_m-t}{T}\right\rfloor\right)\right] \quad (2.4)$$

На рисунку 2.1 зображено модульований сигнал з використанням частотної модуляції. Задані параметри цього сигналу, як у розділі 1: частота 1600 герц, що переносить, швидкість маніпуляції 200 символів/с, відповідно тривалість одного канального символу – це 1/200=0,005 сек. На графіку розгорнуто в інверсному часі передача перших 2 символів. Перший символ 0 тут йде справа, оскільки він був сформований раніше, у канал вийшов першим, далі йде одиниця. Тут показано тільки передача двох бітів для тієї мети, щоб можна було на графіку побачити, що при передачі 0 на інтервалі від 0,005 до 0,01 частота відрізка тригонометричної функції трохи нижча, ніж при передачі 1 на інтервалі від 0 до 0,005.

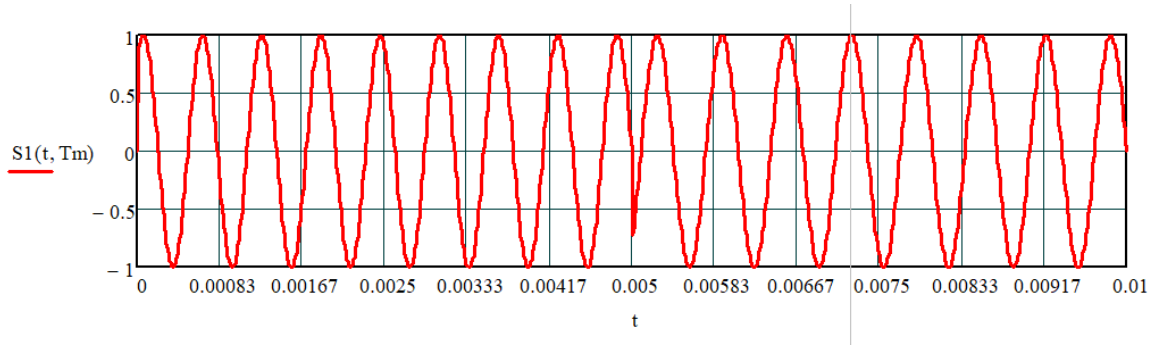


Рис.2.2 Сигнал, що модульовано, з використанням ЧМ
Fig. 2.2 Signal with the frequency modulation

На рисунку 2.3 наведено блок виконання процесу демодуляції з використанням пакету MathLab.

$$Xp := \left|\begin{array}{l}\text{for } i \in 0..n\cdot 8-1 \\ \quad \left|\begin{array}{l} x0 \leftarrow \frac{2}{T}\cdot\int_{i\cdot T}^{(i+1)\cdot T} SS1(t+z\cdot T)\cdot\sin\left[(\omega 0-\Delta\omega)\cdot\left(t-T\cdot\text{floor}\left(\frac{t}{T}\right)\right)\right]dt \\ x1 \leftarrow \frac{2}{T}\cdot\int_{i\cdot T}^{(i+1)\cdot T} SS1(t+z\cdot T)\cdot\sin\left[(\omega 0+\Delta\omega)\cdot\left(t-T\cdot\text{floor}\left(\frac{t}{T}\right)\right)\right]dt \\ xx_i \leftarrow \left|\begin{array}{ll} 1 & \text{if } x1 \geq x0 \\ 0 & \text{otherwise}\end{array}\right.\end{array}\right. \\ xx\end{array}\right.$$

Рис.2.3 Скріншот виконання ЧМ демодуляції
Fig. 2.3 Screenshot of the frequency demodulation

3 Фазова модуляція (ФМ) та демодуляція системи передачі інформації

При ФМ амплітуда та частота гармонійного коливання постійні, змінюється фаза. При передачі логічного нуля припустимо вона 0, тобто передається синус несучої частоти, при передачі логічної 1, припустимо, що вона дорівнює Пі, тобто передається мінус синус несучої частоти. Хоча цей розподіл може бути протилежним [4]. Це несуттєво.

$$S(t,T_m)=\sin\left[\omega_0\cdot\left(T_m-t-T\cdot\left\lfloor\frac{T_m-t}{T}\right\rfloor\right)+X\left\lfloor\frac{T_m-t}{T}\right\rfloor\cdot\pi\right] \tag{3.1}$$

На рисунку 3.1 зображено модульований сигнал з використанням фазової модуляції. Задані параметри цього сигналу співпадають із параметрами у розділах 1 та 2. На графіку розгорнуто в інверсному часі передача перших 24 біта. З отриманого графіка видно, що сигнал складається з відрізків синуса при передачі 0 та мінус синуса при передачі 1, байтовий код ASCII буде 111, 114, 116, це символи, які відповідаю повідомленню ‘ort’.

При демодуляції сигналу ФМ проводиться оцінка модуляційного параметра шляхом обчислення кореляційного інтеграла, прийнятого відрізка сигналу з еталоном форми несучого коливання і відбувається прийняття рішення з урахуванням знаку (рис.3.2).

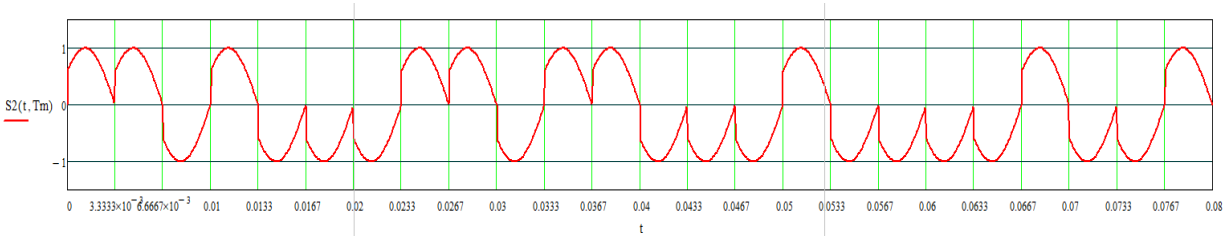


Рис.3.1 Сигнал, що модульовано, з використанням ФМ
Fig. 3.1 Signal with the phase modulation

$$\begin{aligned}&Xp:=\ \Big|\ \text{for}\ \ i\in 0\,..\,n\cdot 8-1\\&\quad\Big|\ x0\leftarrow\frac{1}{T}\cdot\int_{i\cdot T}^{(i+1)\cdot T}SS2(t+z\cdot T)\cdot\sin\left[\omega0\cdot\left(t-T\cdot\text{floor}\left(\frac{t}{T}\right)\right)\right]dt\\&\quad\Big|\ xx_i\leftarrow\frac{-\text{sign}(x0)+1}{2}\\&\Big|\ xx\end{aligned}$$

Рис.3.2 Скріншот виконання ФМ демодуляції
Fig. 3.2 Screenshot of the phase demodulation

Висновки

Стаття є подовженням роботи у напрямку розробки СПІ з методологією, запропонованою у [1], а саме підвищення пропускної здатності за рахунок використання матриць СЛАР, що відокремлюють корисний сигнал від перешкод, які присутні у каналі передачі інформації. Джерела цих перешкод різноманітні, деякі з них наведено у вступі цієї роботи. Розробка сучасних інформаційно-комунікаційних систем неможлива без використання математичних моделей, це впливає на вартість дослідження та є передумовою створення дослідницьких стендів. Тому у роботі запропоновано математичні моделі модуляції та демодуляції простих сигналів: амплітудну, частотну та фазову. Робота містить деталізований опис аналізу отриманих часових розгорток для підтвердження адекватності моделей, формули побудови та параметри, до яких належить частота, швидкість передачі символів та період передачі одного символу. Наведено словесний і математичний опис процесу демодуляції для оцінки правильності графіків модуляції.

За результатом роботи можна зробити висновки, що робота є актуальною, має мету, результат і напрямок подальшого дослідження, що буде визначатися математичними моделями побудови системи перешкод на базі рядів Фур'є та sinc функцій [5], їх адитивним додаванням до корисного сигналу, з подальшим використанням матриць систем лінійних алгебраїчних рівнянь (СЛАР) і порівнянням отриманих результатів зі звичайними методами процесу демодуляції, які побудовані на використанні кореляційних інтегралів.

СПИСОК ЛІТЕРАТУРИ

1. System and method for achieving a clean separation of signal and noise: Patent № 11,394,415 B2 The United States:of America. Date of patent Jul. 19, 2022. https://patents.google.com/patent/US11394415B2/en?oq=US11394415B2+
2. Melkozerova O. M., Rassomakhin S. G, Shlokin V. N. The application of the orthogonal decomposition method for the algebraic solver separator. Bulletin of V.N. Karazin Kharkiv national university, series "Mathematical modeling. Information technology. Authomated control systems". 2022. Vol. 54. P. 35–43. https://doi.org/10.26565/2304-6201-2022-54-04
3. Мелкозьорова О. М., Малахов С. В., Нарєжній О. П. Адаптація ортогонального розподілу матриць для задач розв'язання СЛАР. "Actual problems of modern science"*: 2023 рік* : матеріали Міжн. наук.-практ. конф., 31 січня–3 лют. 2023 р. Бостон : США, 2023. С. 469–474. https://isg-konf.com/uk/actual-problems-of-modern-science/
4. Rassomakhin S. G. Mathematical and physical nature of the channel capacity. Telecommunications and Radio Engineering. 2017. Vol. 71, Issue 16. P. 1423–1451. http://dl.begellhouse.com/journals/0632a9d54950b268,69741fd55cd51128,35cd933625b8086a.html
5. Shannon C. E. A mathematical theory of communication. Bell System Technical Journ. 1948. Vol. 27. P. 379–423, 623-656. https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf
6. Bernard Sklar Digital communication. Fundamentals and application. Communication engineering services. Tarzana, California and university of California, Los Anges, 2003. 1104 p. https://www.mangoud.com/EENG373_files/Book-Sklar.pdf

REFERENCES

1. Brifman, J., Rassomakhin, S. G., Shlokin, V. N., "System and method for achieving a clean separation of signal and noise," U.S. Patent 11 394 415 B2, Jul. 19, 1922. [in English] https://patents.google.com/patent/US11394415B2/en?oq=US11394415B2+
2. Melkozerova O. M., Rassomakhin S. G, Shlokin V. N. "The application of the orthogonal decomposition method for the algebraic solver separator". Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated systems», Vol. 54, pp 35–43, 2022. https://doi.org/10.26565/2304-6201-2022-54-04
3. O. Melkozerova, S. Malakhov and O. Nariezhnii, " Adaptation of orthogonal matrix distribution for SLAR solution problems," Actual problems of modern science: Proceedings of the 6th International Conference on Actual problems of modern science, ICAPMS 2023, Boston, USA, January 31-February 3, 2023. pp. 469-474. [in Ukrainian] https://isg-konf.com/uk/actual-problems-of-modern-science/
4. Rassomakhin S. G. "Mathematical and physical nature of the channel capacity". Telecommunications and Radio Engineering, Vol. 71, Issue 16. pp 1423–1451, 2017. [in English] http://dl.begellhouse.com/journals/0632a9d54950b268,69741fd55cd51128,35cd933625b8086a.html
5. Shannon C. E. "A mathematical theory of communication". Bell System Technical Journal, Vol. 27. pp 379–423, 623-656, 1948. [in English] http://dl.begellhouse.com/journals/0632a9d54950b268,69741fd55cd51128,35cd933625b8086a.html

6. B. Sklar, Digital communication. Fundamentals and application. Communication engineering services. Tarzana, California and university of California, Los Anges, 2003. [in English] https://www.mangoud.com/EENG373_files/Book-Sklar.pdf

Melkozerova Olha *PhD,*
Associate Professor of Cybersecurity of information systems, networks and technologies Department V. N. Karazin Kharkiv National University, Svobody Sq 4, 61022, Kharkiv, Ukraine;

Nariezhnii Oleksii *Associate Professor of Cybersecurity of information systems, networks and technologies Department V. N. Karazin Kharkiv National University, Svobody Sq 4, 61022, Kharkiv, Ukraine.*

Mathematical models of simple signals modulation for algebraic separation of noise in information communication systems

The article is a continuation of the work [1] about the separation of the useful signal from the noise and the works [2,3], in which a method for solving systems of linear algebraic equations using QR decomposition based on the Gram-Schmidt method was proposed. The work is **relevant** because on the frequency axis of information communication systems it is impossible to find a section free from interference, it is always necessary to count on the case that the noise is in the entire available frequency range, a description of some sources of this noise is given in the introduction to this article. The development of modern information and communication systems is impossible without the use of mathematical models, because this affects the cost of research and is a prerequisite for the creation of research stands. The **goal** of this work is to build models for representing useful signals, an important direction in this is compliance with the criteria of mathematical models: adequacy, flexibility, acceptable complexity. The benefit from modeling can be obtained only under conditions when the correct (adequate) reflection of the properties of the original is ensured, and the problem of the complexity of research on real objects is also removed. Therefore, the work is extended in the direction of constructing analytical mathematical models of simple signals using **modulation methods**: amplitude, frequency, phase. The work contains graphs with a time sweep of simple signals, construction formulas and parameters, which include frequency, symbol rate and transmission period of one symbol, and also provides a verbal description of the demodulation process to assess the correctness of the modulation graphs. Therefore, **the result of the work** is analytical mathematical models that have adequacy and acceptable complexity, they can also be used to construct more complex models, for example, constructing a quadrature modulation model, where a change in two parameters is observed: amplitude and initial phase. Based on the results of the work, it can be **concluded** that the work is relevant, has a goal, result and direction of further research, which will be determined by mathematical models for constructing an interference system based on Fourier series and sinc functions, their additive addition to the useful signal, with the subsequent use of matrices of systems of linear algebraic equations (SLAE) and a comparison of the results obtained with conventional methods of the demodulation process, which are based on the use of correlation integrals.

Keywords: *Modulation, demodulation, amplitude modulation, phase modulation, frequency modulation, information communication systems, separation, noise.*

Надійшла у першій редакції 07.09.2025, в останній - 10.10.2025.
The first version has been received on 07.09.2025, the final version – on 10.10.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-09

УДК (UDC) 004.93

Новіков Олексій Едуардович *студент ННІ комп'ютерних наук та штучного інтелекту, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, м. Харків, 61022*
e-mail: novikov2020ki11@student.karazin.ua;
https://orcid.org/0009-0003-3566-531X

Стрілець Вікторія Євгенівна *к.т.н., доцент кафедри комп'ютерних систем та робототехніки, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, м. Харків, 61022*
e-mail: viktoria.strilets@karazin.ua;
https://orcid.org/0000-0002-2475-1496

Модель чат-бота для конфігурування персонального комп'ютера із застосуванням методів NLP

Мета роботи: підвищення зручності та ефективності вибору компонентів персонального комп'ютера шляхом використання Telegram чат-бота з методами NLP для врахування запитів користувача.
Методи дослідження: методи обробки природної мови NLP для інтерпретації користувацьких запитів та формування відповідей чат-бота; методи побудови діалогових систем; підходи до організації компонентів програмного забезпечення. Telegram чат-бот реалізовано на основі клієнт-серверної архітектури, де клієнтська частина забезпечує взаємодію з користувачем у Telegram, а серверна — логіку обробки даних і підбору компонентів ПК. Для реалізації використані технології: мова програмування Python, бібліотека python-telegram-bot для створення чат-бота, інструменти NLP для аналізу та інтерпретації запитів користувача та fuzzy matching для покращення пошуку.
У **результаті** створено модель Telegram чат-бота, який автоматизує процес підбору комплектуючих для персональних комп'ютерів, враховуючи індивідуальні потреби та побажання користувача. Чат-бот дозволяє швидко отримати рекомендації щодо вибору компонентів ПК, таких як процесор, відеокарта, оперативна пам'ять, накопичувач, материнська плата та блок живлення, з урахуванням цінової категорії, призначення (ігри, робота, мультимедіа) та бажаних характеристик. Чат-бот забезпечує зручну взаємодію через Telegram, а серверна частина відповідає за обробку запитів, аналіз тексту користувача та формування оптимальних конфігурацій з використанням методів NLP і fuzzy matching. Для обробки природної мови застосовані бібліотеки та інструменти: Stanza, NLTK (токенізація, стемінг, лематизація), TextBlob; для нечіткого пошуку – RapidFuzz. Використання мови Python та бібліотеки python-telegram-bot забезпечує надійну роботу системи, гнучкість у масштабуванні та можливість швидкого оновлення бази компонентів.
Висновки: створений Telegram чат-бот дозволяє автоматизувати процес підбору комплектуючих для персональних комп'ютерів з урахуванням індивідуальних потреб і побажань користувача. Чат-бот забезпечує можливість підбору компонентів під різноманітні сценарії використання – ігри, робота, мультимедіа, бюджетні або високопродуктивні конфігурації та інше. Це дозволяє користувачам швидко отримувати якісні рекомендації, зменшує ймовірність помилок при складанні конфігурацій і полегшує процес вибору комплектуючих. Отже розроблена модель підвищує зручність користування, спрощує процес вибору компонентів та сприяє більш ефективній взаємодії користувача з системою.

***Ключові слова**: чат-бот, telegram, автоматизація, NLP, NLTK, stanza, fuzzy matching, конфігуратор ПК.*

Як цитувати: Новіков О.Е., Стрілець В.Є. Модель чат-бота для конфігурування персонального комп'ютера з застосуванням методів NLP. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.91-100. https://doi.org/10.26565/2304-6201-2025-67-09

How to quote: O. Novikov, and V. Strilets, "Chatbot model for personal computer configuration using NLP methods" *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 67, pp. 91-100, 2025. 5https://doi.org/10.26565/2304-6201-2025-67-09 [in Ukrainian]

1 Вступ

У сучасних умовах розвитку інформаційних технологій роль персональних комп'ютерів та високопродуктивних систем зростає надзвичайно швидкими темпами. Вони використовуються у різних сферах: від навчання та наукових досліджень до бізнес-процесів і розваг. Разом з тим постає проблема правильного підбору апаратного забезпечення, яке має відповідати індивідуальним потребам користувачів. З огляду на постійне оновлення ринку комп'ютерних

комплектуючих, де щорічно з'являються сотні нових моделей із різними характеристиками, завдання вибору оптимальної конфігурації стає складним навіть для досвідчених користувачів.

Традиційний підхід, що передбачає самостійний пошук інформації, аналіз технічних параметрів та порівняння цін, потребує значних часових витрат і глибоких технічних знань. При цьому користувачі мають різні запити: для одних важливо зібрати недорогий ПК для офісних завдань, для інших – забезпечити максимальну продуктивність у сучасних іграх, а для третіх – створити надійну робочу станцію для проєктування чи обробки великих обсягів даних. Додаткову складність створює необхідність перевірки сумісності комплектуючих, що часто викликає помилки у недосвідчених користувачів.

З цієї причини виникає потреба у створенні автоматизованих систем, здатних полегшити процес підбору обладнання. Використання методів, технологій обробки природної мови NLP [1, 3] та нечіткого пошуку fuzzy matching [2] відкриває нові можливості для побудови інтелектуальних інструментів взаємодії з користувачем. Такі системи дозволяють не лише інтерпретувати запити, сформульовані у довільній формі, але й надавати персоналізовані рекомендації, адаптовані під різні сценарії використання комп'ютера.

Таким чином, створення подібних інтелектуальних систем є актуальним завданням сучасної комп'ютерної науки та практики розробки програмного забезпечення. Зростання обсягів інформації та швидкий розвиток ринку комп'ютерних технологій потребують інструментів, здатних обробляти великі масиви даних та надавати користувачам релевантні результати у зручній формі. Використання методів обробки природної мови у поєднанні з механізмами пошуку та адаптивної фільтрації відкриває можливості для створення систем, що поєднують у собі простоту у використанні та глибину аналітичних можливостей.

2 Аналіз предметної області та формулювання задачі

Сучасний ринок комп'ютерних комплектуючих пропонує користувачам великий вибір інструментів для підбору та порівняння компонентів. Одним із найбільш поширених підходів є використання онлайн-магазинів та каталогів, які дозволяють фільтрувати товари за базовими характеристиками, такими як ціна, бренд або технічні параметри. Такі платформи, як Amazon, Newegg або Rozetka, надають зручний інтерфейс для пошуку компонентів, проте вони обмежені у плані персоналізації та інтерактивності. Користувачу доводиться самостійно аналізувати сумісність компонентів і вибирати оптимальні варіанти.

Крім того, існують спеціалізовані сайти порівняння компонентів та конфігуратори, наприклад, PCPartPicker [9] або Logical Increments [10], які надають більш структуровану інформацію та дозволяють перевіряти сумісність обраних компонентів. Ці сервіси полегшують процес складання ПК і дають можливість обирати готові конфігурації для різних сценаріїв використання — від ігор до офісної роботи або обробки графіки. Проте й вони не завжди можуть врахувати індивідуальні потреби користувача або обробити запити, сформульовані у довільній текстовій формі.

Додатково користувачі звертаються до форумів та спільнот, таких як Reddit або Tom's Hardware, де обмінюються порадами щодо сумісності та ефективності компонентів. Такі ресурси надають великий обсяг практичної інформації та відгуків від досвідчених користувачів. Однак цей підхід потребує значного часу на аналіз і не забезпечує автоматизації підбору комплектуючих.

Нарешті, популярність набирають чат-боти та автоматизовані консультанти у Telegram, Discord або інших платформах, які пропонують швидкі рекомендації або готові конфігурації ПК. Вони дозволяють користувачу взаємодіяти у зручній формі, отримувати відповіді на запитання та уточнювати побажання. На жаль більшість існуючих ботів працюють за простими правилами і не підтримують аналіз довільних текстових запитів. Це обмежує їхню здатність надавати персоналізовані рекомендації, враховувати сумісність компонентів та адаптуватися під унікальні потреби користувачів.

Попри наявність численних інструментів для підбору ПК, існуючі рішення не забезпечують повної персоналізації та інтерактивності. Користувачі часто змушені самостійно аналізувати сумісність компонентів, порівнювати характеристики та приймати рішення на основі обмеженої або частково структурованої інформації. Форми взаємодії з ботами та автоматизованими консультантами залишаються досить статичними і не дозволяють ефективно обробляти довільні текстові запити користувача.

У зв'язку з цим виникає задача створення інструменту, який дозволить користувачу швидко та ефективно підібрати комплектуючі ПК відповідно до власних потреб та побажань. Основна задача

полягає у створенні інтелектуальної системи, здатної аналізувати текстові запити користувача та формувати персоналізовані рекомендації щодо підбору комплектуючих ПК. Система повинна враховувати різноманітні параметри та вимоги користувача, пропонувати варіанти комплектуючих і забезпечувати інтерактивну взаємодію. Важливою вимогою є можливість обробки довільних запитів, навіть якщо вони сформульовані нечітко або неповно, що дозволяє надати максимально адаптовані рекомендації для широкого кола користувачів.

Для забезпечення здатності інтерпретувати довільні, нечіткі або неповні запити користувача ефективним є використання сучасних методів обробки природної мови NLP та алгоритмів наближеного порівняння fuzzy matching, що дозволяють розпізнавати ключові параметри та побажання користувача і формувати персоналізовані рекомендації. Такий підхід забезпечує підвищену точність підбору комплектуючих і дозволяє адаптувати результати до індивідуальних потреб кожного користувача.

Крім того, система має забезпечувати інтерактивну взаємодію з користувачем: звертати увагу на деталі, пропонувати оптимальні варіанти та надавати інформацію про технічні характеристики та співвідношення ціни і продуктивності. Виконання цих задач дозволяє значно спростити процес підбору ПК, економити час користувачів та забезпечувати більш персоналізований і зручний досвід порівняно з існуючими рішеннями, такими як статичні онлайн-каталоги чи базові чат-боти без NLP.

У роботі описану інтелектуальну систему було реалізовано як чат-бот, що поєднує сучасні методи обробки текстової інформації та інтелектуальні алгоритми пошуку. Основним завданням чат-бота є надання користувачеві можливості отримати релевантні рекомендації щодо вибору комп’ютерних комплектуючих, сформовані відповідно до його індивідуальних потреб та вподобань. На відміну від традиційних каталогів чи статичних систем пошуку, чат-бот забезпечує інтерактивну взаємодію, що дозволяє враховувати широкий спектр факторів: від загальних вимог до продуктивності до більш специфічних побажань, які користувач може сформулювати у довільній формі. Такий підхід створює передумови для персоналізації результатів і формує більш зручний користувацький досвід.

Важливою перевагою розробленої моделі чат-бота є її універсальність і можливість адаптації до різноманітних сценаріїв використання. Вона може бути корисною як для новачків, які не мають достатніх технічних знань і прагнуть швидко отримати готові рекомендації, так і для досвідчених користувачів, що хочуть проаналізувати різні варіанти та знайти найкраще рішення. Завдяки цьому чат-бот виступає не лише інструментом пошуку, а й своєрідним цифровим консультантом, що допомагає користувачам орієнтуватися у складному та динамічному ринку комп’ютерних технологій.

3 Методи дослідження та технології обробки запитів користувача

У задачі підбору комплектуючих для персонального комп’ютера користувачі формулюють свої потреби у вільній формі – природною мовою. Часто такі запити містять неточності, неповні характеристики або описові формулювання на кшталт «потрібна потужна відеокарта для ігор» чи «дешевий процесор з низьким енергоспоживанням». Традиційні методи пошуку за ключовими словами в такому випадку виявляються недостатніми, оскільки не враховують варіативність мови та можливі помилки.

Саме тому для коректної інтерпретації запитів доцільно застосовувати методи обробки природної мови NLP [1, 3] та алгоритми наближеного порівняння fuzzy matching [2]. Вони дозволяють:

- виділяти ключові характеристики з тексту,
- зводити слова до базової форми,
- враховувати контекст використання,
- знаходити найбільш релевантні компоненти навіть за умови неточного формулювання запиту.

Такий підхід забезпечує гнучкість та адаптивність системи, роблячи підбір комплектуючих більш зручним і точним для користувача.

3.1 Попередня обробка тексту

Першим кроком у роботі з текстовими даними є їхня попередня обробка, яка має підготувати користувацькі запити до подальшого аналізу. Запити у вільній формі можуть містити орфографічні помилки, різні варіанти написання слів, зайві символи чи неінформативні слова. Якщо одразу передавати такий текст у систему обробки, результати будуть неточними, тому важливо провести кілька етапів очищення й стандартизації.

Одним із базових кроків є токенізація [4] (tokenization) – процес розбиття тексту на окремі елементи (токени), найчастіше слова. Це дозволяє системі працювати не з суцільним рядком, а з окремими частинами, які можна порівнювати, аналізувати й перетворювати. У Python для цього часто використовується метод nltk.word_tokenize [6], який враховує пунктуацію та правила мови.

Другим важливим кроком є нормалізація [5] тексту. Вона включає зведення всіх символів до нижнього регістру (щоб, наприклад, слова «Intel» та «intel» розпізнавалися як однакові), видалення пунктуації, спеціальних символів, чисел, що не несуть корисної інформації, а також усунення стоп-слів (наприклад: «і», «та», «але», «для»). Такі слова не додають смислового навантаження у пошукових запитах і лише заважають точній обробці.

Таким чином, попередня обробка тексту створює основу для подальших етапів аналізу, роблячи дані більш чистими та структурованими. Це суттєво підвищує якість лінгвістичних і семантичних методів, які застосовуються на наступних етапах.

3.2 Зведення слів до основної форми

Після попередньої обробки важливим етапом є зведення слів до базової форми, що дозволяє уникнути проблеми варіативності словоформ. Наприклад, слова «процесор», «процесора», «процесори» мають різні закінчення, але фактично позначають один і той самий об’єкт. Якщо їх не привести до єдиної форми, система може сприймати їх як різні терміни, що призведе до втрати релевантності під час пошуку.

Для цього використовуються два підходи.

Стемінг [7] (stemming) – це спрощене відсікання закінчень слів без урахування граматики. Наприклад, «грає», «грав», «грати» будуть зведені до основи «гра». Такий метод швидкий, але не завжди точний, оскільки результат не завжди збігається зі словниковою формою. Інструменти:

- SnowballStemmer [7] для української мови;
- PorterStemmer [7] для англійської мови.

Лематизація [8] (lemmatization) – це точніший метод, що ґрунтується на словникових базах та граматичних правилах. Він дозволяє привести слово до його канонічної форми (леми). Наприклад, «грає», «грав», «грати» будуть зведені саме до «грати». Лематизація є кращим підходом для роботи з українською мовою, оскільки зберігає правильність форми. Інструменти:

- WordNetLemmatizer [8] для англійської мови;
- Stanza [3] для української мови (підтримує морфологічний та лематизаційний аналіз).

Отже, стемінг доцільно застосовувати у випадках, коли потрібна швидкість і допускається певна втрата точності, тоді як лематизація краще підходить для завдань, що потребують високої точності й коректності мовних форм. У роботі для запитів українською мовою більш доцільним є поєднання обох методів: стемінг для попереднього скорочення словоформ та лематизація для уточнення базового значення слова.

3.3 Лінгвістичний аналіз

На цьому етапі система переходить від роботи з окремими словами до розуміння їхніх граматичних та синтаксичних характеристик у контексті. Це важливо для правильного інтерпретування користувацьких запитів, оскільки одні й ті самі слова можуть мати різні значення залежно від ролі у реченні.

Морфологічний аналіз [9] полягає у визначенні частини мови (іменник, прикметник, дієслово тощо), а також граматичних характеристик — роду, числа, відмінка, часу. Наприклад, у запиті «підбери нову відеокарту для ігор» система має розпізнати, що слово «нову» є прикметником, який описує характеристику «відеокарти», а «для ігор» вказує на ціль використання.

Синтаксичний аналіз визначає залежності між словами в реченні. Це дозволяє встановити, які слова є головними, а які допоміжними. Наприклад, у реченні «хочу потужний процесор для роботи» система повинна зрозуміти, що слово «потужний» описує саме «процесор», а не «роботу».

Для виконання морфологічного та синтаксичного аналізу використовується stanza.Pipeline [3] – багатомовна NLP-бібліотека від Stanford NLP, яка підтримує українську мову. Вона дозволяє отримати повний розбір речення, а саме:

- визначити частини мови для кожного слова;
- встановити граматичні характеристики;
- побудувати дерево синтаксичних залежностей.

Таким чином, лінгвістичний аналіз дає змогу не лише розуміти окремі слова, але й інтерпретувати сенс усього запиту, що є критично важливим для правильної рекомендації ПК-компонентів.

3.4 Семантичний рівень

Після лінгвістичного аналізу система переходить до розуміння значення слів та фраз у контексті запиту, що дозволяє робити більш точні та персоналізовані рекомендації.

Аналіз тональності [10] (sentiment analysis) дозволяє визначати емоційне забарвлення запиту користувача. Наприклад, користувач висловлює позитивне враження, невдоволення або нейтральне запитання. Це може бути корисно для визначення нагальності або пріоритетності рекомендацій. Інструменти: TextBlob [11], який дозволяє отримати полярність та суб'єктивність тексту.

Виділення сутностей [12] (Named Entity Recognition, NER) дає змогу розпізнавати конкретні об'єкти та ключові елементи запиту, такі як назви товарів, брендів, категорії компонентів, моделі, технічні характеристики або терміни, що відносяться до часу чи організацій. Інструменти: stanza [3] та spacy [13], які підтримують багатомовний NER та дозволяють виділяти сутності навіть у складних українських текстах.

Використання семантичного рівня дає можливість системі не просто знаходити ключові слова, а й розуміти, що саме користувач має на увазі, навіть якщо запит сформульований нечітко або містить неточні формулювання. Це важливо для підбору компонентів комп'ютера, адже користувач може описувати свої потреби у вільній формі, без використання стандартних характеристик чи термінів.

3.5 Методи підвищення точності Fuzzy matching

Однією з ключових проблем у сфері підбору комп'ютерних компонентів є неструктурованість і неточність користувацьких запитів. Більшість користувачів формулює свої потреби у вільній формі, не дотримуючись єдиних стандартів написання назв моделей або технічних характеристик. Це призводить до того, що пряме порівняння рядків не дає бажаного результату.

У таких випадках fuzzy matching виступає ефективним методом для підвищення точності пошуку та відбору релевантних варіантів. Суть цього підходу полягає у знаходженні найбільш схожих рядків за певною метрикою, навіть якщо вони відрізняються за написанням. Для цього використовуються алгоритми обчислення відстані редагування (наприклад, Levenshtein distance), що враховують кількість операцій (вставка, видалення, заміна символів), необхідних для перетворення одного рядка в інший.

Завдяки цьому fuzzy matching дозволяє:

- розпізнавати орфографічні помилки у запитах;
- виявляти скорочення та різні варіанти написання назв моделей;
- враховувати різні мовні форми введення користувачем;
- підвищувати релевантність результатів, навіть якщо збіг не є буквальним.

Інструменти fuzzy matching:

- fuzzywuzzy [14] – класична бібліотека для Python, яка застосовує Levenshtein distance для порівняння рядків;
- rapidfuzz [15] – більш оптимізована та швидка альтернатива, що забезпечує високу продуктивність при роботі з великими наборами даних (наприклад, каталогами компонентів).

Практичне застосування fuzzy matching у завданні підбору ПК-компонентів дозволяє створити систему, яка буде толерантною до помилок користувача та надаватиме найбільш відповідні результати навіть у випадку неточного формулювання запиту.

4 Проектування архітектури моделі чат-бота

Реалізований чат-бот побудовано за принципом клієнт-серверної архітектури. Клієнтська частина представлена платформою Telegram – користувачі взаємодіють із ботом через стандартний інтерфейс месенджера. Серверна частина виконує всю бізнес-логіку: прийом повідомлень через Telegram API [16], обробку тексту NLP, пошук релевантних компонентів у базі даних, формування та відправку відповідей. Така архітектура дозволяє розділити інтерфейс взаємодії й обчислювальні ресурси, спростити масштабування та оновлення логіки без втручання у клієнтську частину (рис. 1).

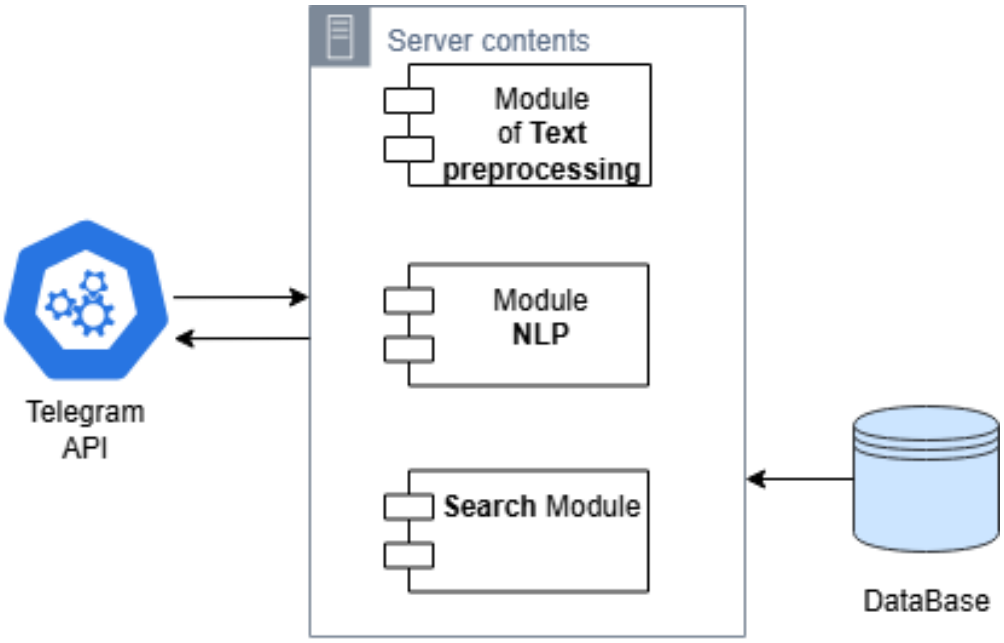


Рис. 1. Схема архітектури чат-бота
Fig. 1. The chatbot architecture diagram

Архітектура складається з таких основних частин:

- Telegram-інтерфейс – обробник вебхуків / поллінгу, що отримує повідомлення від користувачів і надсилає відповіді. Зазвичай реалізовано через бібліотеку python-telegram-bot.
- модуль попередньої обробки тексту відповідає за нормалізацію, токенізацію, видалення стоп-слів.
- NLP-пайплайн виконує стемінг, лематизація, морфологічний та синтаксичний аналіз, міститьNER-модуль [12].
- пошуковий модуль, ранжування — fuzzy matching і логіка відбору компонентів.
- база даних зберігає каталог компонентів, їх характеристик, цінових параметрів та тегів сумісності.

Запит користувача опрацьовується у кілька етапів. На рис. 2 показаний спрощений потік обробки запитів.

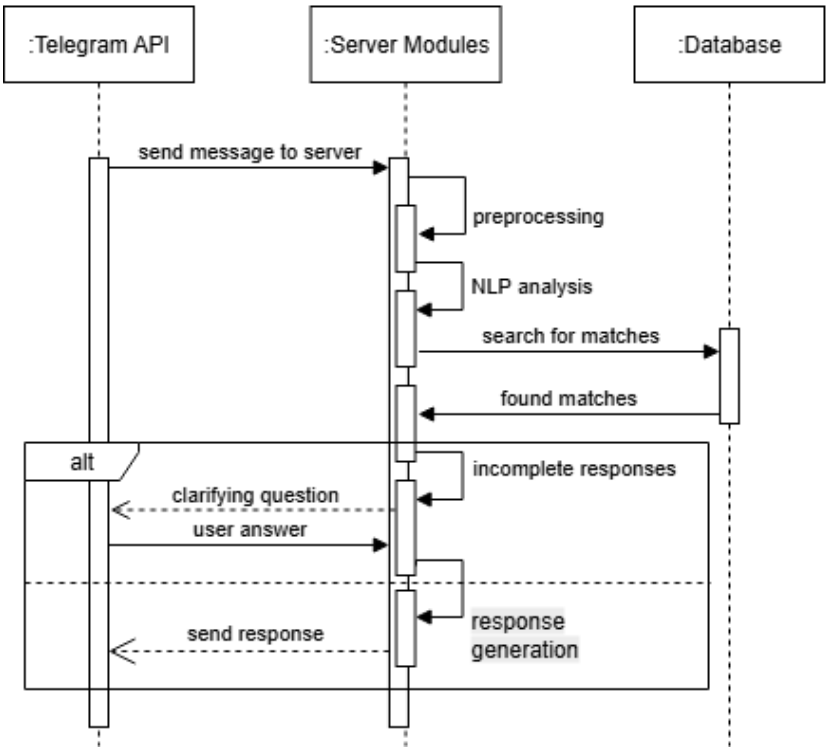


Рис. 2. Спрощена діаграма обробки повідомлень
Fig. 2. Simplified message processing diagram

Користувач надсилає повідомлення в Telegram чат-бот, після цього відбувається послідовність дій, які формують відповідь:

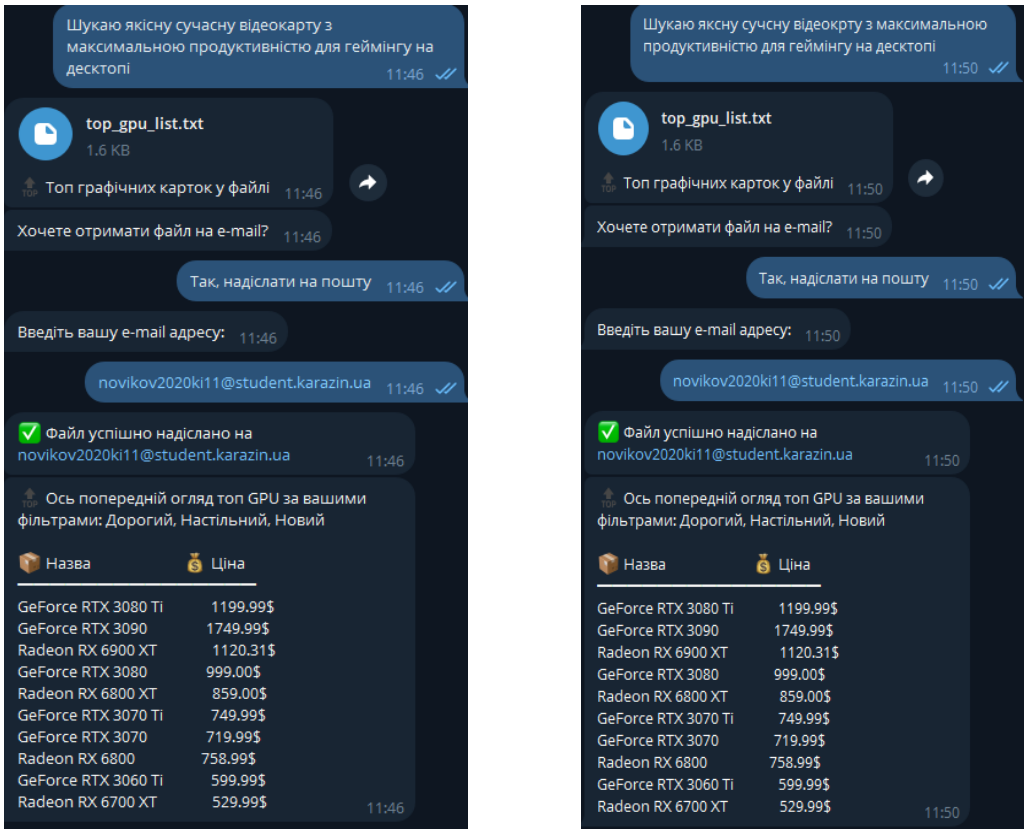
- Telegram пересилає повідомлення на сервер;
- попередня обробка: приведення до нижнього регістру, очищення, токенізація;
- NLP-аналіз: лематизація, стемінг, морфосинтаксичний і лінгвістичний аналіз;
- пошук відповідностей: поєднання структурованих параметрів із записами в базі даних; застосування fuzzy matching для наближених збігів;
- складання варіантів: ранжування знайдених компонентів за релевантністю (враховуючи сумісність, пріоритети користувача, ціновий діапазон тощо);
- діалогова взаємодія: якщо параметрів недостатньо, то бот ставить уточнювальні питання, якщо вистачає – формує відповідь з рекомендаціями.
- відправка відповіді користувачеві через Telegram.

5 Реалізація та приклади роботи системи

Розроблений чат-бот у Telegram забезпечує зручний процес взаємодії з користувачем під час вибору комп'ютерних комплектуючих. Користувач формує свій запит у довільній формі, після чого бот виконує його обробку, застосовуючи алгоритми NLP та методи нечіткого пошуку. У відповідь користувач отримує стислий опис рекомендованих компонентів, які найбільше відповідають критеріям запиту.

Для детальнішого ознайомлення бот формує і додатково надсилає файл із повним описом підібраних комплектуючих, що містить характеристики, ціну та інші параметри. Після цього користувачеві пропонується можливість надіслати цей файл на електронну пошту, і в разі згоди бот автоматично виконує відправлення.

На рис. За продемонстровано роботу чат-бота при надходженні на нього звичайного, коректного запиту користувача. На рис. 3б продемонстровано роботу чат-бота при надходженні на нього запиту з помилками. В обох випадках чат-бот демонструє правильне сприйняття запиту і коректну відповідь на нього.



а) б)

Рис. 3. Приклад роботи чат-бота: а) з коректним запитом; б) із запитом, який містить помилки
Fig. 3. Example of chatbot operation: a) with a correct query; b) with a query containing errors

На рис. 4 продемонстровано зміст файлу який формується і надається ботом користувачу.

Топ GPU за фільтрами: Дорогий, Настільний, Новий

	gpuName	G3Dmark	G2Dmark	total_performance	performance_per_dollar	gpuValue	TDP	powerPerformance	testYear	price	category
GeForce RTX 3080 Ti		26887	1031	27918	23.265194	22.41	350.0	76.82	2021	1199.99	Desktop
GeForce RTX 3090		26395	999	27394	15.653804	15.08	350.0	75.41	2020	1749.99	Desktop
Radeon RX 6900 XT		25458	1102	26560	23.707724	22.72	300.0	84.86	2020	1120.31	Desktop
GeForce RTX 3080		24853	1003	25856	25.881882	24.88	320.0	77.66	2020	999.00	Desktop
Radeon RX 6800 XT		23364	1078	24442	28.454016	27.20	300.0	77.88	2020	859.00	Desktop
GeForce RTX 3070 Ti		23367	1003	24370	32.493767	31.16	290.0	80.58	2021	749.99	Desktop
GeForce RTX 3070		22093	969	23062	32.031000	30.69	220.0	100.42	2020	719.99	Desktop
Radeon RX 6800		20667	1030	21697	28.586674	27.23	250.0	82.67	2020	758.99	Desktop
GeForce RTX 3060 Ti		20206	961	21167	35.278921	33.68	200.0	101.03	2020	599.99	Desktop
Radeon RX 6700 XT		18993	1014	20007	37.749769	35.84	230.0	82.58	2021	529.99	Desktop

Рис. 4. Зміст відправленого файлу
Fig. 4. Contents of the sent file

6 Оцінка ефективності та перспективи розвитку

Розроблений Telegram чат-бот продемонстрував ефективність у вирішенні задачі підбору комп'ютерних комплектуючих на основі запитів користувача у вільній формі. Використання методів обробки природної мови NLP та технологій наближеного порівняння fuzzy matching дозволило системі коректно інтерпретувати як точні, так і неточні або неповні запити. Завдяки цьому досягається висока точність відповідей навіть у випадках орфографічних помилок чи неточних формулювань.

Функціонал бота забезпечує зручність взаємодії: користувач отримує стислий перелік рекомендованих компонентів у чаті та може завантажити файл із розширеним описом. Додатковою перевагою є можливість надсилання цього файлу на електронну пошту, що робить систему більш практичною для кінцевого користувача. Оцінка швидкодії показала, що обробка запиту та формування відповіді відбувається у прийнятний час, що гарантує комфортне користування сервісом.

Подальший розвиток системи може включати кілька напрямів удосконалення. По-перше, інтеграція з онлайн-магазинами дозволить отримувати актуальні ціни на комплектуючі та

формувати персоналізовані кошики. По-друге, розширення датасету забезпечить охоплення більшої кількості моделей і брендів, що підвищить актуальність рекомендацій. Важливим кроком буде додавання посилань на сайти-постачальники у датасеті, щоб користувач одразу міг перейти до покупки обраного товару. Крім того, доцільно впровадити можливість створення списку обраних комплектуючих, що дозволить користувачеві зберігати цікаві варіанти для подальшого порівняння.

Перспективним є також впровадження сучасних моделей NLP на основі трансформерів (наприклад, BERT чи GPT), які здатні значно підвищити якість розуміння користувацьких запитів. Додатково можна розглянути інтеграцію голосових команд, а також реалізацію персоналізації на основі історії попередніх звернень користувача.

7 Висновки

У результаті проведеного дослідження було створено Telegram чат-бот, здатний аналізувати довільні текстові запити користувача та надавати персоналізовані рекомендації щодо підбору комп'ютерних комплектуючих. Система поєднує сучасні методи обробки природної мови NLP та алгоритми fuzzy matching, що забезпечує коректну інтерпретацію навіть неточних або неповних формулювань. На відміну від статичних онлайн-каталогів або базових чат-ботів без NLP, розроблена модель здатна гнучко реагувати на запити користувачів та формувати релевантні рекомендації, враховуючи широкий спектр характеристик комплектуючих.

Завдяки впровадженню NLP-технологій чат-бот здатний аналізувати тексти користувачів, виділяти ключові характеристики та побажання, а fuzzy matching дозволяє коригувати можливі орфографічні помилки чи неповні формулювання. Це значно підвищує точність підбору комплектуючих і робить систему доступною для користувачів із різним рівнем технічних знань.

Таким чином, розроблений Telegram чат-бот значно спрощує процес вибору комп'ютерних комплектуючих, роблячи його більш персоналізованим, інтерактивним та ефективним. Система демонструє практичну цінність і потенціал для подальшого розвитку, забезпечуючи користувачу швидкий доступ до необхідної інформації та полегшуючи прийняття рішень у процесі підбору ПК-компонентів.

REFERENCES

1. Text Processing and NLP in Python : website. URL: https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk
2. What Is Fuzzy Matching and How Can It Clean Up My Bad Data? : website. URL: https://profisee.com/fuzzy-matching/
3. Using Stanza for NLP Tasks in Python : website. URL: https://stanfordnlp.github.io/stanza/
4. Tokenization in NLP : website. URL: https://www.geeksforgeeks.org/nlp/nlp-how-tokenizing-text-sentence-words-works/
5. Text Normalization for Natural Language Processing : website. URL: https://medium.com/data-science/text-normalization-for-natural-language-processing-nlp-70a314bfa646
6. NLTK Documentation : website. URL: https://www.nltk.org/
7. What Is Stemming? | IBM : website. URL: https://www.ibm.com/think/topics/stemming
8. Lemmatization in NLP : website. URL: https://medium.com/@kevinnjagi83/lemmatization-in-nlp-2a61012c5d66
9. What is Morphological Analysis in Natural Language Processing (NLP)? : website. URL: https://www.geeksforgeeks.org/nlp/morphological-analysis-in-nlp/
10. What is Sentiment Analysis? : website. URL: https://www.ibm.com/think/topics/sentiment-analysis
11. TextBlob Documentation : website. URL: https://textblob.readthedocs.io/en/dev/
12. What is Named Entity Recognition? : website. URL: https://www.ibm.com/think/topics/named-entity-recognition
13. Industrial-Strength Natural Language Processing : website. URL: https://spacy.io/
14. FuzzyWuzzy Python Library: Interesting Tool for NLP and Text Analytics : website. URL: https://www.analyticsvidhya.com/blog/2021/06/fuzzywuzzy-python-library-interesting-tool-for-nlp-and-text-analytics/
15. RapidFuzz Documentation : website. URL: https://rapidfuzz.github.io/RapidFuzz/
16. Telegram Bot API Documentation : website. URL: https://core.telegram.org/bots/api
17. How to Build a Telegram Bot in Python : website. URL: https://core.telegram.org/bots/samples

18. PCPartPicker : website. URL: https://pcpartpicker.com/
19. Logical Increments : website. URL: https://www.logicalincrements.com/
20. Rozetka : website. URL: https://rozetka.com.ua/
21. Amazon : website. URL: https://www.amazon.com/
22. Veres O., Hadzalo O. Application of Methods of Recommendations in the Analysis of Computer Components. SISN. 2023. Vol. 14. P. 84–98. https://doi.org/10.23939/sisn2023.14.084 [in Ukrainian]
23. Chatwattana P., Yangthisarn P., Tabubpha A. The Educational Recommendation System with Artificial Intelligence Chatbot: A Case Study in Thailand : article. International Journal of Engineering Pedagogy (iJEP). 2024. Vol. 14, No. 5. P. 51–64. https://doi.org/10.3991/ijep.v14i5.48491
24. Bird S., Klein E., Loper E. Natural Language Processing with Python : textbook. O'Reilly Media. United States of America, 2009. 502 p. https://esearchgate.net/publication/220691633_Natural_Language_Processing_with_Python

Strilets Viktoriia *Ph.D, associate professor of the Department of Computer Systems and Robotics, Education and Research Institute of Computer Sciences and Artificial Intelligence, V.N. Kharkiv National University, 6 Svobody sq., Kharkiv, Ukraine, 61022*

Novikov Oleksii *student of the Education and Research Institute of Computer Sciences and Artificial Intelligence, V.N. Karazin Kharkiv National University, 6 Svobody sq., Kharkiv, Ukraine, 61022*

Chatbot model for personal computer configuration using NLP methods

Objective: to improve the convenience and efficiency of selecting personal computer components by using a Telegram chatbot with NLP methods to process user requests.
Research Methods: methods of natural language processing NLP were used to interpret user queries and generate chatbot responses; methods for building dialogue systems; and approaches to organizing software components. The Telegram chatbot was implemented based on a client-server architecture, where the client side provides interaction with the user on Telegram, and the server side handles data processing and PC component selection logic. The implementation used the following technologies: Python programming language, the python-telegram-bot library for creating the chatbot, NLP tools for analyzing and interpreting user queries, and fuzzy matching to improve search results.
As a **result**, a Telegram chatbot was created to automate the process of selecting components for personal computers, taking into account individual user needs and preferences. The system allows users to quickly receive recommendations for selecting PC components such as CPU, GPU, RAM, storage, motherboard, and power supply, considering price category, intended purpose (gaming, work, multimedia), and desired specifications. The chatbot provides a convenient interaction through Telegram, while the server side handles request processing, text analysis, and generating optimal configurations using NLP methods and fuzzy matching. For natural language processing, the libraries and tools used include Stanza, NLTK (tokenization, stemming, lemmatization), and TextBlob; for fuzzy search, RapidFuzz was applied. Using Python and the python-telegram-bot library ensures reliable system performance, flexibility in scaling, and the ability to quickly update the component database.
Conclusions: The developed Telegram chatbot allows automating the selection of PC components according to individual user needs and preferences. The system enables component selection for various use cases — gaming, work, multimedia, budget or high-performance configurations, and more. This allows users to quickly receive optimal recommendations, reduces the likelihood of errors when assembling configurations, and simplifies the component selection process. The developed system improves user convenience, optimizes the component selection process, promotes more efficient user interaction with the system.

Keywords: *chatbot, telegram, automation, NLP, NLTK, stanza, fuzzy matching, PC configurator*.

Надійшла у першій редакції 23.08.2025, в останній - 03.10.2025.
The first version has been received on 23.08.2025, the final version – on 03.10.2025.

DOI: https://doi.org/10.26565/2304-6201-2025-67-10

УДК (UDC) 004.8

Omelchenko Ihor Valeriiovych *PhD student, Department of Mathematical Modeling and Data Analysis Karazin Kharkiv National University, Svobody Sq 4, Kharkiv, Ukraine, 61022*
e-mail: ihor.v.omelchenko@gmail.com;
https://orcid.org/0009-0007-4474-4916

Strukov Volodymyr Mykhailovych *PhD in Technical Sciences, Associate Professor; Head of the Department of Mathematical Modeling and Data Analysis Karazin Kharkiv National University, Svobody Sq 4, Kharkiv, Ukraine, 61022*
e-mail: volodymyr.strukov@karazin.ua;
http://orcid.org/0000-0003-4722-3159

Impact of decoding methods in LLMs on the correctness of agent action planning in virtual environments

Relevance: The knowledge and skills acquired by Large Language Models (LLMs) from training data can be applied to the task of action planning for autonomous agents. The classical approach to text generation can violate the syntax of a JSON plan, making it difficult or even impossible to parse and use such a plan. A potential solution to this problem is the application of the Grammar-Constrained Decoding (GCD) method, which restricts the set of possible texts for generation according to a specified grammar.

Goal: To investigate the impact of the Grammar-Constrained Decoding (GCD) method (with and without reasoning) compared to classical Unconstrained Decoding (UCD) on JSON schema compliance, accuracy, and planning time for various LLMs in the Minigrid virtual environments.

Research methods: Research methods are computational experiments and comparative analysis. The studied LLM sequence decoding methods are Unconstrained Decoding (UCD) and Grammar-Constrained Decoding (GCD). The planning quality metrics used were: syntactic validity (compliance with the grammar/JSON schema), planning duration, and accuracy of plan generation.

Results: This work proposes the use of Grammar-Constrained Decoding (GCD) for agent action planning tasks that utilize Large Language Models (LLMs). A dataset of plan examples was prepared for the Minigrid environments: SimpleKeyDoor, KeyInBox, and RandomBoxKey. A comparison was conducted between Unconstrained Decoding (UCD), Grammar-Constrained Decoding (GCD), and GCD with reasoning across 10 open LLMs (from the Qwen3, DeepSeek-R1, Gemma3, and Llama3.2 families). Using the GCD method ensured the validity of the generated plans according to the grammar specified by the JSON schema. A reduction in planning time was achieved for the Qwen3:4b model by a factor of 17-25 and for the Qwen3:30b model by a factor of 6-8, by limiting the number of tokens in the reasoning chains. On average, the application of the GCD decoding method improved the accuracy of plan generation.

Conclusions: This research demonstrates that the Grammar-Constrained Decoding (GCD) method is effective in action planning tasks with LLMs. The GCD method guarantees the syntactic validity of plans according to the JSON schema, which is difficult to achieve with the UCD method. The GCD method also allows for the flexible determination of the length of reasoning chains through grammar rules, thereby controlling the planning duration.

Keywords: *artificial intelligence, machine learning, deep learning, artificial neural networks, intelligent information systems, automated information systems, natural language processing, large language model, prompt, decision making, agent, virtual environment, Minigrid.*

How to quote: I. Omelchenko and V. Strukov, “Impact of decoding methods in LLMs on the correctness of agent action planning in virtual environments”, *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology. Automated control systems*, vol. 67, pp. 101-112, 2025. https://doi.org/10.26565/2304-6201-2025-67-10

Як цитувати: Omelchenko I., and Strukov V. Impact of decoding methods in LLMs on the correctness of agent action planning in virtual environments. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.101-112. . https://doi.org/10.26565/2304-6201-2025-67-10

1 Introduction

The use of language models in agent systems has become an active area of research [1-3]. Agents operate in environments, perceive the state of the environment through observations, and execute actions chosen from a list of valid actions for that specific environment. Upon executing a chosen action, the agent receives feedback in the form of a changed environment state and, possibly, a reward signal. In each new environment, the agent must find an optimal policy. In the case of deep reinforcement learning, the agent begins learning with limited prior information about the environment. Additional prior information about the environment can be obtained without training by using pre-trained large language models (LLMs). Language models acquire generalized world knowledge from extensive training text corpora. This knowledge can be applied to specific environments. The task of planning sequences of actions, particularly abstract ones, is of special interest [4]. The planning procedure can be performed using language models.

The use of language models as a planning module in autonomous agents requires these models to have the ability to generate sequences that strictly adhere to a given plan schema. In their early stages of development, language models emerged as free-form text generators, lacking a mechanism to constrain generation to a set of texts with a predefined structure. Input and output text is represented as a sequence of tokens. The set of available tokens is defined by a token vocabulary, which is formed by training on large text corpora such that tokens consist of the most statistically common character sequences. However, to successfully solve the planning task, language models must generate token sequences that conform to a specific grammar.

Language models are trained on the task of next-token prediction in a text sequence. Pre-trained language models can be used for various tasks without fine-tuning through the method of In-Context Learning [5], where the model receives textual demonstrations of correct behavior, based on which it determines a generalized approach for solving the task. One of the approaches to generating structured data is adding examples of structured data to the training set. This enabled language models to generate structured data such as JSON, XML, and code in many programming languages with high accuracy [6]. However, this method still allows for errors in structured data generation, which lead to parsing errors and the inability to convert the generated text into data.

The need to generate strictly structured data led to the application of grammar-constrained decoding (GCD) methods to language models [7-9]. Grammar-constrained decoding ensures that text generated by language models conforms to a predefined grammar. The GCD method uses a formal grammar to describe the valid strings in a language. To describe the formal grammar, BNF (Backus-Naur Form) is used, which is a standard notation for defining the syntax of formal languages.

GCD modifies the probability distribution of tokens from the vocabulary such that tokens forbidden by the formal grammar receive a zero probability of being chosen. The generated sequences are always valid according to the schema and plausible according to the token probability distribution computed by the language model. GCD allows one to abstract away from the implementation details of the decoding mechanism and concentrate on developing a grammar that describes the sequence's structure. The grammar is represented in a declarative form and guarantees that the generated sequences will always conform to the schema.

2 Problem formulation

2.1. General problem formulation

Let V be a finite vocabulary of tokens. A language model with parameters θ defines a conditional distribution $p_\theta(w_{i+1}|w_{1:i})$, which describes the probability of the next token $w_{i+1}\in V$ following a prefix $w_{1:i}\in V^i$, where V^i is the set of all possible prefixes of length i and $w_{1:i}=(w_1,\dots,w_i)$ is the prefix of the generated sequence of tokens.

In the case of unconstrained generation, the language model computes the probability distribution over tokens as a softmax function of the logits [10]:

$$p_\theta(w_{i+1}|w_{1:i})=\frac{exp(\ell_\theta(w_{i+1}|w_{1:i}))}{\sum_{v\in V} exp\,(\ell_\theta(v|w_{1:i}))},$$

where the logit $\ell_\theta(w_{i+1}|w_{1:i})$ is the output of the final layer of the neural network for an arbitrary token w_{i+1}. Note that the right-hand side of the given equation is the definition of the $\mathrm{softmax}(\ell)$ function for a vector of logits ℓ.

At each step i, one token is selected from the conditional distribution $p_\theta(\cdot)$. Various methods for token selection exist; the simplest is selecting the token with the maximum probability. The token selection procedure can also include a temperature parameter τ, which controls the flattening of the token probability distribution. This results in an increased probability of selecting low-probability tokens and a decreased probability for high-probability ones. A lower temperature value leads to the generation of more deterministic sequences, whereas a higher temperature results in more diverse sequences. At the temperature value of τ=0, a non-zero probability remains only for the initially most probable token, and the generation becomes deterministic.

Decoding at step i can be expressed as follows:

$$w_{i+1} \sim p_\theta^{(\tau)}(w_{i+1}|w_{1:i}) \quad \text{(stochastic sampling)}, \tag{2.1}$$

$$w_{i+1}=\underset{w\in V}{argmax}\, p_\theta^{(\tau)}(w_{i+1}|w_{1:i}) \quad \text{(deterministic sampling)}. \tag{2.2}$$

For tasks that require the generation of texts with a strict structure, such as planning tasks, a lower temperature value reduces the probability of selecting tokens that violate the structure. In the case of probabilistic token selection, a language model can be made deterministic by fixing the initialization of the random number generator. Under these conditions, for a fixed input, the language model will produce a fixed output. In such a case, the language model can be represented as a deterministic mathematical function $s_{\text{out}}=g_\theta(s_{\text{in}};r)$, where s_{in} and s_{out} are input and output token sequences, respectively, and r is the initialization value for the pseudorandom number generator. For a fixed r and a fixed input prefix, the model generates a deterministic output.

To solve a task in an environment, it is necessary to sequentially select and execute actions that lead to the desired goal. Actions in the environment have a sequential nature: the success of subsequent actions depends on the outcome of previous actions. The planning task can be formulated as follows. Let the agent operate in an environment with discrete time steps t=0,1,2, ... At each time step t, the language model takes the prompt s_{prompt} and the observation o_t as input and computes a new text sequence $s_{\text{out},t}$. This sequence is a tuple $(s_{\text{reasoning},t}, s_{\text{plan},t})$, where $s_{\text{reasoning},t}$ is a string containing the model's reasoning (which may be empty), and $s_{\text{plan},t}$ is a string containing an action plan formulated based on the instructions in the prompt s_{prompt} and the observation o_t.

When a language model strictly adheres to the grammar of a plan, the generated sequence takes the following form:

$$s_{\text{plan},t}=(a_{t,1}, a_{t,2}, \dots, a_{t,n_t}),$$

where each action $a\in A$, and A is the set of valid actions in the environment.

The task of plan generation imposes structural constraints on the generated output s_{out}. First, the plan must be represented as an ordered sequence of discrete actions. Second, in any specific environment, the set of valid actions may vary, and each action has its own signature — a name and a set of parameters. This imposes a requirement on the language model that the generation process must produce a sequence structured as a series of actions, where each action conforms to one of the valid action signatures.

When selecting the next token according to the probability distribution, the chosen token may violate the plan's schema. To address this problem, the Grammar-Constrained Decoding (GCD) method can be applied, which restricts the selection of tokens to only those that do not violate the grammar.

In this work, we investigate the impact of Grammar-Constrained Decoding (GCD) method on agent action planning using language models in the Minigrid virtual environment.

Let G denote a context-free grammar that specifies the valid textual sequences of plans. We denote the language as $L(G)\subseteq V^*$, where V^* is the set of all possible strings formed from tokens in the vocabulary V. At step i, for a prefix $w_{1:i}$, we introduce the set of allowed tokens:

$$C(w_{1:i})=\{w\in V|\exists\sigma\in V^*: w_{1:i}\cdot w\cdot\sigma\in L(G)\},$$

that is, a token w is allowed if there exists some sequence continuation σ such that the concatenation of sequences $w_{1:i}\cdot w\cdot\sigma$ belongs to the language $L(G)$ with the context-free grammar G.

Then, Grammar-Constrained Decoding (GCD) restricts the possible tokens at step i to the set $C(w_{1:i})$. After applying temperature, we obtain the masked logits:

$$\ell_\theta'^{(\tau)}(w|w_{1:i})=\begin{cases}\ell_\theta^{(\tau)}(w|w_{1:i}), & \text{if} w\in C(w_{1:i});\\ -\infty, & \text{otherwise.}\end{cases}$$

The probability distribution over tokens is obtained by applying the softmax function to the masked logits [9]:

$$p_\theta^{(\tau,C)}(w|w_{1:i}) = \frac{exp(\ell_\theta'^{(\tau)}(w|w_{1:i}))}{\sum_{v\in V} exp\,(\ell_\theta'^{(\tau)}(v|w_{1:i}))}. \quad (2.3)$$

Decoding is performed by selecting the next token from the masked distribution (2.3) either stochastically (2.1) or deterministically (2.2).

2.2. Studied Environment

For the decision-making task, a set of environments from the Minigrid library was selected. This library provides a toolkit for creating two-dimensional environments that require sequential action planning. We used three environments of increasing complexity:

- **SimpleKeyDoor:** The agent must find and pick up a key which position is not known in advance, then find a door, navigate to it, and open it. This is a basic sequential planning task.
- **KeyInBox:** The key is located inside a box. The agent must first find and open the box, take the key, and then find and open the door. This increases the plan length.
- **RandomBoxKey:** The key can be located either inside a box or outside of it. The agent has to either find and pick up the key, or find and open the box. This creates a branching of choices.

Objects in the environment have attributes such as object type, position, and color. The observations from the environment include colors, but the actions selected by the agent do not. In the environments used, the color of the door and the key always match; therefore, color is not used in the planning schema.

2.3. Language Models and Tools

The application of language models in agents imposes several requirements. These include high speed of sequence generation to ensure agent responsiveness and the ability to execute language models on low-performance devices for use in robotic systems. The following are examples of language models that are freely available and can be executed on accelerators with low computational power. The Qwen3 family of language models [11] includes models with parameter numbers ranging from 0.6 to 235 billion. These models support a reasoning mode that allows for the dynamic scaling of computational resources to improve task performance. The Qwen3 family of language models demonstrates good performance on many benchmarks for code generation, mathematical reasoning, and agent tasks. The Gemma 3 family of language models [12] is designed to run on accessible accelerators with limited memory, such as personal computers with graphics cards. The Llama family of language models [13] is also freely available and can be executed on limited computational resources. Models from the DeepSeek R1 family [14] are trained using the distillation method from a large version of DeepSeek R1 onto models from the Qwen 2.5 and Llama 3 families. A distinctive feature of these models is that they are trained using reinforcement learning to generate long chains of reasoning.

Therefore, a set of modern open-weight language models was selected based on the following criteria: the ability to run on graphics accelerators with up to 24 GB of VRAM, the availability of quantized versions, and being instruction-tuned. The models used in the study are:

- **Qwen3:** qwen3:1.7b, qwen3:4b, qwen3:8b, qwen3:30b;
- **DeepSeek-R1:** deepseek-r1:1.5b, deepseek-r1:8b;
- **Gemma3:** gemma3:4b, gemma3:12b, gemma3n:e4b;
- **Llama3.2:** llama3.2:3b.

The Ollama software tool was used to execute the models and apply the grammar-constrained decoding (GCD) method, as it supports text generation conforming to a JSON schema.

For all language models, a modified prompt from the work [15] was used, which included a task description, a JSON schema, reasoning instructions, and examples of the correct planning.

3 Methods

3.1. Decoding Methods

The following sequence decoding methods were used in the language models. The first method was Unconstrained Decoding (UCD), where the most probable token is selected at each step. The prompt included an instruction to generate only JSON without reasoning; however, this does not guarantee the

syntactic correctness of the generated text. Additional post-processing was applied to the generated text to remove the reasoning fragment, if present, and to extract the substring containing the JSON object. The second method was Grammar-Constrained Decoding (GCD) without reasoning. In this case, the model generated a string that strictly conforms to the plan's schema, which guarantees syntactic correctness and requires no additional post-processing before parsing the JSON object. The third method, Grammar-Constrained Decoding (GCD) with reasoning, involved adding an optional "reasoning" field to the JSON schema, allowing the model to generate textual reasoning before formulating the final action plan.

3.2. Plan Schema

We define a plan configuration as a structure consisting of a "reasoning" text field and a "plan" sequence of actions. A plan, $P=(a_1,\dots,a_m)$, contains from 1 to 7 actions ($1 \le m \le 7$). Each action a_i is chosen from a set of options: "explore for objects", "go to object", "pick up", "drop", "toggle". The "object" parameter is a single element, while "objects" is one or more elements from a defined set $O=\{\text{door,key,box}\}$.

3.3. Evaluation Metrics

To evaluate the correctness of a plan, we used the Mean Exact-Prefix Accuracy (MEPA) metric. This metric measures the fraction of prefixes of the generated plan that exactly match the ground-truth plan. Let the ground-truth plan be $T^{(n)}=(t_1,\dots,t_n)$, where n is the number of steps in the ground-truth plan, and the generated plan be $P^{(m)}=(p_1,\dots,p_m)$, where m is the number of steps in the generated plan. The indicator function for a correct action is:

$$c_i=\begin{cases}1, & \text{if} i \le n \wedge i \le m \wedge t_i=p_i;\\ 0, & \text{otherwise.}\end{cases}$$

Let us denote the indicator of an exact prefix match for prefix of length i as

$$\sigma_i=\prod_{j=1}^{i} c_j.$$

That is, $\sigma_i=1$ if and only if all of the first i steps match. Then, MEPA for a single example is defined as the average of these indicators over all prefixes of the ground-truth plan:

$$\text{MEPA}(T^{(n)},P^{(m)})=\frac{1}{n}\sum_{i=1}^{n}\sigma_i.$$

Note that the case $n=0$ does not occur, as our dataset does not include examples with an empty plan.

As an example of calculating the MEPA metric, if $T=(\text{explore,go to,toggle})$ and $P=(\text{explore,go to,drop})$, then $\sigma_1=1,\sigma_2=1,\sigma_3=0$. The metric value will be $(1+1+0)/3 \approx 0.67$.

For a set of K examples, we calculate the macro-averaged MEPA:

$$\text{MEPA}_{macro}=\frac{1}{K}\sum_{k=1}^{K}\text{MEPA}_k,$$

where $MEPA_k$ is the MEPA value for the k-th example. Hereafter, the metric $\text{MEPA}_{\text{macro}}$ will be denoted as MEPA.

4 Experiments

We conducted computational experiments in three environments from the Minigrid suite: SimpleKeyDoor, KeyInBox, and RandomBoxKey.

4.1. Experimental Setup

All experiments were conducted on hardware with an NVIDIA RTX 3090 GPU (24 GB VRAM). We used the Ollama 0.11.10 framework to run quantized versions of the models (Q4_K_M). To ensure reproducibility, the generation temperature parameter was set to 0, and the top_k sampling limit was set to 1. The maximum length of the generated sequence was limited to 4096 tokens.

We compared three decoding methods:

1. **UCD (Unconstrained Decoding):** The model was instructed to generate only JSON. The output was then post-processed to extract a JSON object.
2. **GCD (Grammar-Constrained Decoding):** Generation was constrained by a JSON schema for the plan, which did not include a field for reasoning.
3. **GCD+R (GCD with Reasoning):** The JSON schema included an optional "reasoning" field, allowing the model to generate reasoning before the plan.

4.2 SimpleKeyDoor Environment

For the SimpleKeyDoor environment, there are six unique abstract states of the environment. When colors are taken into account, this results in 31 examples with a correct plan. The task of the language model is to generate a plan based on an observation that most closely matches the ground-truth plan.

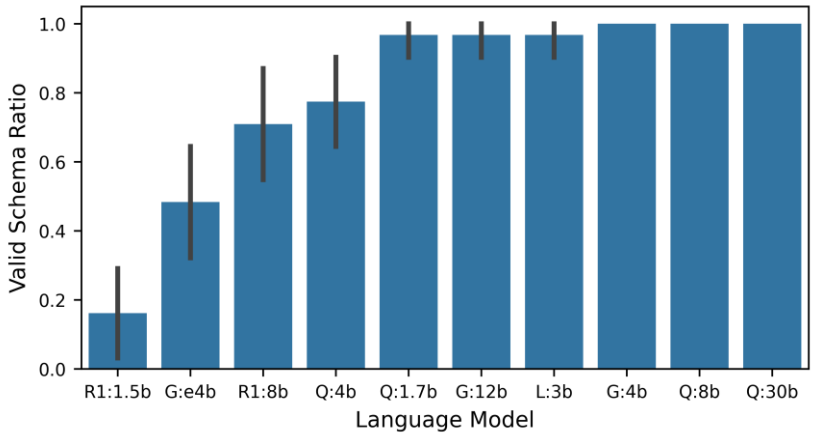


Fig. 4.1 Fraction of correctly generated plans according to the schema for different language models. The black vertical bars represent the 95% confidence interval. Legend for language models: Q:1.7b — Qwen3:1.7b, Q:4b — Qwen3:4b, Q:12b — Qwen3:12b, Q:30b — Qwen3:30b, G:4b — Gemma3:4b, G:12b — Gemma3:12b, G:e4b — Gemma3n:e4b, L:3b — Llama3.2:3b, R1:1.5b — DeepSeek-R1:1.5b, R1:8b — DeepSeek-R1:8b.

Рис. 4.1 Частка коректно згенерованих планів відповідно до схеми для різних мовних моделей. Чорні вертикальні лінії позначають 95% довірчий інтервал. Умовні позначення мовних моделей: Q:1.7b — Qwen3:1.7b, Q:4b — Qwen3:4b, Q:12b — Qwen3:12b, Q:30b — Qwen3:30b, G:4b — Gemma3:4b, G:12b — Gemma3:12b, G:e4b — Gemma3n:e4b, L:3b — Llama3.2:3b, R1:1.5b — DeepSeek-R1:1.5b, R1:8b — DeepSeek-R1:8b.

When using the UCD method (Fig. 4.1), not all models were able to generate syntactically correct JSON, which made further processing impossible. The DeepSeek-R1:1.5b and Gemma3n:e4b models proved to be the least reliable. The GCD method, by definition, guarantees 100% schema correctness.

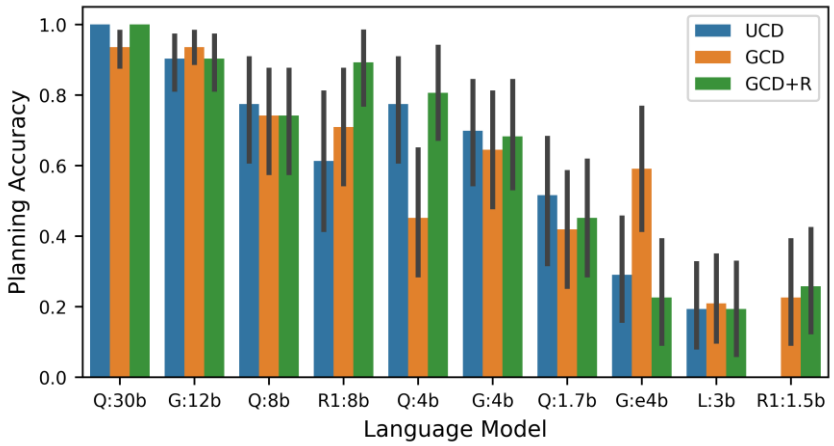


Fig. 4.2 MEPA for different language models and generation methods. The legend and confidence intervals are the same as in Fig. 4.1

Рис 4.2 МЕРА для різних мовних моделей та методів генерації. Умовні позначення та довірчі інтервали збігаються з такими для Рис. 4.1

Figure 4.2 shows that for this relatively simple task, the most powerful models (Qwen3:30b, Gemma3:12b) achieve nearly perfect accuracy regardless of the decoding method. This indicates that the task is comparatively simple for them. Meanwhile, for smaller models such as Gemma3n:e4b, the GCD method slightly improves the result. Overall, for 7 out of 10 models, the planning performance either remained unchanged or improved.

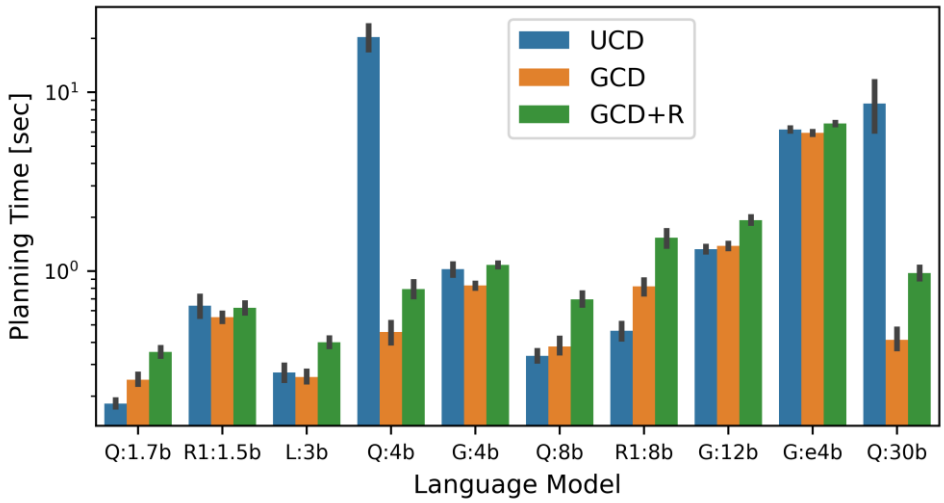


Fig. 4.3 Average generation time for a single plan for different language models and generation methods. The legend and confidence intervals are the same as in Fig. 4.1

Рис. 4.3 Середній час генерації одного плану для різних мовних моделей та методів генерації. Умовні позначення та довірчі інтервали збігаються з такими для Рис. 4.1

The vertical axis of Fig. 4.3 shows the plan generation time on a logarithmic scale. As is evident from this figure, the UCD method for the Qwen3:4b and Qwen3:30b models was significantly slower: by a factor of ≈ 25 for Qwen3:4b and ≈ 8 for Qwen3:30b. This is because these models, despite instructions not to generate reasoning, produced long chains of reasoning before the JSON response. The GCD method causes the Qwen3:4b and Qwen3:30b models to immediately generate a structured result, which significantly reduces planning time and makes these models more suitable for real-time agent systems.

4.3. KeyInBox Environment

In the KeyInBox environment, the planning complexity increases as additional steps appear: find the box, open it, and only then take the key and open the door.

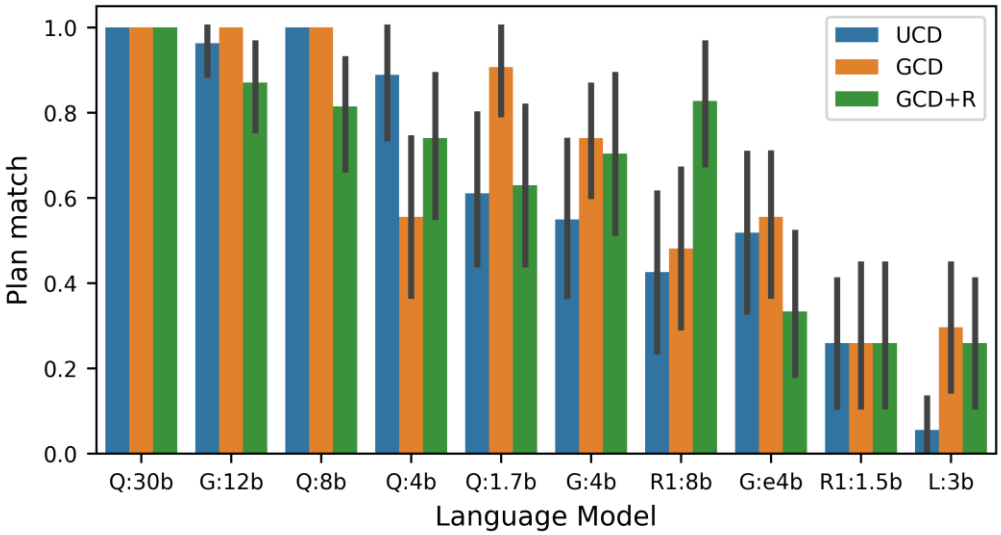


Fig. 4.4 MEPA for the KeyInBox environment for different language models and generation methods. The legend and confidence intervals are the same as in Fig. 4.1

Рис. 4.4 MEPA для середовища KeyInBox для різних мовних моделей та методів генерації. Умовні позначення та довірчі інтервали збігаються з такими для Рис. 4.1

Fig. 4.4 demonstrates that models such as Qwen3:30b continue to show high accuracy. The result for the DeepSeek-R1:8b model is particularly interesting: its accuracy significantly increases when using

GCD with reasoning. This may indicate that for models trained to generate long chains of reasoning, providing a special "reasoning" field within the JSON schema improves the quality of the final plan. Overall, for 9 out of 10 models, the planning results remained unchanged or improved.

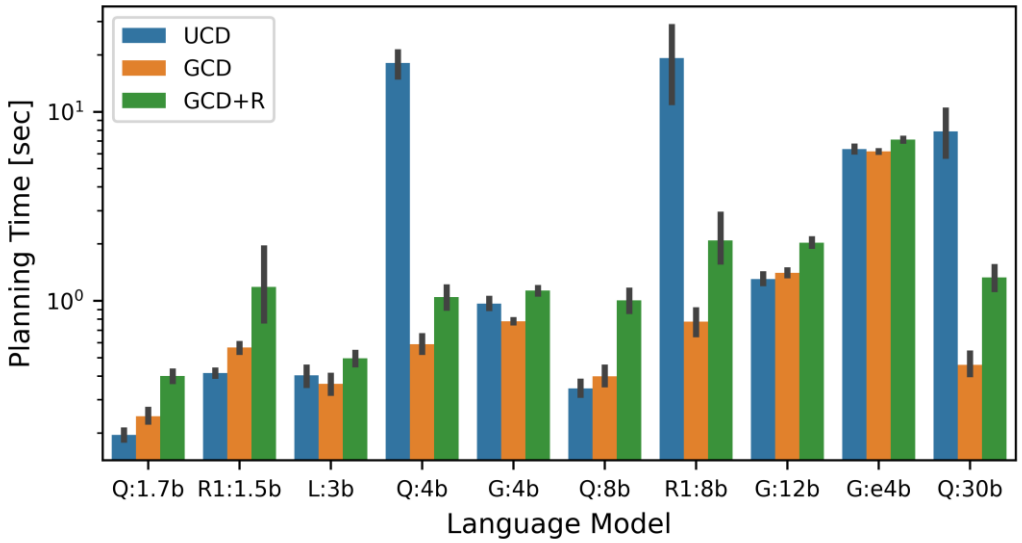


Fig. 4.5 Average single-plan generation time for the KeyInBox environment across different language models and generation methods. The legend and confidence intervals are the same as in Fig. 4.1

Рис. 4.5 Середній час генерації одного плану для середовища KeyInBox для різних мовних моделей та методів генерації. Умовні позначення та довірчі інтервали збігаються з такими для Рис. 4.1

Fig. 4.5 confirms the trend observed in the previous experiment: the Qwen3:4b and Qwen3:30b models take significantly more time to generate a plan using the UCD method due to the generation of redundant reasoning, whereas GCD ensures a fast and predictable response time. For this environment, unlike the previous one, this phenomenon is also observed for the DeepSeek-R1:8b model. Unconstrained generation was approximately 17 times slower for the Qwen3:4b model, 6 times slower for Qwen3:30b, and 9 times slower for DeepSeek-R1:8b.

4.4. RandomBoxKey Environment

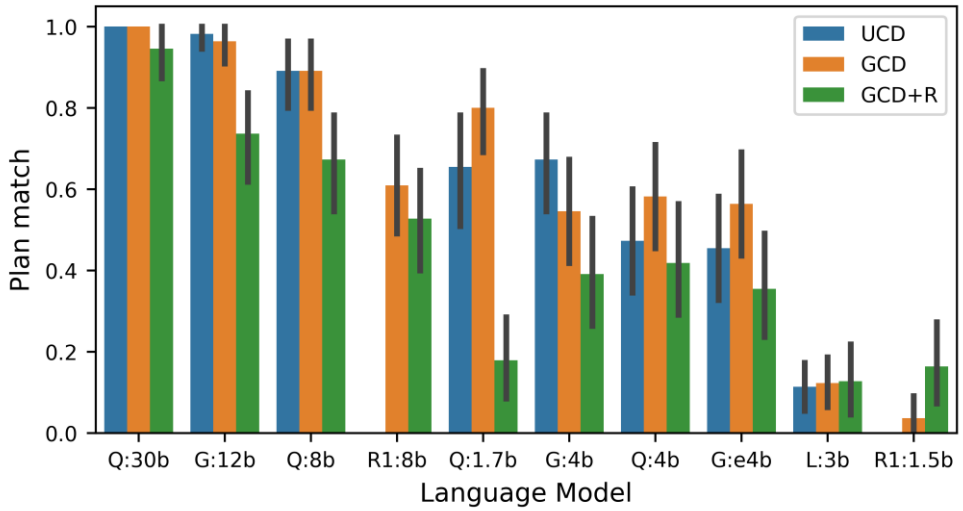


Fig. 4.6 MEPA for the RandomBoxKey environment for different language models and generation methods. The legend and confidence intervals are the same as in Fig. 4.1

Рис. 4.6 MEPA для середовища RandomBoxKey, різних мовних моделей та методів генерації. Умовні позначення та довірчі інтервали збігаються з такими для Рис. 4.1

The results in Fig. 4.6 highlight the importance of GCD in complex tasks. Both DeepSeek-R1 models failed completely with unconstrained generation, as they exceeded the 4096 token limit by generating redundant reasoning. GCD not only allowed them to generate a response but also significantly improved accuracy. For models like Qwen3:30b, accuracy remains high, but the speed advantage of GCD becomes significant. Overall, for 8 out of 10 models, the planning accuracy either did not change or improved.

Similar to the results for the previous two environments, Fig. 4.7 shows a significantly longer planning time for the Qwen3:4b and Qwen3:30b models when using the UCD method, making this approach impractical for complex tasks. Unconstrained generation was ≈ 22 times slower for the Qwen3:4b model and ≈ 6 times slower for Qwen3:30b.

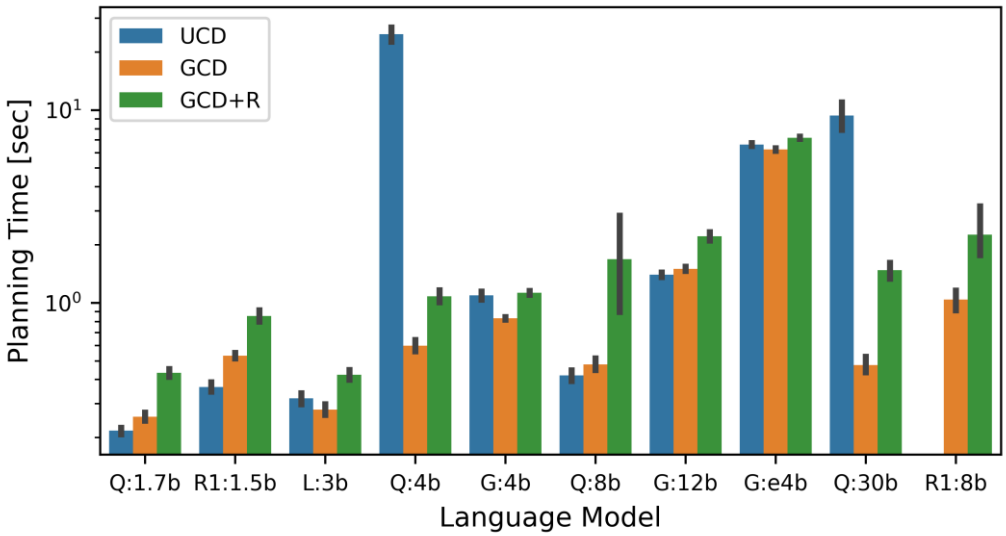


Fig. 4.7 Average generation time for a single plan for the RandomBoxKey environment for different language models and generation methods. The legend and confidence intervals are the same as in Fig. 4.1
Рис. 4.7. Середнє значення часу генерації одного плану для середовища RandomBoxKey для різних мовних моделей та методів генерації. Умовні позначення та довірчі інтервали збігаються з такими для Рис. 4.1

Table 1. A comparison of different LLM mean MEPA precentage improvement

Табл. 1 Порівняння середнього значення відсоткового відносного покращення MEPA для різних LLM

LLM	Mean MEPA improvement, %
deepseek-r1:1.5b	-
llama3.2:3b	151.2
deepseek-r1:8b	69.9
gemma3n:e4b	44.9
qwen3:1.7b	19.4
gemma3:4b	4.5
qwen3:4b	3.5
gemma3:12b	1.8
qwen3:30b	0
qwen3:8b	-1.4

5 Discussion

Table 1 demonstrates the average relative percentage improvement for the MEPA metric across three environments for various language models using the GCD method relative to the baseline (results

of the UCD method). The DeepSeek-R1:1.5b model had a zero MEPA value for the UCD baseline, making it impossible to calculate the relative improvement. For models with high MEPA value (Qwen3:8b, Qwen3:30b, Gemma3:12b), the increase is not significant, as their results were already close to the maximum.

The conducted experiments lead us to several key conclusions about the impact of the GCD method on the accuracy, schema compliance, and action planning time of agents using language models.

The UCD method, despite explicit instructions in the prompt prohibiting reasoning, cannot guarantee the absence of reasoning or control the length of reasoning chains. Many models, especially smaller ones (DeepSeek-R1:1.5b, Gemma3n:e4b), often generated syntactically incorrect JSON or added redundant reasoning, which made automatic parsing of the result impossible. This creates significant obstacles for building stable agentic systems. In contrast, the grammar-constrained decoding (GCD) method completely solves this problem by guaranteeing 100% syntactic validity of the output. This is a crucial advantage for automated data processing systems.

One of the most significant results is the substantial acceleration of the planning process when using GCD. Models prone to generating extensive reasoning (in particular, the Qwen3:4b and Qwen3:30b models for all studied environments, and DeepSeek-R1:8b for the KeyInBox environment) demonstrated significantly longer generation time with the UCD method. By constraining the output to a strict JSON schema, GCD allows for controlling the presence or length of the reasoning chain. This makes language models much more suitable for systems that require real-time decision-making.

In simple environments (SimpleKeyDoor), the advantages of GCD in planning accuracy were minor for relatively large models. However, as the task complexity increased, the impact of structured generation became more noticeable. In the most complex environment (RandomBoxKey), the unconstrained approach led to the complete failure of the DeepSeek-R1 models due to exceeding the token limit. GCD not only made it possible to obtain a response from them but also significantly increased its correctness.

6 Conclusions

This study accomplished the following. We proposed the use of Grammar-Constrained Decoding (GCD) for agent action sequence planning tasks in virtual environments, as an alternative to the classic Unconstrained Decoding (UCD) method. A dataset containing examples of correct action plans was prepared for three Minigrid environments. Computational experiments were conducted to generate agent action plans in environments from the Minigrid suite using UCD, GCD, and GCD with reasoning. The performance of these methods was evaluated based on the following metrics: syntactic validity, planning duration, and plan generation accuracy. Finally, we analyzed and compared the results of applying the UCD, GCD, and GCD with reasoning methods to the agent action planning task across the three Minigrid environments.

The results of the study demonstrated that, unlike the classic UCD method, applying the GCD method ensures that the generated sequence conforms to the specified plan grammar. This eliminates syntax errors and guarantees the successful syntactic parsing of the generated JSON plan. This outcome is particularly significant for relatively small language models, such as DeepSeek-R1:1.5b and Gemma3n:e4b. When using the classic UCD method, these models generate grammatically incorrect sequences in more than 50% of cases, whereas applying GCD guarantees adherence to the grammar.

Measurements of planning duration indicate that the classic UCD method does not guarantee a limit on the number of tokens in the generated sequence; specifically, a significant number of tokens are generated in reasoning chains. Some language models ignore instructions that prohibit the generation of long reasoning chains. Applying the GCD method resulted in a significant reduction in planning time compared to the UCD method across various environments for the following models: the planning duration for the Qwen3:4b model decreased by a factor of 17–25, for the Qwen3:30b model by a factor of 6–8, and for the DeepSeek-R1:8b model by a factor of 9 (in the KeyInBox environment). The application of GCD led to this substantial reduction in planning time by constraining the length of the reasoning chains.

Applying the GCD method improves, on average, the plan generation accuracy as measured by the MEPA metric for most models. The most significant accuracy gain was observed for the DeepSeek-R1 models. When using the UCD method, these models generated excessively long reasoning chains, which led to exceeding the token limit and resulted in failed plan generation. In contrast, with the GCD and GCD with reasoning methods, these models were guaranteed to generate a plan.

This research has shown that applying GCD improves, on average, the plan generation accuracy for most of the models tested and guarantees the syntactic validity of the generated plan according to the JSON schema. In the case of LLMs that already demonstrate high accuracy, the main benefit of GCD is the reduction in planning time rather than an improvement in accuracy. Therefore, the use of the GCD method is beneficial for enhancing the performance of language models in planning tasks.

REFERENCES

1. I. Dasgupta et al., "Collaborating with language models for embodied reasoning", arXiv [cs.LG]. 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.00763.
2. W. Huang et al., "Inner Monologue: Embodied Reasoning through Planning with Language Models", arXiv [cs.RO]. 2022. Available: DOI:10.48550/arXiv.2207.05608.
3. B. Hu, C. Zhao, P. Zhang, et al., "Enabling Intelligent Interactions between an Agent and an LLM: A Reinforcement Learning Approach", Reinforcement Learning Journal, Vol. 3, P. 1289–1305, 2024. https://arxiv.org/abs/2306.03604
4. R. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning", Artificial Intelligence, Vol. 112, P. 181–211, 1999. https://people.cs.umass.edu/~barto/courses/cs687/Sutton-Precup-Singh-AIJ99.pdf
5. T. B. Brown et al., "Language Models are Few-Shot Learners", arXiv [cs.CL]. 2020. [Online]. Available: https://arxiv.org/abs/2005.14165.
6. S. Minaee et al., "Large Language Models: A Survey", arXiv [cs.CL]. 2025. [Online]. Available: https://arxiv.org/abs/2402.06196.
7. Y. Dong et al., "XGrammar: Flexible and Efficient Structured Generation Engine for Large Language Models", rXiv [cs.CL]. 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2411.15100
8. S. Geng, M. Josifoski, M. Peyrard, and R. West, "Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning", arXiv [cs.CL]. 2024. [Online]. Available: https://doi.org/10.18653/v1/2023.emnlp-main.674.
9. L. Beurer-Kellner, M. Fischer, and M. Vechev, "Guiding LLMs The Right Way: Fast, Non-Invasive Constrained Generation", arXiv [cs.LG]. 2024. [Online]. Available: https://dl.acm.org/doi/10.5555/3692070.3692216
10. K. Murphy, "Probabilistic machine learning: an introduction", MIT press, 2022.
11. A. Yang et al., "Qwen3 Technical Report", arXiv [cs.CL]. 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.09388.
12. G. Team et al., "Gemma 3 Technical Report", arXiv [cs.CL]. 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.19786
13. A. Grattafiori et al., "The Llama 3 Herd of Models", arXiv [cs.AI]. 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.21783.
14. DeepSeek-AI et al., "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning", arXiv [cs.CL]. 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.12948 .
15. I. Omelchenko and V. Strukov, "On the impact of prompts on agent performance in a virtual environment", Bulletin of V. N. Karazin Kharkiv National University, series Mathematical modelling. Information technology, Automated control systems, Vol. 65, P. 56–63, 2025. https://doi.org/10.26565/2304-6201-2025-65-07

Омельченко Ігор Валерійович *Аспірант, кафедра математичного моделювання та аналізу даних Харківський національний університет ім. В.Н. Каразіна. майдан Свободи, 4, Харків, Харківська область, 61022*

Струков Володимир Михайлович *к.т.н., доцент; завідувач кафедри математичного моделювання та аналізу даних Харківський національний університет ім. В.Н. Каразіна. майдан Свободи, 4, Харків, Харківська область, 61022*

Дослідження впливу методів декодування у мовних моделях на коректність планування дій агентів у віртуальних середовищах

Актуальність. Знання та навички, отримані великими мовними моделями (LLM) з навчальних даних, можуть бути використані в задачі планування дій автономних агентів. Класичний підхід до генерації тексту може порушувати синтаксис JSON-плану, що ускладнює або робить неможливим синтаксичний розбір та використання такого плану. Можливий підхід до вирішення цієї проблеми полягає у застосуванні методу декодування з обмеженням граматики (GCD), що обмежує множину можливих текстів для генерації відповідно до заданої граматики.

Мета. Дослідити вплив методу декодування з обмеженням граматики GCD (з міркуваннями та без) порівняно з класичним необмеженим декодуванням UCD на відповідність JSON-схемі, точність та час планування дій різними LLM у віртуальних середовищах Minigrid.

Методи дослідження. Методи дослідження: обчислювальний експеримент, порівняльний аналіз. Методи декодування послідовностей в LLM: Unconstrained Decoding (UCD), Grammar-Constrained Decoding (GCD). Використані метрики якості планування: синтаксична валідність (відповідність граматиці/JSON-схемі), тривалість та точність планування.

Результати. Запропоновано використовувати метод декодування з обмеження граматики (GCD) в задачах планування дій агентів з використанням великих мовних моделей (LLM). Підготовлено датасет з прикладами планів для середовищ Minigrid: SimpleKeyDoor, KeyInBox, RandomBoxKey. Проведено порівняння методів Unconstrained Decoding (UCD), Grammar-Constrained Decoding (GCD) та GCD з міркуваннями для 10 відкритих LLM (сімейств Qwen3, DeepSeek-R1, Gemma3, Llama3.2). Використання методу GCD забезпечило валідність згенерованого плану відповідно до граматики, заданої JSON-схемою. Досягнуто скорочення часу планування для моделей Qwen3:4b у 17-25 разів, для Qwen3:30b — у 6-8 разів за рахунок обмеження кількості токенів в ланцюжках міркувань. У середньому застосування методу декодування GCD покращило точність генерації плану.

Висновки. Дослідження демонструє, що застосування методу декодування з обмеженням граматики (GCD) є доцільним в задачах планування дій з використанням LLM. Метод GCD гарантує синтаксичну валідність планів відповідно до JSON-схеми, що складно досягти з методом UCD. Метод GCD дозволяє гнучко визначати довжину ланцюжків міркувань через правила граматики і тим самим контролювати тривалість планування.

Ключові слова: *штучний інтелект, машинне навчання, глибоке навчання, штучні нейронні мережі, інтелектуальні інформаційні системи, автоматизовані інформаційні системи, обробка природної мови, велика мовна модель, промпт, прийняття рішень, агент, віртуальне середовище, Minigrid.*

Надійшла у першій редакції 09.09.2025, в останній - 10.10.2025.
The first version has been received on 09.09.2025, the final version – on 10.10.2025.

ВІСНИК ХАРКІВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
імені В.Н. Каразіна

серія **«Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»**

Випуски даної серії розповсюджуються у академічних та наукових колах України та за її межами з метою оперативного висвітлення досліджень у таких актуальних галузях: математичне та комп'ютерне моделювання, обчислювальний експеримент, теорія і прикладні методи обробки інформації, захист інформації, програмно-апаратні системи інформаційного або управляючого призначення, застосування математичного моделювання та системного аналізу у високих, наукоємних технологіях, враховуючи технології створення програмної продукції. Приймаються роботи, що відносяться до напрямів фізико-математичних і технічних наук (бажаний об'єм 6-18 сторінок). Усі рукописи рецензуються.

Примітка. Протягом 2025-26 рр. редакційна колегія при інших рівних умовах надаватиме перевагу роботам, що представлені англійською мовою, якщо стаття отримала схвалення при рецензуванні.

Офіційний сайт **http://periodicals.karazin.ua/mia**
http://mia.univer.kharkov.ua
Email: journal-mia@karazin.ua

Bulletin of V.N. Karazin Kharkiv National University

series **«Mathematical modeling. Information technology. Automated control systems»**

This series are distributed in academic and scientific circles of Ukraine and abroad for the purpose of timely coverage of research in the following topical areas: mathematical and computer modeling, computational experiment, theory and applied methods of information processing, information protection, software and hardware systems of control and information management, applications of mathematical modeling and system analysis in high, science-intensive technologies, including technologies of software products creation. Articles belonging to the fields of physical, mathematical and technical sciences are accepted (recommended length 6-18 pages). All submissions are peer-reviewed.

Note. For the years 2025-26, all other conditions being equal, the Editorial Board will give preference to articles submitted in English and approved by the peer-review.

Official website **http://periodicals.karazin.ua/mia**
http://mia.univer.kharkov.ua
Email: journal-mia@karazin.ua

Наукове видання

Вісник Харківського національного університету імені В. Н. Каразіна

Серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління

Випуск 67

Збірник наукових праць

Українською та англійською мовами

Комп'ютерне верстання О. О. Афанасьєва

Підписано до друку 31.10.2025 р.
Формат 60x84/8. Папір офсетний. Друк цифровий.
Ум. друк. арк. – 13,5.
Обл.– вид. арк. – 16,9.
Наклад 50 пр. Зам. № 53/2025
Безкоштовно

Видавець і виготовлювач
Харківський національний університет імені В. Н. Каразіна
61022, м. Харків, майдан Свободи, 4
Свідоцтво суб'єкта видавничої справи ДК №3367 від 13.01.09

Видавництво Харківський національний університет імені В. Н. Каразіна
тел.: 705-24-32